Proceedings of the 4th WSEAS Int. Conf. on Information Security, Communications and Computers, Tenerife, Spain, December 16-18, 2005 (pp271-276)

A Distributed Cooperative Multi-layer Security Architecture For Virtual Networks

SIOMON O'SULLIVAN, REINER DOJEN, TOM COFFEY
Data Communications Security Laboratory
Department of Electronic and Computer Engineering
University of Limerick
IRELAND
http://www.dcsl.ul.ie

Abstract: - The proliferation of the Internet in recent years has radically altered the landscape of network security. Despite continuing attempts by organisations to secure corporate networks, the high rate of successful network attacks have led to significant financial losses. This paper concerns the protection of virtual networks against network attack. A Distributed Cooperative Multi-layer Security Architecture (DCMSA) for virtual networks is proposed. This architecture provides distributed security enforcement combined with centralised management. Further, it uses the synergy of different security tools working together and multiple layers of defence to provide enhanced protection.

Key-Words: - Network security architectures, firewalls, intrusion detection systems

1. Introduction

The proliferation of the Internet in recent years has radically altered the landscape of network security. According to the *2004 CSI/FBI Computer Crime and Security Survey* [1], 98% of the companies surveyed employed firewalls and 68% IDS, to protect the communication network. Despite these security measures, 53% of the companies surveyed reported unauthorised use of their computer. It is evident that despite the high utilisation of network security tools, computer networks are still not adequately protected. Further, the emergence of virtual networks, due to the many benefits for both employers and employees, create new challenges for the provision of network security. Current network security tools, such as firewalls and IDS, are often LAN oriented and fail to provide adequate protection for all systems in a virtual network.

This paper proposes a *Distributed Cooperative Multi-layer Security Architecture(DCMSA)* capable of protecting all hosts in a virtual networks. This architecture uses distributed security enforcement combined with centralised management. Further it employs cooperation of a wide range of security tools and multiple layers of security to greatly enhance the protection afforded to virtual networks.

2. Limitations of Current Techniques

Despite the level of protection offered by current network security tools such as firewalls [2] and Intrusion Detection System (IDS) [3], the persistently high rate of intrusions underlines the need for improvement. Network security tools clearly possess a number of vulnerabilities that expose the network to attack. These vulnerabilities can be grouped under the following headings: location of deployment, complementary and lack of defence-in-depth.

2.1 Location of Deployment

Network security tools have two possible locations of deployment, namely network-based and host-based, each location having its own associated shortcomings.

2.1.1 Network-based Security Tools

Network-based security tools are ubiquitous at present. Organisations extensively use network-based firewalls and IDS to protect the network. Network-based security tools offer advantages such as ease of management, concentrated security and the ability to detect distributed attacks. However, network-based security tools have the following shortcomings:

- Network-based security tools fail to protect all systems in the *virtual network*. Systems located outside the LAN in which the network security tool is deployed, are afforded no protection and hence represent weak links in network security.
- Network-based security tools generally do not possess the necessary keys to analyse host-to-

host encrypted traffic [4]. Network-based security tools are therefore incapable of fully protecting a network, since attacks present in host-to-host encrypted traffic cannot be detected or prevented.

- Network-based security tools are single points of failure. Network-based security tools protect many hosts simultaneously. Hence a single vulnerability in network-based security tools exposes many hosts to attack.
- Network-based security tools are required to analyse traffic relating to many hosts. Due to increasing networking speeds, network-based security tools face a constant battle to avoid dropping packets or reducing network throughput.
- Network-based firewalls cannot prevent insider attacks since a firewall cannot filter traffic it doesn't see. After virus incidents, insider abuse of network access is the most cited form of attack or abuse [1].
- Switched networks limit the effectiveness of NIDS as the NIDS can now only monitor traffic belonging to the segment it is located on. Additional NIDS need to be deployed to monitor traffic on other segments.
- The capability of NIDS to accurately detect malicious activity depends on its ability to reconstruct packets in a manner consistent with the receiving host. However, insufficient information on the wire can result in both entities seeing different packet streams, increasing the number of false positives and false negatives [5].

2.1.2 Host-based Security Tools

While moving security to the host solves many of the problems associated with network-based security tools, host-based security tools possess the following shortcomings:

- Host-based security tools generally lack the ability to detect distributed attacks. For example, an attempt to connect to multiple systems on a specific port could indicate an attacker trying to map the network. Host-based security tools typically only have access to information available on the host they reside on and hence cannot correlate activity on different systems.
- Host-based security tools often lack centralised management. In the absence of centralised management, each host-based security tool must be managed separately, increasing the administrative burden.

2.2 Complementary Shortcomings

To avail of the benefits of interconnectivity, a firewall must permit some level of network access to the network or host. However, attacks can be present in this permitted network traffic. Lack of attack detection capabilities can result in malicious traffic being unwittingly admitted by a firewall. An IDS can fill this gap by detecting attacks in network traffic admitted by the firewall. Thus firewalls and IDS can be thought of as naturally complementary. However, the passive nature of an IDS requires human intervention to actually prevent an attack from completing. This lack of real-time prevention can result in an attack damaging a host before an effective response to an alert is initiated. Additionally in a large network with many sensors deployed, the strain of monitoring alerts can result in critical alerts being missed, enabling attacks to successfully complete. Further an IDS often is prone to a high rate of false positives [6].

2.3 Lack of Defence-in-Depth

The principle of *defence-in-depth* states that multiple layers of security offer greater protection than that which can be achieved by any single layer. Each individual layer of security represents a separate obstacle that must be overcome in order to complete a successful attack. A failure in one layer does not mean that security as a whole is compromised as there are still additional layers defending the resource.

Network security tools often rely on too few layers of security to protect the network or host. Each technique and information source used by a network security tool represents a separate obstacle for an attacker to overcome. Network security tools using a single technique or information source fail to comprehensively protect the network or host. For example, a firewall solely using packet header information to prevent attacks, leaves the network or host vulnerable to application-level attacks. These attacks are better prevented using information in the packet payload. In addition, an IDS solely using *misuse detection* [7] to detect attacks, leaves the network or host vulnerable to new attacks. These attacks are better detected by *anomaly detection* [7]. It is imperative to realise that there is no silver bullet in relation to network security. No one technique or information source used can protect against all attacks.

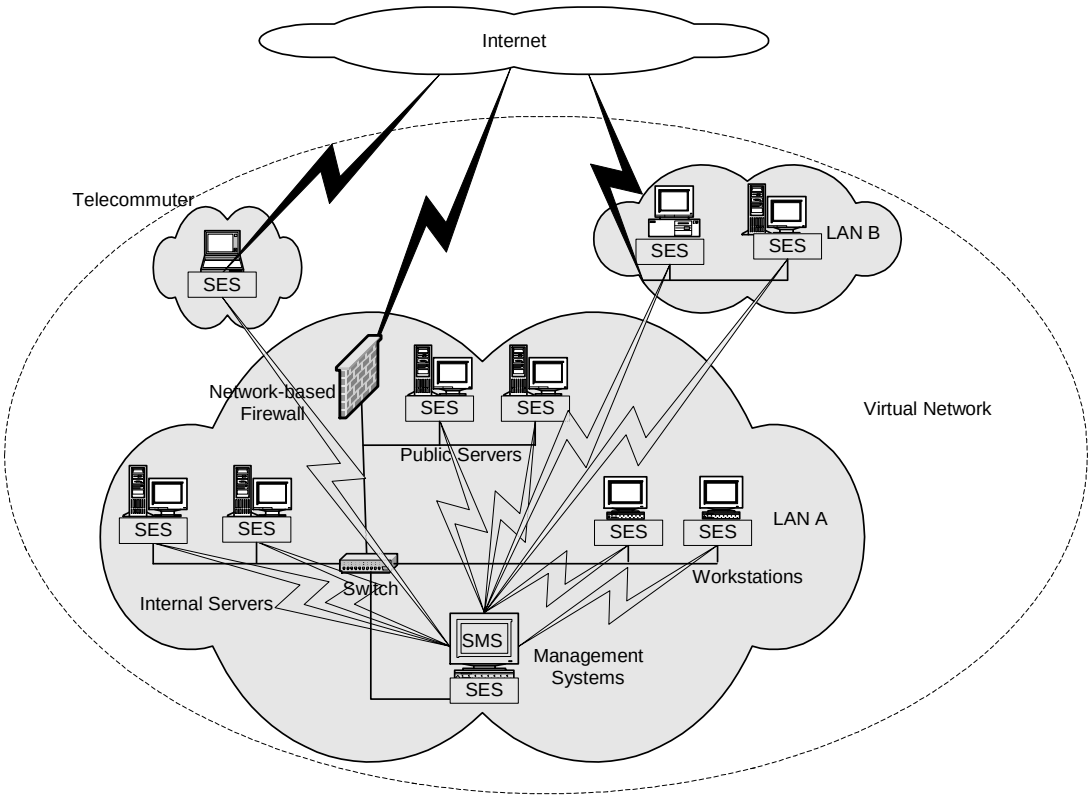

Fig. 1: Distributed Cooperative Multi-layer Security Architecture

3. A Distributed Cooperative Multi-layer Security Architecture

The following proposes a new Distributed Cooperative *Multi-layer Security Architecture (DCMSA)* for *virtual networks* (Figure 1). The *DCMSA* comprises a number of features designed to enhance the protection afforded to the *virtual network*. The *DCMSA* employs distributed security enforcement combined with centralised management. This scheme overcomes location of deployment shortcomings, maximising the protection offered to the *virtual network*. In addition the *DCMSA* facilitates cooperation of a wide range of network security tools. The synergy of network security tools working together allows greater protection than when working separately. Further the *DCMSA* adheres to the principle of defence-in-depth by using multiple layers of security. This increases the difficulty of successfully attacking the network.

The security enforcement system (SES) (Figure 2) enforces security on each host in the virtual network. The SES comprises a network sensor, application sensor, system call sensor, multi-level firewall (m-firewall), multi-level IDS (MIDS) and response module, all of which reside on the host.

3.1 Sensors

The function of each sensor is to intercept an event (packet, message or system call), extract sufficient information on the event and send the information to both the *m-firewall* and *MIDS*. The event is subsequently frozen until such time as the sensor receives a decision from the *response module*. Based on this decision, the sensor can either pass on the event as normal or drop the event completely. Each sensor represents a separate obstacle that must be overcome in order to mount a successful attack. Thus the *SES* can block an attack at three different levels – the network level (*network sensor*), application level (*application sensor*) and system call level (*system call sensor*).

3.1.1 Network Sensor

The *network sensor* intercepts packets between the IP layer and the network adapter of the host**.** Packets received from the network adapter are intercepted before reaching the IP layer. Thus network-level attacks can be prevented from damaging the host. In addition packets being transmitted to the network are intercepted before reaching the network adapter. Thus covert communication such as the leaking of sensitive information, attacks on other hosts etc. can be prevented.

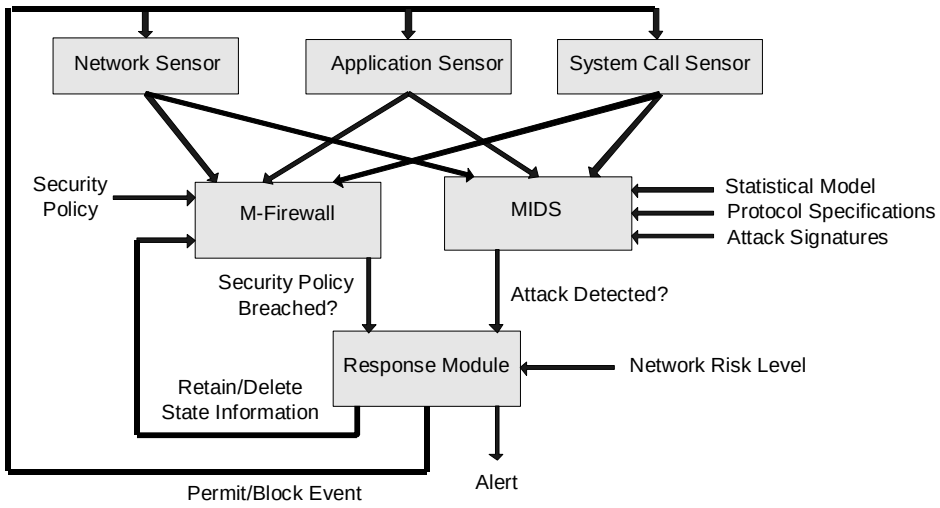

Fig. 2: Security Enforcement System

The *network sensor* sends all the relevant information about the packet (IP address, port number etc.) to both the *m-firewall* and the *MIDS*. If the *response module* indicates that the packet should be permitted then the *network sensor* passes the packet on as normal. However, if the *response module* indicates that the packet should be blocked then the *network sensor* drops the packet.

3.1.2 Application Sensor

The *application sensor* intercepts messages between the TCP/UDP layer and the application. Thus application-level attacks can be prevented from subverting the application. In addition messages passed from the application are intercepted before reaching the TCP/UDP layer. Thus covert communication such as the leaking of sensitive information, attacks on other hosts etc. can be prevented. The *application sensor* must fully understand the used protocols in order to send the relevant information (for example service fields, commands etc.) to the *m-firewall* and *MIDS*. Hence a different *application sensor* is needed for each application. If the *response module* indicates that the message should be permitted then the *application sensor* passes the message to the application as normal. However, if the *response module* indicates that the message should be blocked then the *application sensor* drops the message.

3.1.3 System Call Sensor

The *system call sensor* intercepts system calls made by an application. The *system call sensor* sends all the relevant information about the system call (for example user ID, process ID, system resource being accessed etc.) to the *m-firewall* and *MIDS*. If the *response module* indicates that the system call should be permitted then the call is passed to the corresponding kernel function and completes as normal. However, if the *response module* indicates that the call should be denied then the system call is dropped and a standard kernel error is sent to the application.

3.2 Multi-level Firewall

The *multi-level firewall (m-firewall)* controls access to network and system resources by enforcing a multi-level security policy. To facilitate this requirement the security policy contains three distinct categories of rules, each category directly corresponding to the network, application or system call level.

Network level rules are similar to traditional *packet filter* rules. Application level rules control access to the network using information about the message such as protocol service fields, commands etc. System call level rules control access to system resources using information about the system call such as user ID, resource being accessed etc.

3.3 Multi-level IDS

The *Multi-level IDS* (*MIDS*) investigates whether an event (packet, message or system call) is an attack. Thus the *MIDS* is in some respects similar to an NIDS. However, the *MIDS* extends basic NIDS functionality beyond detecting attacks at the network and application level to also detecting attacks at the system call level. The *MIDS* uses multiple techniques to detect a wide range of attacks. Three techniques are proposed for inclusion: *protocol anomaly detection*, *pattern*

matching and *statistical models*. However, the *MIDS* can be extended to include newly emerging techniques, should the need arise in the future.

3.4 Response Module

The function of the *response module* is to issue an appropriate response to each event. Responses include permitting an event, blocking an event and sending an alert to the security management system (*SMS)*. To determine the appropriate response, the *response module* receives the verdicts of both the *m-firewall* and *MIDS*. If an event breaches the security policy and/or contains an attack then the *response module* sends a message to the corresponding sensor indicating that the event should be blocked and/or send an alert to the management console. On the other hand, if an event is authorised by the security policy and/or no attack has been detected, the *response module* sends a message to the sensor indicating that the event should be permitted.

The *response module* also uses the *network risk level* as a factor in determining the appropriate response. The *network risk level* is a global signal issued by the management console, representing the current state of security of the network as a whole.

3.5. Security Management System

The security management system (SMS) manages the security enforcement system (SES) of each host in the virtual network. The SMS is designed to both improve the accuracy of security enforcement on each host and ease management overhead. The SMS comprises a vulnerability scanning module, statistical model module, security policy module, protocol specification module, pattern matching module and alert correlation module, all of which reside on a single management console.

3.5.1 Vulnerability Scanning Module

The vulnerability-scanning module is periodically run to create an accurate up-to-date network profile of each host. The network profile contains information such as the operating system type and version, network services running, vulnerabilities etc. on each host. This information enables other management modules to tailor their operation specifically to each host. The vulnerability database is a repository of known vulnerabilities for various operating systems, protocols and applications. This database is used by the vulnerability scanning module to identify potential vulnerabilities on each host. The database is updated automatically with the latest vulnerabilities for optimum protection.

3.5.2 Statistical Model Module

The statistical model module uses various system attributes (protocols used, applications used, network utilisation, system resources accessed etc.) to generate a model of normal behaviour for each host in the virtual network. The statistical model module must monitor the host over a period of time to build an accurate statistical model of the host in various conditions. This information is used by the MIDS on each host to detect attacks that manifest themselves as anomalous system attributes. The statistical model for each host is stored in the statistical model database and can later be retrieved by the host when required.

3.5.3 Security Policy Module

The security policy module uses the network profile to select security policy rules from the policy database explicitly for each host. The policy database is a repository of policy rules for common applications, hosts and users. The security policy for each host and user is also stored in the policy database and can be later retrieved when required. For example, each time a user logs onto a host, the security policy for that particular user is transmitted to the host.

3.5.4 Protocol Specification Module

The protocol specification module uses the network profile to select protocol specifications from the protocol database explicitly for each host. The protocol database is a repository of protocol specifications. The protocol specification module stores the relevant protocol specifications for each host in the protocol database, which can later be retrieved by the host when required. This information is used by the MIDS to detect attacks that manifest themselves as an anomaly in the specification of the used protocol (for example buffer overflow attack). The protocol database is automatically updated with the latest protocol specifications for optimum protection.

3.5.5 Pattern Matching Module

The pattern matching module uses the network profile to select attack signatures from the signature database explicitly for each host. The signature database is a repository of attack signatures for known network level, application level and system call level attacks. The pattern matching module stores the relevant attack signatures for each host in the signature database, which can later be retrieved

by the host when required. This information is used by the MIDS to detect attacks uniquely identified by the presence of a particular pattern (for example directory traversal attack). The signature database is automatically updated with the latest attack signatures for optimum protection.

3.5.6 Alert Correlation Module

The alert correlation module receives alerts from the response module of each host in the virtual network and correlates these alerts to detect distributed attacks (attempts to map the network etc.). Each alert describes the event (packet, message or system call) in sufficient detail to enable the alert correlation module to determine whether the event is malicious or in fact benign. The alert correlation module also calculates a global signal representing the state of security of the network as a whole, namely the network risk level. The network risk level is based on the quantity and nature of alerts received. The network risk level is sent to the response module of each host and is factored into its decision to permit/deny an event. The alert correlation module provides the host-based components of the DCMSA with an overview of the current security state of the entire network.

4. Conclusion

In this paper a Distributed Cooperative Multi-layer Security Architecture (DCMSA) for virtual networks was proposed. This architecture comprises a number of features designed to enhance the protection afforded to the virtual network. The proposed architecture employs distributed security enforcement combined with centralised management. Further it facilitates cooperation of a wide range of security tools and provides multiple layers of security.

Moving security enforcement to the host ensures that all systems in the virtual network are protected. Further benefits include the ability to defend against attacks hidden in host-to-host encrypted traffic and the avoidance of single points of failure. In addition insider attacks can be prevented. At the same time the centralised management incorporated into the DCMSA ensures that important capabilities such as detecting distributed attacks and ease of management are retained.

The DCMSA uses the synergy of security tools working together to offer greater protection than when working separately. A greater level of confidence in host security is achieved by involving both the m-firewall and MIDS in access decisions. To facilitate real-time prevention of attacks, false positives are reduced by the generation of a network profile. Further the generation of a global signal representing the state of security of the network as a whole, enables host-based tools to make intelligent access decisions.

The DCMSA follows the principle of defence-in-depth by using multiple layers of security. Security is enforced at three different levels, namely the network, application and system call level. At each level the m-firewall and MIDS use the corresponding information source to prevent attacks. This scheme allows security policies to exercise greater control over the set of actions permitted on the host. Similarly by operating at multiple levels the MIDS can detect a wider range of attacks. At each level multiple IDS techniques to increase the probability of detecting an attack.

References:

[1] CSI/FBI, 2004 Computer Crime and Security Survey, http://www.gocsi.com/, 10 (6) (2004).

[2] Anuar, B., Yaakob, M. and Idna, Y., “Red Alert: Approach for Firewall Policies Update Mechanism”, WSEAS Transaction on Computer, Vol. 3, No. 5, Nov. 2004, pp.1451-1454.

[3] Kang,D., Kim, B. and Oh, J., “Protocol Anomaly and Pattern Matching Based Intrusion Detection System, WSEAS Transaction on Communications, Vol. 4, No. 10, Oct 2005, pp. 994-1001.

[4] Ioannidis, S., Keromytis, A.D., Bellovin, S. M. and Smith, J. M., “Implementing a Distributed Firewall”, ACM Conference on Computer and Communications Security, Athens Greece, 2000, pp.190-199.

[5] Watson, D., Smart, M., Malan, G.R. and Jahanian, F., “Protocol Scrubbing: Network Security Through Transparent Flow Modification”, IEEE/ACM Transactions on Networking, Vol. 12, No. 2, 2004, pp.261–273.

[6] Eschelbeck, G. and Krieger, M., “Eliminating Noise from Intrusion Detection Systems”, Information Security Technical Report, Vol. 8, No. 4, 2003, pp. 26-33.

[7] Verwoerd, T. and Hunt, R., “Intrusion Detection Techniques and Approaches”, Computer Communications, Elsevier, Vol. 25, No. 15, 2002, pp.1356-1365