MANAGEMENT OF SECURITY IN TCP/IP HOSTS USING DEDICATED MONITORING APPLICATIONS

Rui Costa Cardoso and Mário M. Freire
Networks and Multimedia Group, Institute of Telecommunications- Covilhã Lab
Department of Informatics, University of Beira Interior
Rua Marquês d'Ávila e Bolama, P-6200-001 Covilhã, Portugal

Abstract: In this paper, we present an approach for detection of vulnerabilities in network systems, using autonomous applications. The main aim is to enable the dynamic, intelligent and autonomous detection of vulnerabilities and exposures in systems and to make it available to network administrators. Our approach will reduce the amount of data sent to network administrators by currently used tools, and therefore present only relevant information preprocessed by our application, which by it self can bring a natural enhancement to the performance of the network overall security.

Key words: Network Security, Vulnerabilities.

1. INTRODUCTION

There are innumerous security problems that arise from the use of networked environments. Today's networks are bigger and complex. There are many elements to manage in a network (hosts, switches and routers). Making every active element of the network secure, it is a sizable task, which is liable to allow security breaches. Moreover, system administrators often found themselves attacked before they even knew the existence of the vulnerability. Hackers often access to that information before the vendors are able to correct the vulnerabilities and it is difficult for network administrator to keep update. There is also lack of skills among system administrators to security tasks. Monitoring for vulnerabilities and security breaches, verify

The original version of this chapter was revised: The copyright line was incorrect. This has been corrected. The Erratum to this chapter is available at DOI: 10.1007/978-0-387-35703-4_21
D. Gaïti et al. (eds.), *Network Control and Engineering for QoS, Security and Mobility II*

security holes fix and maintaining a security policy, are issues not neglected, but are cared without the proper attention from the network administrators; in many occasions because their overworked. The amount of knowledge that a system administrator needs to stay actualized about the new network security threats is overwhelming [1]. New network threats appear in a daily basis [2] as software and hardware development continue, and new products emerge. Hackers will seek those vulnerabilities, and discover new and innovative ways of exploiting them. The lack of knowledge about the latest network vulnerabilities is a big liability to network security. Every day use of the network also exerts pressures on the security, through using insecure services, which can produce unwanted security breaches in the network. The huge amount of traffic on the network also masks unauthorized behavior.

Research activities on intrusion and fault detection started in early 1980s with the introduction of the concept of computer threats and detection of misuse [3]. This was the advent of intrusion detection systems, proposed to detect statistically unexpected patterns of behavior. Investigators used a database of rules to describe known attacks against systems, and the intrusion detection mechanisms look for events matching those rules. This help drive the collection of system vulnerabilities and their integration into a database; and since it was desirable to have the intrusion detection system to detected new attacks, the interest in classifying vulnerabilities grew, looking towards being able to detect new ones.

Although the goal of intrusion detection is simply to detect intrusions, these systems do not detect intrusions; they only identify evidence of intrusions either while they are in progress or after they have occurred [4]. Techniques used for intrusion detection lie into two basic categories: anomaly detection and misused detection, both, with its own advantages and drawbacks [4]-[6]. On the other hand, detection of security faults (holes) in hosts can anticipate the occurrence of service failures and compromises. This paper focus on the detection and monitoring security faults, vulnerabilities and threats in network environment systems.

The remainder of this paper is organized as follows. In Section 2 we briefly describe related work methodologies used for detection of vulnerabilities. In Section 3, we present a set of vulnerability assessment metrics, which could be used to improve the Vulnerability Assessment of our application. In section 4, we present an architectural overview of our tool based on the integrated implementation of the components of the DeNoFaS agent (Agent for Detection and Notification of Security Faults in Networked Systems). Main conclusions are presented in Section 5.

2. RELATED WORK

There are several available models [7], applicable in the area of intrusion detection systems, based on behavior modeling and fault trees. Test and failure models provide also host based analysis by checking logs, software versions and monitoring performance metrics [8], [9]. There are specific packages of software that work with vulnerabilities, like NESSUS [10] and SAINT [11], but do not are dedicated monitoring tools. They only present results based in the detection process.

Our solution takes in consideration data from previous scans (historical data) and uses it in the inference process. Our application acts independently of the requester, monitors the hosts systems and makes decisions based on his knowledge, leading to personalized reports, which could be used to improve the overall network security.

3. VULNERABILITY ASSESSMENT METRICS

In our approach, we considered a set of vulnerabilities metrics to give us a more insightful view of our network hosts security risk. Although each host is relevant in our network security, they present differentiated risks. In our metrics, we considered each machine graded with a specific factor of vulnerability risk based in the following criteria:

1) The type of service(s) performed
2) The number of services available
3) The relevance to the network of the services provided in each host (ex DNS)
4) The net segment where it stands (behind a firewall, in a DMZ)
5) The relevance of the host to the users
6) The relevance of the service to the users
7) The number of open ports
8) The number of vulnerability associated with a service
9) Grade of each service vulnerability
10) The number of Trojans that use open ports available the host machine
11) Number of system vulnerabilities associated with each host
12) Grade of each system vulnerabilities
13) Number of route/firewall depth from the network gateway
14) Host Latency

With the establishment of vulnerability metrics, we can measure the state of security of your network Although we cloud use more intrusive methods to gather more information about specific services vulnerabilities, we

decided to use only port scan techniques, banner extraction and basic interaction with principal services to gather information about our network.

4. TOOL

4.1 Components

Our agent architecture is built from four main components: A Interface to interact with the outside world. A Knowledge Base which aggregate, several sources of information in a coherent fashion. The information provided by the Knowledge Base gave the agent a better understanding of the status of the network and allow him to produce better reports using his inference tool. The last component the Vulnerability Assessment is essential to the process because it give us the information about our network in a specific instant. Components and sub-components of our tool are represented in figure 1.

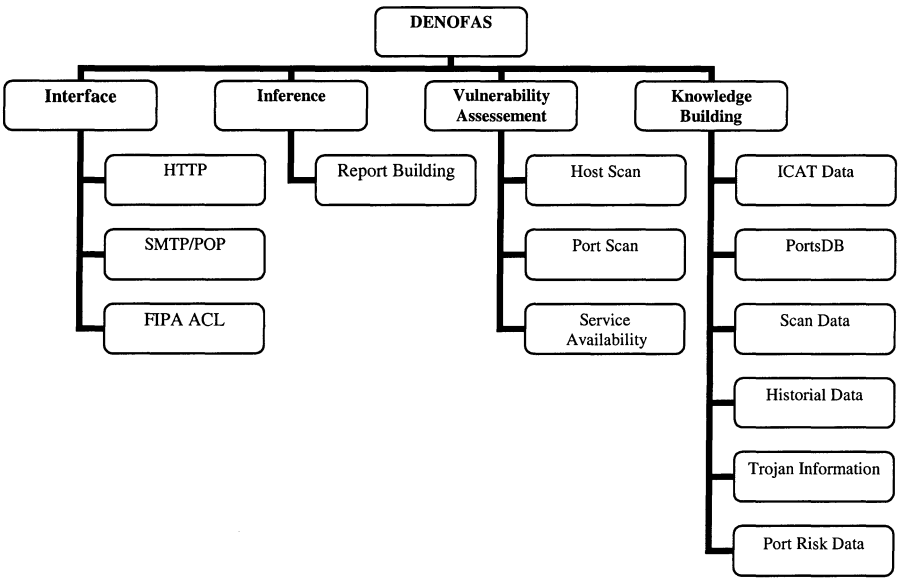

Figure 1. Components of Denofas Agent.

4.2 Interfaces

Here, we present the interfaces of Denofas Agent. Our application has two operational interfaces that are used to interact with the user. The HTTP interface is used to receive solicitations from the users. The SMTP/POP Interface allows the Denofas Agent to receive requests from its users and to send personalized reports about the network status to the network administrator.

4.2.1 HTTP Interface

The functionalities associated with this interface are the following:

Single IP Audit

Auditing Features: Performs a scan on IP for open ports, gathering information regarding the system (all ports from 0-1024, along with a set of ports that are known to host several services and possible Trojans, database services and specific open ports from operating system). Examines the gathered information, for matching with ICAT [12] records, displaying the security warnings found. Scan the IP from the user system with vulnerability tests (must be the same IP of the HTTP Client). Can be run whenever the scan is requested, or periodically reporting the changes in the system as well as new vulnerabilities found.

Specific Network Audit

Auditing Features: Tracks the given IP's for live hosts, auditing all live hosts, giving a full report on each one of them. Report is sent to the network administrator reported in the Whois [13] database records. It performs a scan in a similar away has the previous Audit. It also can be run whenever the scan is requested or in a monitoring way reporting the changes in systems directly to the administrator periodically.

Implicit Network Audit

Auditing Features:Tracks the network from which the given IP belongs by using information gathered from Whois database records. Reports are sent to the network administrator email, provided by a query to the Whois database records. In the rest is equivalent to the previous audit.

In figure 2, we present a screen from our interface showing a request for a Single IP audit.

4.2.2 SMTP/POP Interface

The SMTP and the POP are used to exchange messages with the users of the Agent. The Denofas uses SMTP to send his reports to the administrator our to a specific email address in case of a single audit. The POP is accessed by the Agent to inquire if there is any pending request.

Specific Network Audit

To specify a Network Audit the user has to use a specific syntax (set of rules) recognized by the Agent, which allow him to send requests to the Agent.

Implicit Network Audit

To specify a Implicit Network Audit the user needs to known the syntax used by the agent to be correctly interpreted. The reports in this case has in the previous one, use the Whois database to get the email were to send the results.

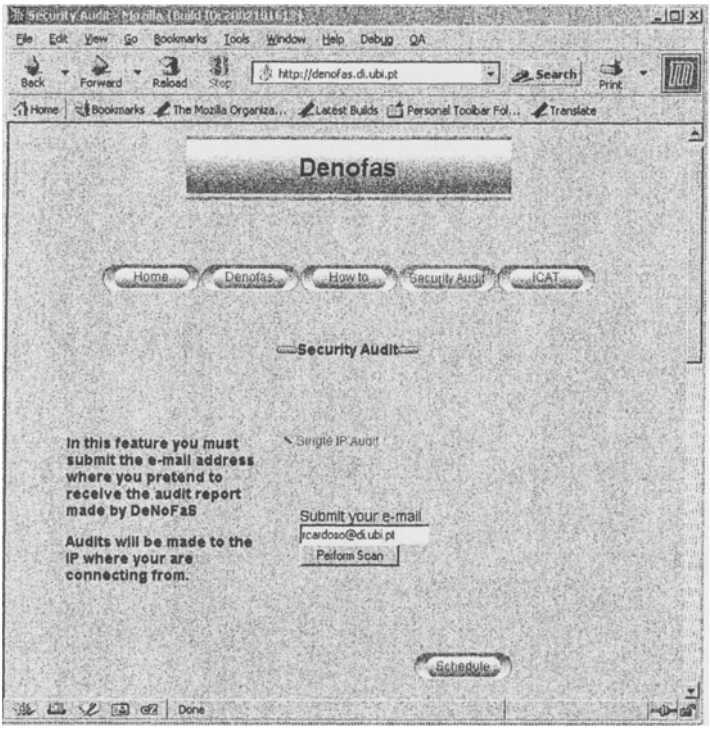

Figure 2. HTTP Interface.

4.2.3 FIPA Compliant Interface

In the future, we intend to implement a FIPA Interface. It will be full integrated in our Agentcities Testbed [14], which will allow making queries about the networked systems through inquire agents.

A representation of that system is shown in the figure 3.

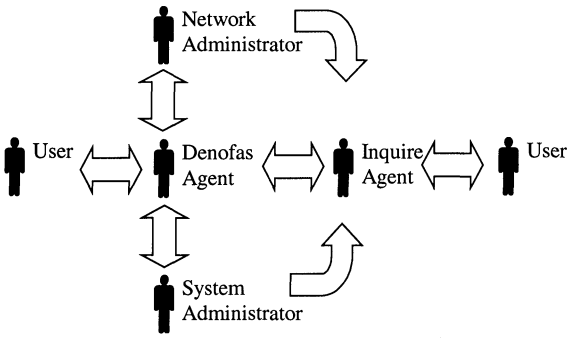

Figure 3. Denofas Agent Interactions with Agents and Users.

4.3 Building of Knowledge

Using the vulnerability information collect from external databases our tool compiles a list of the discovered vulnerabilities and displays them in a report to be send to the network administrator. It will provides details about each vulnerability, such as the vulnerable hosts and the level of severity of the vulnerability, a description of the vulnerability, and actions that should be taken to correct the weaknesses. As new vulnerabilities are discovered, they are periodically incorporated in our knowledge database by our update knowledge facilitator.

The process of Denofas Knowledge Building integrates several data sources, from online resources to scan data. A synthetic overview of the knowledge building follows.

ICAT [12], [15] data used by the agent is automatic updated from the web. The agent uses the complete ICAT database (exporticat.htm) and the information about all vendors, products, and version numbers contained within ICAT (vpv.htm). All this data is processed internally and is used as input to the Denofas Knowledge Base.

In the knowledge building, the agent also uses the PortsDB [16] to specify what service is using what port. Despite IANA [17] maintains a list with this information that is also used to build PortsDB database, crackers do not register their port usage at IANA and neither many companies do.

Concerning the Port Risk data and the Trojan lists, they represent data collected from the Internet with the latest information about the Ports that present more risk to systems, and the Trojans associated with ports. All this information is processed internally by its knowledge base. When the agent starts another scan, it will use the stored available data to produce a personalized report. A representation of that system is shown in figure 4.

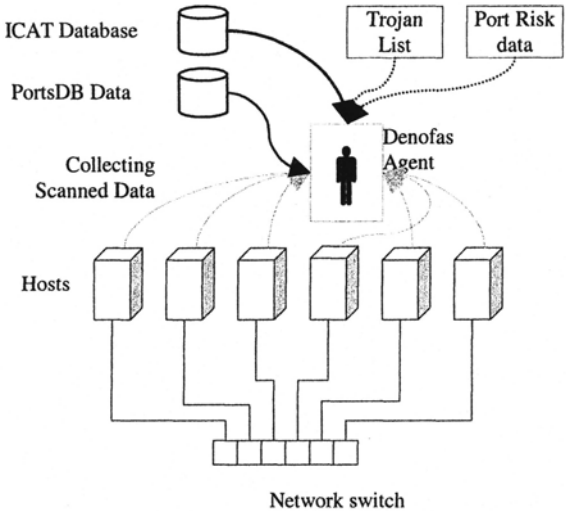

Figure 4. Denofas Knowledge Building.

4.4 Scan Procedure

The scanner component of our tool tries to discover security weak points on the network before intruders can exploit them. The scanner allows us to automatically compile relevant data about the network hosts by identifying all information about open ports and vulnerabilities associated with network services. The process is finished by gathering all the collected data, which is used as an input to our knowledge base component.

The scan process is used in three different modes: as a single host scan, as a network block scan (whois based approach), and as range scan to specific subnets in a network block. Even though the scan procedure could be used in a specific IP audit, the scan process should be used on a recurring basis to monitor the network using schedule scans in regular periods.

The Denofas information gathering about the host/network starts by getting data from the user about IP address. It is followed by information from Whois Servers [13], which give the agent specific data about the

network and the administrator. With this information, it will start the scan process by detecting alive hosts, after which it will scan each host for open ports. For each open port, it will try to get information about the running services.

The reception of a specific IP address or a range of IP addresses, in one of the two interfaces, activates the detection process. The Agent, with that information, access the corresponding WHOIS Server, to collect data about the network and network administrators, namely: inetnum, netname descr, country, admin-c, person, address, phone, fax-no, e-mail, nic-hdl; from the corresponding network. The gathered data is used to detect network boundaries and to build personalized reports to the network administrator. The following process is to try to detect open ports in each IP referred initially. After this process is concluded, the Denofas Agent starts to get data about open services available in the ports detected, and the detection process ends.

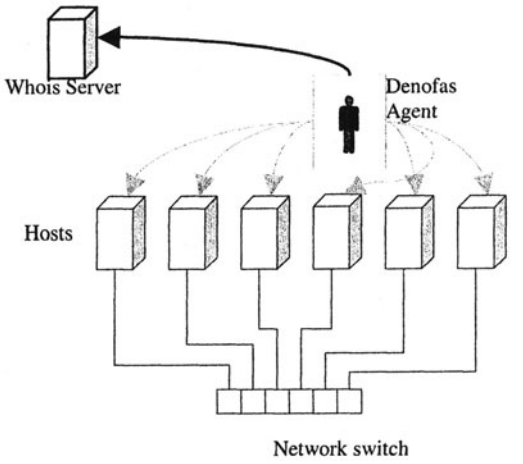

Figure 5. Denofas Network Scan Procedure.

4.5 Building of Reports

One of the most relevant components of our tool is the elaboration of the report. The role of the Denofas is to process the acquired data so that it could present clever suggestions and recommendations to the network administrator, based on his knowledge of the situation. Also important is to present this data in an easy and legible format for the network administrator. We intend to give information to the network administrator that allows him to have an accurate idea of the type of problems that exist in its server. The

information presented in the report is the same data that was acquired previously. However, that data are now treated in a way that a user of the agent has more easiness in analyzing the report.

Before the notification process can begin, Denofas initiate the report building procedures; by upgrading his knowledge base with the new data collected in the scan process and also upgrade (if needed) the data received from external sources. The most important procedure in the Agent activities is the knowledge extraction. This procedure is started after all data is available in the Denofas Knowlege Base, partially implemented in a PostgreSQL database.

5. CONCLUSIONS AND FUTURE WORK

In our agent application, the autonomy and decision capabilities were a relevant factor to the development. We tried to implement a set of rules that could be used to make the report procedure as clear as possible with sufficient focus in the main problems. We tried to enforce this flexibility and capability of extracting valuable information from his knowledge base by producing accurate results capable of corresponding to the needs of host and network administrator.

The main contribution of this work was the development of a VA tool that uses actualized sources of CVE [18], and therefore is permanently updated. Based in previous knowledge, it can track the evolution of the security in a timely scale, preventing future security errors and improving the overall security of the network. It also allows immediate notification of severe threats to security. In the future, we intend to develop the FIPA [14], [19] compliant interface to allow interaction with other agents that work on the behalf of administrators and users.

ACKNOWLEDGEMENTS

Part of this work has been supported by the Group of Networks and Multimedia of the Institute of Telecommunications - Covilhã Lab, Portugal, within SAPA (Software Agents for Prevention and Auditing of Security Faults in Netwoked Systems) Project.

We also acknowledge the National Institute of Standards and Technology NIST [20], for developing the ICAT and for keeping it available to developers of security tools, and the PortsDB [16] for their effort in trying to provide updated information about all known services associated with

specific ports. This work would not be possible without these freely available sources.

REFERENCES

1. A. Householder, K. Houle, and C. Dougherty, "Computer Attack Trends Challenge Internet Security", IEEE Computer, Security and Privacy - Supplement, April 2002, pp. 5-7.
2. URL: http://www.cert.org .
3. J. P. Anderson, "Computer Security Threat Monitoring and Surveillance", James P. Anderson, Co. Fort Washington, PA, 1980.
4. R. A. Kemmerer and G. Vigna, "Intrusion Detection: A Brief History and Over-view", IEEE Computer, Security and Privacy - Supplement, April 2002, pp. 27-29.
5. C. Manikopoulos and S. Papavassiliou, "Network Intrusion and Fault Detection: A Statistical Anomaly Approach", IEEE Communications Magazine, Vol. 40, No. 10, pp. 76-82.
6. B. Kim, J.Jang, and T. M. Chung, "Design of Network Security Control Systems for Cooperative Intrusion Detection", in Information Networking, I. Chong (Ed.), Heidelberg, Springer Verlag, LNCS 2344, 2002, pp. 389-398.
7.. D. E. Denning, "An Intrusion-Detection Model", IEEE Transactions on Software Engineering, February 1987.
8. G. Quo, J. Rudraraju, R. Modukuri, S. Hariri, "A Framework for Network Vulnerability Analysis", Proceedings of IASTED International Conference Communication, Internet & Information Technology, November 18-20, 2002, pp. 289-294.
9. M. Yi, C. Hwang, "Design of fault tolerante Architecture for Intrusion Detection systems Using Autonomous Agents", Proceedings of The International Conference on Information Networking (ICOIN'2003), February 12-14, 2003, pp. 913-922.
10. URL: http://www.nessus.org .
11. URL: http://www.saintcorporation.com .
12. URL: http://icat.nist.gov .
13. URL: http://www.ripe.net/perl/whois .
14. URL:http://www.agentcites.org .
15. P. Mell, "Understanding the World of your Enemy with I-CAT (Internet-Categorization of Attacks Toolkit)", in 22nd National Information System Security Conference, October 1999.
16. URL: http://www.portsdb.org .
17. URL: http://www.iana.org .
18. URL: http://www.cve.mitre.org .
19. URL: http://www.fipa.org .
20. URL: http://www.nist.gov .