SANS Security Essentials
GSEC Practical Assignment
Version 1.2f

UNIX System Security in a Large Enterprise Environment – Axent ESM
Michael O'Neill
June 22, 2001

Introduction

A problem that many companies with a large enterprise environment face is maintaining a strong, consistent security policy across many and varied UNIX servers. Quite often system administration activities are inconsistent, build documents are not kept up to date, patch levels fall behind and OS versions get out of sync. What you can easily end up with is a server or two that are not as secure as they should be. Since the enterprise is only as secure as its weakest link, having a tool that can centrally manage a large enterprise can be very useful in maintaining a uniform set of policies. I would like to present to you a review of a product that is designed to perform such a function, Symantec Corporation's Axent Enterprise Security Manager (Axent ESM). I will focus on UNIX implementations using Axent ESM 5.1. I will discuss an overview of the product, structuring a large enterprise into manageable domains and how Axent ESM security policy modules relate to common UNIX server security issues.

Product Overview

Based on the principle that security strategy can be grouped into three broad areas: prevention, detection and response, Axent ESM can be categorized as a prevention/detection tool. Axent ESM is a software-auditing tool designed to manage and report security data and policies across a wide range of client/server platforms. Axent ESM differs from security scanning products such as Nessus and ISS Internet Scanner in that it is a host-based tool. A security policy template is built and is resident on the agent server. An agent daemon (esmd) is listening on the agent server. Policy runs are initiated from a central manager, the agent server runs the policy checks and the results are sent back to the manager. Axent ESM will run on a variety of UNIX platforms, NetWare, Windows NT and OpenVMS. Some of the major functions of Axent ESM include:

- Manage security policies.
- Evaluate system conformance with security policies.
- Check systems for vulnerabilities or unauthorized privileges.
- Provide integrity checks.
- Detect changes to security setting or files.

Equally important to note is that Axent ESM won't:

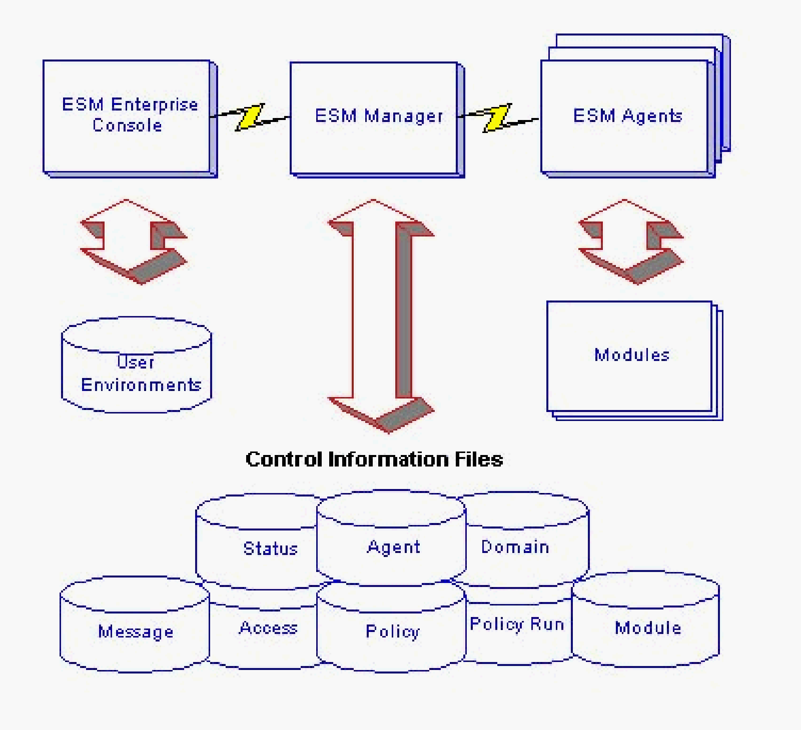
- Change operating Systems
- Detect Dynamic Reconfigurations
- Run in real time.

Using the concept of a centrally managed security policy database, Axent ESM uses a rating system that allows you to measure your level of server security across a network.

Vulnerabilities are weighted according to their severity. A user can easily discern the state of a server's security based on the value of the rating. This rating, of course, is fairly meaningless if the proper effort is not put into building a good security policy template. Axent ESM report generation can be configured to provide a brief summary of the rating of each agent or domain, or a very detailed report on every individual check within each policy module. Reports are created in HTML format and can be configured to include pie charts or bar graphs. Since reports are stored on the console hard drive, I recommend you take the time in the reports menu to find some suitable middle ground on report detail, for the sake of console workstation disk space.

Typical Topology

The three components of an Axent ESM installation are the Manager, agents and the console. The manager is the communications hub between the agents and the console and houses the security policy database. The console is the user interface to manage the Axent ESM environment. The console has five privilege levels that allow the primary security administrator to create accounts that can clearly delineate the separation of duties between security administration and system administration. There is no need for the security administrator to have superuser privileges on the agent servers for Axent ESM to function. And, system administrators can be denied the privilege of altering security policy templates. Accounts can also be created with read-only report generation privileges. There are two levels of password protection on the consoles. One allows access to the console and the other allows access to the actual server security data. The agent component is installed on all managed servers and is used to gather information based on the specific security policy checks. There is a scheduler feature that allows for regularly scheduled policy checks on any basis from hourly up to yearly. Managers can be UNIX, Windows NT or Windows 2000. UNIX managers will support Windows agents and vice versa. The communications between the components is TCP protocol on ports 5600 and 5601. When there are firewalls separating components, additional ports may need to be opened. All communications between components is encrypted. The encryption is a 128 bit key, Diffie-Hellman key agreement scheme. The following diagram depicts the Axent ESM architecture:



The diagram depicts a single manager and console. However, in a large enterprise environment, it would be common to have multiple consoles and possibly multiple managers handling hundreds of agents spread across many Axent ESM domains. Since most large enterprise environments utilize DNS it is important that any managers and all agents are registered with DNS. System resources such as CPU and memory are an important consideration, especially on the agents. Security policies that run against the agents can be very CPU intensive; even though the manufacture states that ESM runs at idle priority (this means that the operating system gives them CPU time only when other threads and processes are waiting for I/O etc.). I found that systems with many users, or systems with less robust memory and I/O configurations, were impacted by the policy runs. Minimum configuration requirements are available online from the vendor. I found the installation of the various components to be a very simple task.

Structuring Agent Domains

Large companies today consist of many and varied organizations that can be spread across the globe. The IT infrastructure that supports large companies will be equally varied. And the types of vulnerability exposure also will vary. As stated in SANS Kickstart regarding Data Classification, " In corporate environments, classification is more loosely controlled and, in some cases, not applied at all. Information in commercial applications needs to flow much more freely, and the people using them are often more concerned with getting information quickly than in protecting it properly." Therefore, in a large enterprise it is helpful to be able to group network resources in a way that server security policies can not only be uniquely built for the varied platforms and applications, but also centrally managed. It is my experience (I help manage the security of over 400 servers in several different countries) that Axent ESM is a useful tool in this type of environment. Axent ESM applies policies to domains in order to assess the security of a computing environment. Policies represent the standards established by company security and administrative personnel and the checks derived from those standards. Domains contain the groups of agents that apply the policy checks.

Some suggestions for types of grouping are:

- **Organization.** Accounting systems in one domain, personnel systems in another domain and production in a third domain.
- **Function.** If systems have dedicated functions such as network servers in one domain and network printers in another.
- **Physical Location.** Physical location quite often affects the type of policy to be run.
- **Security/Administrator Responsibility.** If systems in an organization have been assigned to specific security or system administrators you can group the systems for each administrator into a separate domain.

Building Server Security Policy Templates

In general a security policy establishes what must be done to protect information stored on computers. A security policy should cover the three basic goals of confidentiality, integrity and availability. Based on the general policy in the area of data integrity an Axent security policy template can be built that reflects the level of protection required for a server or a set of servers in a domain. Since Axent ESM acts on a snapshot of a supposedly hardened server, it is critical to start with a sound system security hardening process. Lawrence Livermore National Laboratories suggests that the primary elements are:

1. Proactive installation of all security related patches.
2. Minimization – the removal or disablement of all non-essential features or capabilities.
3. Configuration – the secure configuration of
   a) Security accounts; including the password, groups, and other privileges
   b) Security programs: including elevation of privilege by the program, and access to secured resources.
   c) Non-secured programs: validation these programs have no influence over the security configurations or capabilities.

After you have defined and implemented your server security policy, Axent ESM can provide a continuous verification of the configuration.

Axent ESM categorizes its policy templates into three general categories: User Accounts and Authorizations/Network and Server Settings/File Systems and Directories. There are eighteen vendor-supplied modules with checks that are supposedly designed to detect the most common server vulnerabilities for UNIX, NT Server and Windows 2000. The default modules can be run at varying modes that will do lesser or greater checking. It is my recommendation that these default modules only be used as a basis for building policy modules that truly suit the environment in which the server functions. This is where I feel Axent ESM can be somewhat cumbersome to work with. The checks within the modules use template files that, even with the policy editor feature and the User Manual instructions, can be difficult and very time consuming to properly configure. Once an acceptable policy is created an initial policy run is initiated, and a snapshot of the agent server is recorded in ESM database files on the agent. Subsequent policy runs compare the current state of the agent against the snapshot database files as well as the policy template files. Any deviations are reported as possible vulnerabilities and assigned a rating based on the perceived severity of the vulnerability. An overall server rating supposedly gives you a gauge of the server's security.

Now let's examine the UNIX modules and how they can help secure a UNIX server against some of the common UNIX system vulnerabilities.

User Accounts and Authorizations:

Account Integrity Module – Identifies account privileges that differ from the established policy. It examines the password file and records the initial user and group ids. A common but poor practice of system administrators is to create multiple superuser accounts. This module will report if there is more than one user with a uid of 0. This module will also report on any illegal user shell.

Login Parameters Module – Checks for failed logins and dormant accounts and expired passwords. Unused accounts can create an area of vulnerability. Compromising one of these accounts can give a hacker access to a system that may go undetected for a long time. Hackers view unused accounts as a place to hide.

Password Strength Module – Checks that passwords conform to format and length as well as checks for expired passwords. Given that passwords are the first and sometimes only line of defense against interactive attacks on a system, a good password checking mechanism is an essential tool. Axent ESM uses a dictionary attack method. There are many default dictionaries to start from, you can build your own or download from the web. While this module is good for eliminating the common habit of the ordinary user to have password equal username or something equally weak, for critical ids such as root, a tool such as Crack7 that utilizes hybrid attack and can run in brute force mode should be run occasionally. This module, if configured poorly, becomes extremely computational intensive and will impact system performance.

User Files Module – Examines user logon scripts and home directories for proper access control and ensures that file ownership and permissions match the original baseline snapshot. This module will identify any files with setuid or setgid permissions. These permissions on a program allow a user to get the privileges (effective id) of the specified user or group for the execution of the program. A hacker can replace a setuid program with one that has the same functionality but may hide a back door into the system. Axent ESM does report every setuid/setgid as a severe vulnerability regardless of whether the id is root or an ordinary user. This module does an excellent job of examining user home directories for suspicious files, file names matching system commands, hidden directories, special device files and mount points. I have found it especially useful for uncovering .rhost, host.equiv and .netrc files that create insecure host trusts. Putting a plus sign in a .rhost or host.equiv file can allow hackers to gain entry if they can obtain a trusted id. This module will report this as a severe vulnerability.

Network and Server Settings:

Network Integrity Module – Examines Internet connections, Network File System (NFS) software, trust relationships and other network software. There are a number of NFS configuration errors that can allow unauthorized access to exported file systems. Often files are exported without restriction. If a file system is exported with root access to one system, this implies that the file system is exported without root access to all other systems. NFS access options must be properly used to limit the scope of NFS exports. This module displays all NFS exports for examination of proper access control.

Object Integrity Module – Identifies changes in ownership, permission, and other software objects for device-specific files in the system device directory or account. Insufficient permissions on device files such as /dev/mem or /dev/kmem can open the entire system up to a hacker.

System Queues Module – Checks cron and at utilities. There are two ways in which hackers try to exploit the cron functionality to obtain unauthorized root privilege. The first is to gain write access to any directory leading to root's crontab file. If a hacker gains access to root crontab file, they can enter a privileged command in the file and have that command execute by root at some time in the future. A second method is to analyze the contents of root's cron file to try and find writable files. If root invokes a program from within cron that is world writable, then any user can modify that program to include a privileged command that might create a backdoor for later use. This module points out all world writable files in cron, as well as displaying the cron.allow and at.allow contents.

Startup Files Module – Examines the contents of the UNIX "rc" startup scripts and examines inetd.conf for all installed services on the system. Similar to the previously mentioned cron utility vulnerability, malicious programs can be launched at startup if scripts called from startup files are not properly protected. The inetd daemon is one of the most important network service daemons on a UNIX system. Starting unnecessary services or ignoring unidentifiable services is inviting disaster. This is especially true of the Berkeley "r" services, which can allow access without password authentication. This module enables the administrator to create a template of the allowed startup services and the options to be used when they are invoked.

File Systems and Directories:

File Access Module – Examines the permissions of user-specified files and identifies the user accounts allowed to access the files as allowed by the overall security policy. This module acts as an access control list (ACL) in that the default list such as passwd, group and host can be built upon to add any file to the list and designate the users with access. It is critical to note that this is not real-time prevention from access. It will only report deviations from the original policy access list.

File Attributes Module – Indicates specifically chosen changes in selected system file attributes, including read, write and create privileges. It also performs and records a CRC checksum of the specified files. A checksum is a number computed from the binary bytes of the file that can be used to determine whether a file's contents are correct. Checksums are a very difficult file attribute for a hacker to circumvent. It should be noted that this option can be very I/O intensive and much consideration should be given as to when to run it. This is still not a foolproof method of guaranteeing file integrity since there are known viruses that will intercept directory listings and checksum commands and return supposedly legitimate values. Storing a "known good" copy of a checksum utility outside the system will give you greater assurance if you are doing system forensics after a system hack has been detected.

File Find Module – Checks for world-writable files, improper device files and "sticky bit" files. There are few good reasons for a file to be world-writable. Yet, inexperienced administrators often make directories wide open during system trouble shooting or trying to get a script to run, and sometimes will leave them that way. This module will report every world-writable file on the system. It will also report any device special files outside of the normal device directories such as /dev/ or /devices. The "sticky bit", which is represented by a t in the other execute access slot in a long listing, traditionally was used to keep a program in memory after it completes, in preparation for its next invocation. However, when used on a directory, it will allow any user to create files in a directory but not allow other users to remove those files. Although this does create flexibility for directories such as /tmp, it can also create excellent hiding places for hacker code. This module reports all files with the "t" bit set.

When designing a solid defense-in-depth security model, one that will protect the network, host, application and data files, a host-based server-scanning tool can be very effective piece of the host security layer. There are a variety of tools such as Tripwire, lsof, Crack and TCP Wrapper that will do as well, and in some instances better and certainly cheaper, than Axent ESM. Axent ESM is not a real time application. And, it should be noted that this is only one piece of good sever security. Any discussion of UNIX server security must mention the importance of system log analysis. While Axent ESM may help protect the integrity of UNIX system logs, it does not analyze them. However, when you need central management control of the server security policies of a large global enterprise, with a broad range of server platforms and server functions. I found Axent ESM does do a good job in this type of environment.

References

1. The Lawrence Livermore National Laboratory "Recommended UNIX Security Configuration" URL: http://www.lbl.gov/ITSD/CIS/UNIX/security/
2. Sandra Henry-Stocker "Building blocks to security: Passwords -- the first line of defense" May 2001. URL:http://www.itworld.com/AppDev/1313/UIR010509buildingblocks/
3. Mo Budlong "Security Basics, Part 1 – Understanding File Attribute Bits and Modes" October 2000. URL: http://www.itworld.com/Comp/3380/swol-1020-unix101/
4. Listing of UNIX server scanners at Purdue University URL: ftp://ftp.cerias.purdue.edu/pub/tools/unix/scanners/
5. Pipken, Donald L. "Halting the Hacker – A Practical guide to Computer Security" 1st Edition 1997
6. Frisch, Aeleen "Essential System Administration" 2nd Edition December 1995