Computer Security for FIO clusters

Overview

This document provides an overview of the general concepts for host based firewalls and intrusion detection systems, as well as many of the software packages I have recently investigated for use in the CERN IT-FIO environment. This should provide the reader with a understanding of the concepts and issues which will be presented in the ELFms session on Tue 28th June 2005.

Firewalls and Network Security

Firewalls are a useful method of reducing (but not eliminating!) the risk of intrusion by restricting or at least regulating access to a resource. This is typically done using a rule set which can include specifying the allowed destination and source ports/services, allowed IP addresses and various other options [http://www.netfilter.org/]. Firewalls can also help us to monitor information about what data flows in to, and out of, our machines in terms of which ports are used and which machines are connected to or from.

Unfortunately providing a firewall on the network e.g. at the router, makes it difficult to provide machine specific rules and as a result is very limiting for dynamic and varied environment. Although host based systems require more configuration and maintenance, in terms of system administrator time, they provide a much greater degree of flexibility in tailoring their rule set to the specific machine.

IPTables

IPTables is part of the Netfilter suite of software which is included in the Linux kernel 2.4.x and 2.6.x releases. IPTables evolved out of the IPChains package and is backward compatible with the IPChains rules. IPTables provides stateless packet filtering for IPv4 and IPv6 and stateful packet filtering for IPv4. This means that all packets which are received by the system can be filtered by the kernel before they reach the application layer. Likewise data which is being send out of the system or forwarded to other machines can be accepted or blocked based on the defined rules. This is analogous to having a moat around your system which can have multiple bridges over it which reduces the total number of entry points you have to monitor.

The full capabilities of IPTables are significant and a useful tutorial is available at [http://iptables-tutorial.frozentux.net/iptables-tutorial.html].

What can IPTables do? Iptables allows you to accept or block communications to or from specific ports and IP addresses. For example, you can create rules which allow you to connect using SSH from any host, but restrict which machines can access other ports and services to, for example, machines in your cluster. You can also restrict access based on the protocol being used, so for SSH you would allow TCP connections but ignore UDP and ICMP packets.

What can't IPTables do? IPTables **will not** solve fundamental problems in your services. If you are allowing unlimited access to SSH and someone has got a valid username / password then this system is not going to stop them gaining access. If there is a buffer overflow problem with a service IPTables will not stop this. The functionality of IPTables will be discussed in more detail in the ELFms presentation.

The NCM Component which we are using to configure IPTables at the moment was written by Joao Paulo Martins [martinsj@lip.pt] and is currently being maintained by Alasdair Earl [alasdair.earl@cern.ch]. With this component the order of the rules in the node profile does not matter as they are rearranged to conform to a standard format of Log rules, Accept rules, and Drop rules. Logging functionality is fairly basic at the moment and this is the first item on our todo list with the component. Logging and monitoring issues will be addressed further in a later section.

Snort

Snort [www.snort.org] is a network intrusion detection system which monitors packets entering or exiting the network interface and compares what it sees to a pattern database of previous attacks. It is very similar to IPTables in using rules based on various variables to trigger actions. Unlike IPTables, the range of options available to Snort is larger and it makes greater use of the concept of state – attacks have been known where sending a set number of packets to a machine using a specific sequence of ports triggers a root kit – IPTables would not be able to detect this but Snort would.

Snort has several methods of reporting attacks to the sysadmin including logs, emails and Windows pop-ups. These patterns are simple to write and are normally available shortly after new attacks are discovered. There seems to be an active community of developers and users.

Snort was developed to solve the problem traditional NIDSs have in terms of high overhead costs in terms of purchase, system administrator time and resource usage. Snort was designed for small, lightly used networks. It has also been run at CERN by the Computer Security team to investigate attacks. At the moment we don't see a high value in deploying Snort on our nodes.

Intrusion Detection Systems

Intrusion detection systems can take several different forms including systems to monitor file, network and kernel changes and intrusions. The type we are interested in at the moment - although we may extend this in the future - are file integrity checkers. These take a snapshot of the system at a specific point, typically immediately after build when we are sure attackers have not tampered with it, by creating a database of the hash values of files. The files monitored are typically executables and library files which do not change frequently. Obviously it would be preferable to do this before the system is connected to the Internet but in the case of our clusters this is logistically difficult.

Tripwire

Tripwire is an early IDS (1990's) aimed at single host/processor systems. The developers have now developed a commercial version with considerably more functionality and network support. Currently there seems to be little development work being done on the open source version but this can be taken to be a testament to the simplicity and robustness of the existing software.

We installed Tripwire on a SLC3 machine. Apart from having to reconfigure the gcc3 variables which involved rebuilding the RPM, the installation went smoothly. We then looked into developing an NCM component for the installation. Tripwire installations require passwords for the site files (configuration and policy) and local system (database and reports). The site files can be reused between machines but the local ones should be unique. Theoretically we can deploy this system by building the passwords into the Tripwire RPM **HOWEVER** this is a very, very bad idea from a security stand point. Work being done by Marc Poulhies on the secure distribution of passwords may solve this problem.

Reports can be kept on the local system or email to the administrator. From experience at other sites systems which have multiple users and services tend to produce a large number of reports which can swamp administrators if not careful.

AIDE

AIDE is an open source project which is attempting to provide an alternative to Tripwire. AIDE aims to reproduce the functionality of Tripwire, including format of the profile, to make migration easier. At the moment it does not offer the full range of functionality that Tripwire does and as always with small software projects there is a concern about long term support. Currently the documentation and general level of support available for AIDE is less than for Tripwire.

Our experience with installing AIDE on a single system is that it is perfectly adequate for a home or small installation. It does not currently come in RPM form so we would need to develop this. Like Tripwire it is intended for a single machine. Unlike Tripwire the database is not an encrypted file and the developers recommend that the configuration file and database be copied to read only media as they are created. Because of this it doesn't scale to the levels we need and we do not intend to pursue it.

Snare

Snare is an intrusion detection system which uses the logs produced by the operating system and various applications : web servers, firewalls, routers; to provide an audit of the system. The documentation states that it can potentially support central logging but we have not tested this. Unfortunately Snare requires its software to be built into the kernel. This is an obvious disadvantage for testing and from the documentation we think that unless we planned to deploy it on web servers it is not necessarily the most ideal solution for our needs.

Samhain

Samhain is a more advanced IDS in terms of the scope of its functionality. It is released under an open source license and appears to have an active development community with a fair amount of documentation. Like Tripwire and AIDE it creates a database of file hash values but unlike the other packages this database can be stored in a remote DBMS including Mysql, Postgres, and Oracle. This allows us to manage information from multiple nodes at a single point. However, Samhain does support the update of profiles at the central node so an upgrade to say, SSH, could be registered and the change not cause alarm messages from every node in the system.

At the moment we can only find information about Samhain scaling up to a few hundred systems so we are investigating how we can support multiple clusters and larger scale installations.

We are currently working on developing an NCM Component for nodes which will link into a central database. This is still in the alpha phase of development.

Monitoring

Having these security systems in place is obviously useful, but is only the first step in improving security as it does not address the issue of monitoring. This is necessary to see whether we have been attacked and if the intruder has succeeded. To add monitoring support we looked at several different systems to evaluate which is appropriate for our needs and has scalability and long term support.

Prelude

Prelude is a management and reporting system which is able to receive reports from various firewall and intrusion detection systems, including Samhain and Snort. It is adaptable, simple to write interfaces to and provides a web interface for the administrator.

Beltane

Beltane is a monitoring system developed for Samhain. It provides a web interface to the Samhain database (Yule server) allowing the administrator to review the logs from all nodes from a single point and to update profiles.

Lemon

The LHC ERa Monitoring (Lemon) is the default monitoring system for alarms and summarise log information. We are currently developing a Lemon sensor for reporting the IPTables output from log files. This is specific to the node at present.

Conclusions

The rule sets needed for host based firewalls are typically specific to the node or cluster in question. We have an adaptable and extensible component for configuring IPTables which is in production usage and that can be used to do this now. Some additional work is needed on this component to enable more advanced usage such as monitoring and logging but this is relatively minor in terms of development time.

The issue of installing an intrusion detection system is more complex than IPTables, due to the need for secure passwords and configuration files being transferred over the network. However the rule set, or profile, for nodes once the initial node is installed is fairly standard and requires less communication and feedback from users. Because of this we expect to have started deploying Samhain to at least one cluster in the near future.

We claim that the intrusion detection system is less of an administrative issue than IPTables is for certain clusters, for example Lxplus. Lxplus will be difficult to secure with IPTables given the amount of user software, open ports and IP address ranges which must be accessible. Samhain, or any other IDS, in contrast should only be concerned with the system software, which is standard, and therefore it does not matter whether the system has one user or one thousand, the system software will still be the same.

Monitoring and alerts are going to be the biggest administrative issues for the security system both from a design and administration viewpoint. IPTables reports can be monitored by the existing security infrastructure as they use the standard system logfiles and syslog for reporting. The intrusion detection system offers more choice as the database of monitoring information which means that we can use multiple front end systems if necessary. This also means that it is more important for us to chose the right IDS than monitoring system as changing the latter is a far easier and less time intensive task.

We hope that the presentation which this document leads into will generate discussion on what information we need to ensure the system is capable of collecting and what we can safely ignore from people who have experience in this area.