Intrusion Detection, Theory and Practice

by David "Del" Elson

last updated Monday, March 27, 2000

Relevant Links

AIDE
Rami Lehti

Ethereal
Gerald Combs

FCheck
Michael A. Gumienny

ISS RealSecure Engine
Internet Security Systems

LIDS
Xie Hua Gang

NFR
Network Flight Recorder

OpenWall
Openwall Project

Psionic Software
Psionic Software, Inc.

Tripwire
Tripwire, Inc.

Introduction

Network security has been an issue almost since computers have been networked together. Since the evolution of the internet, there has been an increasing need for security systems. One important type of security software that has emerged since the evolution of the internet is intrusion detection systems.

This article gives an overview of several types of intrusion detection systems, and introduces the reader to some of the concepts and practices involved in intrusion detection. Be aware that this article is only introductory, and while I have suggested a number of possible systems, further research should always be undertaken before trusting in the strength of your intrusion detection system.

What is Intrusion Detection?

Intrusion detection, is the attempt to monitor and possibly prevent attempts to intrude into or otherwise compromise your system and network resources.

Simply put, it works like this: You have a computer system. It is attached to a network, and perhaps even to the internet. You are willing to allow access to that computer system from the network, by authorised people, for acceptable reasons. For example, you have a web server, attached to the internet, and you are willing to allow your clients, staff, and potential clients, to access the web pages stored on that web server.

You are not, however, willing to allow unauthorised access to that system by anyone, be that staff, customers, or unknown third parties. For example, you do not want people (other than the web designers that your company has employed) to be able to change the web pages on that computer. Typically, a firewall or authentication system of some kind will be employed to prevent unauthorised access.

Sometimes, however, simple firewalling or authentication systems can be broken. Intrusion detection is the set of mechanisms that you put in place to warn of attempted unauthorised access to the computer. Intrusion detection systems can also take some steps to deny access to would-be intruders.

Why use Intrusion Detection?

The underlying reasons why you might use intrusion detection systems are relatively straightforward: You want to protect your data and systems integrity. The fact that you cannot always protect that data integrity from outside intruders in today's internet environment using mechanisms such as ordinary password and file security, leads to a range of issues.

Adequate system security is of course the first step in ensuring data protection. For example, it is pointless to attach a system directly to the internet and hope that nobody breaks into it, if it has no administrator password! Similarly, it is important that the system prevents access to critical files or authentication databases (such as the NT SAM or the Unix /etc/passwd or /etc/shadow files) except by authorised

systems administrators.

Further measures beyond those normally expected of an intranet system should always be made on any system connected to the internet. Firewalling and other access prevention mechanisms should always be put in place. While it may be acceptable to allow NT logon, file sharing, or telnet access to a system that is entirely internal, an internet server should always use more secure mechanisms, such as firewalling off the NT file sharing (SMB protocol) ports such as TCP/UDP ports 137 - 139, and using secure shell (SSH) instead of telnet for access to Unix systems.

Intrusion detection takes that one step further. Placed between the firewall and the system being secured, a network based intrusion detection system can provide an extra layer of protection to that system. For example, monitoring access from the internet to the sensitive data ports of the secured system can determine whether the firewall has perhaps been compromised, or whether an unknown mechanism has been used to bypass the security mechanisms of the firewall to access the network being protected.

What types of Intrusion Detection systems are there?

Intrusion Detection systems fall into two broad categories. These are:

- Network based systems. These types of systems are placed on the network, nearby the system or systems being monitored. They examine the network traffic and determine whether it falls within acceptable boundaries.
- Host based systems. These types of systems actually run on the system being monitored. These examine the system to determine whether the activity on the system is acceptable.

I will also discuss briefly a more recent type of intrusion detection system: Those that reside in the operating system kernel and monitor activity at the lowest level of the system. These systems have recently started becoming available for a few platforms, and are relatively platform specific.

Network Based Intrusion Detection

Introduction

Network based intrusion detection systems are those that monitor traffic on the entire network segment. A network interface card (NIC) can operate in one of two modes, these being:

- Normal mode, where packets which are destined for the computer (as determined by the ethernet or MAC address of the packet) are relayed through to the host system.
- Promiscuous mode, where all packets that are seen on the ethernet are relayed to the host system.

A network card can normally be switched from normal mode to promiscuous mode, and vice-versa, by using a low-level function of the operating system to talk directly to the network card to make that change. Network based intrusion detection systems normally require that a network interface card is in promiscuous mode.

Packet Sniffers and Network Monitors

Packet Sniffers and Network Monitors were originally designed to aid in the process of monitoring the traffic on an Ethernet network. The first of these were two products; Novell LANalyser and Microsoft Network Monitor.

These products basically capture all packets that they see on the network. Once the packets are captured, a number of possibilities arise:

- Packets can be counted. Counting the packets that come past, and adding together their total size over a period of time (including overheads such as packet headers) gives a pretty good indication of how heavily loaded the network is. Both LANalyser and Microsoft Network Monitor provide load graphs or meters to show the relative load of the network.
- Packets can be examined in detail. For example, you might want to capture a set of packets arriving at a web server to diagnose some problem with the server.

Packet sniffing products have become more sophisticated in recent years. Programs such as Ethereal and more recent versions of Network Monitor can disassemble the insides of various types of packets to show what type of communication is happening inside that packet.

One final word about packet sniffers: These tools can be used to do evil as well as good. For example, packet sniffing can be used to find out someone's Unix password by sniffing telnet packets to the machine that they connect to. Once an attacker has compromised your network, one of the first things they might install is a packet sniffer of some kind.

Packet Sniffing and Promiscuous Mode

All packet sniffers will require that a network interface is in promiscuous mode. Only in promiscuous mode will every packet received by the NIC be passed up to the packet sniffer itself. The packet sniffer normally requires administrative privileges on the machine being used as a packet sniffer, so that the hardware of the network card can be manipulated to be in promiscuous mode.

Another point to consider is the use of switches, rather than hubs, in a network. Note that packets received on one interface of a switch are not always sent to other interfaces of the switch. For this reason, a heavily switched environment, rather than an all-hubs (single segment) environment, will often defeat the use of packet sniffers.

Network Based Intrusion Detection: The Evolution of the Packet Sniffer

Unfortunately, from a security point of view, a packet sniffer is of limited benefit. The task of capturing every packet on the network, disassembling it, and manually taking action based on the contents of the packet is far too time-consuming, even for a horde of specially trained network gnomes. What if we were to have some software that automated the process for us (after all, that is what computers are for in the first place, is it not?).

This is basically exactly what a network based intrusion detection package does. Two packages that are available that perform this type of intrusion detection are the ISS RealSecure Engine and Network Flight Recorder.

Here is an example of some of the types of intrusion detection that the RealSecure Engine can perform:

- Examine packets that pass through the network.
- For legitimate packets, allow them to pass (perhaps recording them for future analysis).
- Where a packet endangers the security or integrity of a target system, stop transmission of the packet by sending TCP "connection closed" or ICMP "port unreachable" messages to both the target system and the system sending the packet.

In this manner, RealSecure can perform an effective second layer defence for a target system when hosted behind a firewall. Some implementations have also used RealSecure in place of a firewall, which is, however, something that I do not recommend.

Network based intrusion detection can perform a few other tasks, for example:

- Monitoring the network for obvious port scans. Before compromising a system, a cracker will often port scan the system to determine what vulnerabilities might exist. Port scan attempts from a host on the internet can often be a signal that a person on such a host intends to damage your network.
- Monitor valid connections for well known attacks. Accessing a web server host on the web server port (80) might be seen as a relatively harmless activity, but some access attempts are in fact deliberate attacks, or attempts at attacks. For example, an access that looks like "GET /../../../etc/passwd HTTP/1.0" is probably a bad sign, and should be blocked.
- Identify IP spoofing attempts of various sorts. The ARP protocol that is used to convert IP addresses to MAC addresses is often a target for attack. By sending forged ARP packets over an ethernet, an intruder who has obtained access to one system can also "pretend" to be operating as a different system. This can lead to denial of service attacks of various sorts, as well as system hijacking, whereby an important server (such as a DNS server or authentication server) is "spoofed". Crackers can use this "spoofing" to redirect packets to their own system, and perform "man in the middle" type attacks on what would otherwise be a secure network. By keeping a register of ARP packets, a network based intrusion detection system can identify the source (ethernet address) of a compromised system and flush out would-be crackers.

When unwanted activity is detected, network based intrusion detection can take action, including interfering with future traffic from the intruder, or reconfiguring a nearby firewall to block all traffic coming from the intruder's computer or network.

Host Based Intrusion Detection

Introduction

Once a network packet has arrived at the host that it was intended for, there is still available a third line of defence behind the firewall and network monitor. This is called "host based intrusion detection", and comes in several flavours.

The two main types of host based intrusion detection are:

- Network monitors. These monitor incoming network connections to the host, and attempt to determine whether any of these connections represent a threat. Network connections that represent some kind of intrusion attempt are acted on. Note that this is different to network based intrusion detection, as it only looks at network traffic coming to the host it is running on, and not all traffic passing the network. For this reason it does not require promiscuous mode on the network interface.
- Host monitors. These monitor files, file systems, logs, or other parts of the host itself to look for particular types of suspicious activity that might represent an intrusion attempt (or a successful intrusion). Systems administration staff can then be notified about any problems that are found.

Monitoring Incoming Connections

It is possible on most hosts to monitor packets that attempt to access the host before those packets are passed onto the networking layer of the host itself. This mechanism attempts to protect a host by intercepting packets that arrive for the host before they can do any damage.

Some of the actions that can be taken include:

- Detect incoming connection attempts to TCP or UDP ports that are unauthorised, such as attempts to connect to ports where there are no services. This is often indicative of a possible cracker having a "poke around" to find weaknesses.
- Detect incoming portscans. Again, this is a definite issue that should be addressed, and alerting a firewall or modifying the local IP configuration to deny access from a possible intruder host (eg: by using ipchains on Linux) is one action to take.

Two software products that perform this type of monitoring include the RealSecure Agent by ISS, and PortSentry.

Monitoring Login Activity

Despite the best efforts of the network administrator, and the most recently deployed and monitored intrusion detection software, occasionally an intruder will slip by and manage to log on to a system using a heretofore unknown type of attack. Perhaps an attacker will have obtained a network password by some means (packet sniffing or otherwise) and now has the ability to log on to the system remotely.

Looking for unusual activity on a system is a job for a product such as HostSentry. This type of package monitors log-in and log-out attempts, and alerts the system administrator to activity that is unusual or unexpected.

Monitoring Root Activity

The aim of all intruders is to obtain super-user (root) or administrator access on the system that they have compromised. Well-maintained and reliable systems that are used as web servers and databases will usually have little or no activity by the super-user, except at particular times of the day or night for scheduled system maintenance. Fortunately, crackers do not believe in system maintenance. They rarely stick to scheduled downtime windows and often work at odd hours of the day. They perform activities on the system that are unusual for even the most propeller-

headed system administrator.

There is one more line of defence: Monitoring any actions performed by the root user or system administrator. Many unix systems allow logging or other monitoring of all activity by the root user, and packages such as Logcheck can then scan these logs for unusual activity and notify others about it.

Administrators of open-source operating systems have one final option: alter the kernel to log specific types of activity. How to do this is outside of the scope of this article, however there are a number of sites on the internet that detail how this can be done.

Monitoring the File Systems

Once an intruder has compromised a system (and despite your best hopes, and the best efforts of the intrusion detection systems that you have laid down, one cannot discount entirely the possibility that one day an intruder will compromise a system), then they will start to change files on the system. For example, a successful intruder might want to install a packet sniffer or portscan detector, or modify some of the system files or programs to disable some of the intrusion detection methods that they have worked around.

Installing software on a system usually involves modifying some part of that system. These modifications will usually take the form of modifying files or libraries on the system.

Programs such as Tripwire, Fcheck, and AIDE are designed to detect when files change on the system, and alert the system administrator to any changes.

For example, the following approaches can be taken:

- Create MD5 or other cryptographic checksums of all of the system files, and store these in a database. When a file changes, its checksum will change.
- Store the creation or modification date and time of every system file. Look for any changes in these timestamps.
- Keep a record of any suid (run-as-root) program on the system. If any of these change, or if new ones are installed, or any are deleted, then there is a problem.

The methods used by Tripwire, Fcheck, and AIDE vary somewhat, but they are all based on the above mechanisms. These programs also take care to make sure that the database of known cryptographic checksums itself isn't compromised in any way. For example, it would be possible for an intruder with a knowledge of the operating system and the intrusion detection software to modify the system files and also modify the cryptographic checksum database so that it would appear that nothing had changed.

Kernel Based Intrusion Detection

Kernel based intrusion detection is a relatively new art form, and one that is starting to become prevalent, especially within Linux.

There are two main kernel based intrusion detection systems currently available for Linux. These are OpenWall and LIDS. These systems take the approach of preventing buffer overflows, increasing file system protection, blocking signals, and generally making it difficult for an attacker to compromise a system. LIDS also takes

steps to prevent certain actions by the root user, such as installing a packet sniffer or changing firewall rules.

Kernel Protection vs File System Monitoring

Obviously, systems like LIDS and systems such as Tripwire take a rather different approach to attempting to achieve the same thing. Both of these packages attempt to prevent a cracker from using the system for unauthorised purposes.

At first glance, one might think that a system such as Tripwire is less than perfectly useful. While it is a Good Thing to monitor file systems for signs of abuse, it is readily accepted that once your system has been compromised by an external intruder, it is time to shut down and rebuild. The damage has been done, and the system integrity cannot be guaranteed so it is best to re-build your operating system from the pristine version supplied on CD from your vendor. The approach offered by LIDS seems much more attractive - to protect the system from damage rather than to note that the gate was left open after the horse has bolted, so to speak.

Although I tend to agree with this analysis in principle, there is a stronger level of security offered by running both LIDS and Tripwire together. While LIDS is exceptional in its capability to protect the file system, it is worth using a file system monitoring package such as Tripwire as an "independent" auditor, in the event that an exceptionally knowledgeable hacker should manage to somehow defeat LIDS.

Conclusions

It is possible, using the most up to date tools that are available, to protect against virtually every type of threat that is currently known about. Unfortunately, new threats and security holes in some software package or another are being discovered on a daily basis.

It is important in any environment to know what types of threats you might be facing. Be aware of any potential security holes in your system, and take care to prevent attacks against these. For example, a web server that is connected to the internet and placed behind a firewall may be reasonably secure against most packet based attacks, but a CGI program on the server might expose a vulnerability. Pay special attention to ensuring that CGI programs correctly bounds check all arrays and validates input data before processing. An intrusion detection program between the firewall and the web server might configured to throw out any accesses that are suspicious.

Staying Up To Date

Most of the intrusion detection tools that I have mentioned are regularly updated to include information about new threats as they are discovered, however it is important to keep up to date with the latest version of these tools.

Watching certain mailing lists (such as BUGTRAQ) and security web sites can help you stay informed about the latest security issues affecting software that you have installed. If you are alerted of a vulnerability in a software package that you are using, or in a firewall product, or perhaps even in an intrusion detection software package itself, then don't be shy about contacting the vendor for a fix.

Which Tools?

I have discussed a number of different types of tools in this paper, all with different

functions.

To keep your environments as secure as possible, it is important to choose tools from across the range of functions. Each of the tools forms an extra layer of protection in case the others are defeated. So, your first layer of protection should be a firewall. Behind that, a network based intrusion detection system will catch any breaches of the firewall. Behind that again, a set of tools that monitor connection attempts, such as PortSentry and HostSentry form an extra layer of protection. Finally, if all else fails, tools to catch an actual break-in, such as LogCheck or Tripwire, form the final layer of defense.

David Elson (Del) is a security and technology consultant working for Wang New Zealand in Christchurch, on the South Island of New Zealand. With 15 years IT experience, he consults to various clients on security and networking issues. He also maintains a set of web pages on Linux and other related security topics, and has given talks on various security and networking issues at conferences in Australia and New Zealand.