Intrusion Detection and Intrusion Prevention on a Large Network: A Case Study

Tom Dunigan and Greg Hinkel
Oak Ridge National Laboratory

Intrusion Detection and Intrusion Prevention on a Large Network. A Case Study.

Tom Dunigan, Network Research
Oak Ridge National Laboratory
Greg Hinkel, Computer & Network Security
Oak Ridge National Laboratory

Abstract

This paper describes the general requirements for an Intrusion Prevention and Detection System and the methods used to prevent and detect intrusions into Oak Ridge National Laboratory's network. In this paper we describe actual intrusions, how they were detected, and how they were handled. We also describe the monitoring tools we use for detecting intrusions.

Introduction

At Oak Ridge National Laboratory (ORNL), we have an open environment in which researchers around the world must collaborate with ORNL researchers. These users want and need easy access to each other's data, programs, and correspondence. Furthermore, many of the researchers have been accustomed to unfettered access to and from the Internet. Obviously, we also have data that should not be available to external users.

Our network consists of approximately 18,000 computers running a variety of operating systems, including UNIX, VMS, Windows, and MacOS. Our users abilities range from "untrained" desktop users to highly trained supercomputer programmers.

An open environment like ORNL's poses many security concerns. The dynamic nature of the work performed at ORNL introduces additional security concerns in that new project initiatives, with new users and new computers, begin almost daily. These new projects often create sudden increases in network activity from new and different computer systems, and the sudden increases make it difficult to weed out "new project" traffic from intrusion attempts. Also, many of our "users" are not physically located at ORNL. Trying to determine if a remote user is the "legitimate user" is not an easy task. The question, "Was login information sniffed by a hacker who is now logging in?," is quite difficult to answer.

A security plan is essential. Knowing what to look for takes time, experience, diligence, and a lot of luck. Our plan needed to answer the following questions.

- What is the threat?
- What can happen if an intrusion occurs?
- What should we watch for?
- What should we report?
  - What should our intrusion detection system report to us?
  - Should we report intrusions to someone and if so, to whom?
- What should we do if and when we suspect an intrusion?

Intrusion prevention is our goal. However, it was clear that we would not be able to completely prevent intrusions, so we decided to:

- try to reduce the number of possible intrusions, and
- quickly detect any intrusions that did occur.

A simple solution to intrusion prevention and detection was not possible at ORNL. Trying to reduce the number of intrusions would have to be accomplished by providing secure mechanisms for end users to access their computer systems and then educating those users and their system administrators about the proper use of those secure mechanisms. Additional hardware and software would be required for intrusion detection. Detecting intrusions in real time is preferable and in isolated cases is possible. However, to reduce the likelihood of terminating a legitimate connection and to be more effective at detecting intrusions, it was clear that we would have to log and analyze users' activities. There are commercial packages that satisfy some of our requirements; however, none would satisfy all of our requirements. Therefore, we had to implement a specialized program that used commercial packages in conjunction with solutions developed in-house.

At ORNL, we use a layered approach to network security because multiple layers make penetration more difficult while making detection a bit easier. We define our layers as follows:

1. firewall for limiting access,
2. external monitoring for detecting attacks,
3. internal monitoring for detecting attacks and reducing vulnerabilities,
4. system administration for reducing vulnerabilities, and
5. end users for reducing vulnerabilities.

The security staff must be knowledgeable security professionals and they must:

- know what to look for (i.e., what kinds of attacks might occur?);
- know what they are seeing (i.e., what does the data "on the wire" or in the log files represent?);
- know (or have an idea) what to do if their computers come under attack and know who to contact for additional information or assistance; and
- educate the users and system administrators about computer and network security issues, and keep them informed of current attack methods and counter measures.

We think it is important for the security staff to have a good rapport with users. Users and system administrators should trust the security professionals and look to them for advice; users should be able to depend on the security professionals to keep them abreast of current attack methods and countermeasures.

At ORNL, we need fast data collecting machines that are tightly controlled, with all unneeded services turned off. Encrypted communication is the only means of entry into these machines (except for console access). Our security staff is trained to use these encrypted channels correctly.

We also need plenty of disk space for log files because we planned to keep at least one month's worth of data online.

Policy Decisions

Q. What is the threat?
A. We generally consider the users on our network to be "trusted." Our main concern is people outside our network trying to get into our network. Many of our users log in through their ISP (Internet Service Provider); from a conference floor; or from a remote network (e.g., at a collaborator's site) using insecure applications, such as telnet, ftp, or POP. Therefore, we have determined that our biggest threat is from authorized remote users who access our machines and have their login information sniffed at the remote site.

Our second greatest threat is misconfigured or unpatched systems. We have several users that cannot (or do not want to) spend time/money to ensure the integrity of their machines, or they do not understand the threat and importance of keeping their machines secured. Our computers have been "hacked" because of "misconfigured" or unpatched systems. However, our decision to use a commercial security package, Internet Security Scanner (ISS), and to develop customized tools for checking for network vulnerabilities, have significantly reduced our vulnerabilities.

An internal scan may show no vulnerabilities one day, but that is no assurance that a vulnerability will not be present the next day. For example, we scanned our address space for "named" service and notified appropriate system administrators of potential vulnerabilities. The day after that scan, one of our users rebooted his machine from Windows 95 to Linux, which was running an unpatched "named." That night our network was scanned by a remote site, and that machine was compromised via a buffer overflow in "named."

Q. What can happen if an intrusion occurs?
A. Possible problems for us include:

1. loss of data;
2. modification of data, which can be more serious than loss of data;
3. misuse of equipment;
4. loss of employee time and/or CPU time;
5. time spent assessing damage and cleaning up; and
6. embarrassment to the company/project/ individual.

Q. What should we watch for and what should our intrusion detection system report to us?
A. Because our biggest security threat is legitimate users having their login information sniffed at a remote site, we need to watch for unusual activity for each user. For example, if a user typically logs in from Knoxville and suddenly logs in from Peru, we need to be notified. Likewise, if a user typically uses a computer for editing, compiling, and running FORTRAN programs, and suddenly begins using IRC (Internet Relay Chat), we need to be notified. Following the activity patterns of users requires monitoring the commands they issue, which meant a network keystroke logger was needed.

Because port scanning is very popular and because we need to watch other network services (in addition to those that the keystroke logger picks up), it seemed prudent to detect incoming connection requests, which meant we also needed a "touch logger."

These monitors do not usually provide real-time notification, so we use a third party Intrusion Detection System (IDS) which does provide real-time notification. We also knew that there would be times when we would have to monitor specific services and/or hosts (for "special case" needs), so we added an additional machine for this purpose.

Q. What should we do if/when we suspect an intrusion?
A. Possible actions are:

1. remove compromised machine from the network,
2. setup additional monitoring,
3. deny access to effected machines and/or subnets,
4. deny access to specific users, and
5. notify essential personnel.

Q. Should we report intrusions (and attempts), and to whom should we report them?
A. We decided to report everything we determined to be "unfriendly activity." We report them to CIAC as well as to the registered administrator of the "attacking" subnet. We notify others as necessary, such as AUSCERT and EUROCERT. Our hope is that by notifying central security facilities, information about attacking sites could be disseminated to other "legitimate" facilities who could then watch more closely for activity from those sites. We have also received very favorable responses from ISPs, where, in many cases, accounts were terminated. Figure 1 shows the number of messages we sent to remote sites in 1998, as well as the number of intrusions that we had last year.

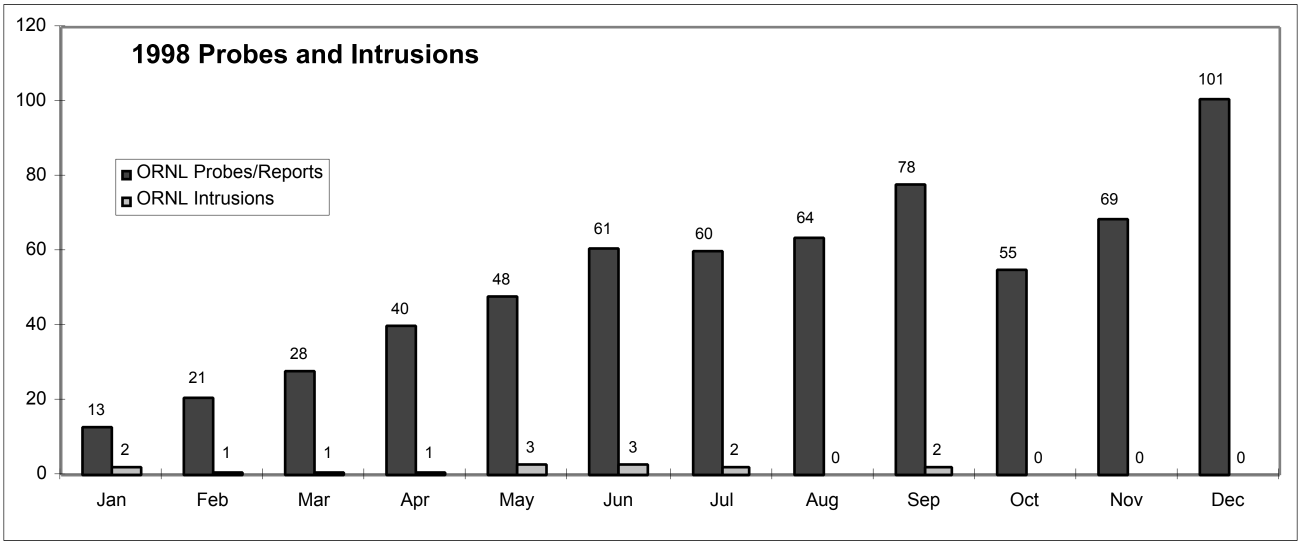
Hardware Configuration

At ORNL, we have several dedicated computer systems that collect network data coming from and going to external networks. They are all time-synchronized to ensure accurate "reconstruction" of each attack. Time synchronization is also necessary because with that information, personnel at the offending site can track the attacker much more easily.

These machines log successful and unsuccessful TCP connections, UDP packets, and user keystrokes with tools developed in-house. An additional computer logs selected connections and sessions with a third party IDS package. Another system assimilates the data, processes it, and generates human readable reports, which are mailed to security personnel several times a day. These reports show "interesting" traffic patterns, where "interesting" is defined as things that appear to be potentially unfriendly. A few examples follow:

- external machines connecting to TCP ports 111 (portmap) or 15 (netstat) or UDP port 31337 (the default for BackOrifice),
- external machines attempting to log into our central name servers,
- external hosts scanning internal hosts or ports,
- connections to/from local hosts that have recently been compromised,
- connections to/from hosts that have recently been the source of an attack, or
- hosts that generate the most network traffic.

Figure 1



We have additional computers that can be used for "special case" monitoring when needed. Such special cases would include monitoring all traffic going to/from a given host or subnet (in the case of a hacked system), or it may include collecting all data related to a particular port (in the case of an ongoing port scan). These machines also can be configured to provide real-time alerts for specific activities.

Logging all TCP connections, whether successful or not, helps in the detection of port scans and also is valuable in determining where an attacker comes from and goes after gaining access to one of our machines. We also use this information to determine which of our computers respond to particular services. When an attacker runs a port scan against our computers, we collect the responses so that we can follow up on them (hopefully before the attacker does).

Our most useful intrusion detection tool is our "keystroke logger," which records session keystrokes. It is virtually impossible to detect intrusions without this tool. Because our users log in from around the world at all times of the day and night and all have different tasks, it is not possible to determine authorized sessions from unauthorized sessions by simply looking at the connections. Also, attackers often "clean up" system log files, so relying on end users and system administrators is not sufficient. Our reviews of users' keystrokes often indicates hacker activity.

Additional computers can be placed on selected subnets for special-case monitoring or for "replacement" of a hacked system. We do this to learn "back door" accounts that the attacker may have planted, as well as to learn more about the attacker's methods. Case 3 below shows an example of a modified telnet daemon that appears, to the attacker, to allow access to a recently compromised system for a given user. In that case, we held the attackers attention long enough to contact his ISP and determine his location.

Implementation

At ORNL, we use a layered approach to network security because multiple layers make penetration more difficult while making detection a bit easier. Our layers, both physical and administrative, are defined as follows:

1. a firewall,
2. external monitoring,
3. internal monitoring,
4. system administration, and
5. end users.

Our layered approach follows:

1. **A Firewall.**
   A firewall is used to limit access to our network. Services, hosts, and subnets are selectively permitted or restricted from accessing our network. Occasionally, local hosts are restricted from accessing the Internet.

2. **External Monitoring.**
   Several computers, as mentioned previously in "Hardware Configuration," are set up to monitor traffic coming from and going to our network. These machines detect probes and intrusion attempts and alert security personnel of such events.

3. **Internal Monitoring.**
   We have several machines (honeypots) instrumented and alarmed to notify security personnel of particular events (e.g., a follow-up attack trying to exploit a vulnerability in a service that an attacker identified through a previous port scan). We also have our main WEB, mail, and DNS servers instrumented to notify security personnel of suspicious events, such as a remote user attempting to log into our name servers.

   Using ISS, we scan our network for computers with known security vulnerabilities, and then reports are sent to the system administrator. Custom scans are developed to detect other vulnerabilities where ISS is insufficient. For example, scans for the latest "named" exploit were developed and used. The administrators for all machines identified as running "named" were notified, and they either patched or disabled "named." In another case, a teardrop check was developed to allow users to test their machines and verify vulnerabilities to a teardrop attack. Review of the latest CIAC and CERT advisories, as well as frequent checks of hacker resources such as bugtraq, provide critical knowledge of the "newest" attacks. These advisories and hacker resources are often the stimulation for development of a custom scan.

   We also have a mechanism where we "bait" sniffer programs. The intent is to detect a sniffer program that may have been installed on a local machine but gone undetected. We inject packets onto the network that appear as a login session. Any sniffer should log the "session." We judiciously put different login "sessions" on various subnets, and we change them occasionally. If the attacker then tries to log in to one of the "baited" accounts, security personnel can determine where the attempted login came from but more importantly, can pinpoint on which subnet the sniffer was running and the time period it was running.

4. **System Administration.**
   Good system administration is encouraged. A "scan me" web page was developed to allow system administrators to easily run ISS against their machines to determine vulnerabilities so that they could be patched quickly. System administrators are informed of current attack methods and vulnerabilities. They are encouraged to run md5 checksums on system files, and a "central" repository was established for master checksum lists. We encourage the use of TCP Wrappers for host-based access and for logging. Default routes are not set unless necessary. System administrators review system log files, including syslog and web server logs, and they notify security personnel of unusual activity. Administrators are encouraged to "know their users" and "know their machines."

5. **End Users.**
   Ongoing efforts are made to educate the end users. They are instructed in selection of good passwords, on how clear text reusable passwords can be used against them, and about the importance of their account. A "central" web site was developed for security-related issues, making it easier for users (and system administrators) to locate necessary information and report unusual activity. We also provide guidelines for the "proper use" of .rhosts files and the "proper use" of ssh and s/key. End users are made aware of issues that may result in embarrassment to the company or to themselves, or issues that may harm their reputations or the company's reputation.

We look for signs of intrusion, such as hackers sharing information among themselves about their recent discoveries. Other things to look for (coming from external machines or from internal machines) include:

- probes/port scans;
- excessive pinging or pinging our broadcast addresses;
- top source addresses, destination addresses, and service ports;
- unauthorized access; and
- "changes," such as sudden increases in traffic by one machine.

Cases

Case 1 - Misconfigured System

ORNL's address space was scanned (by another government facility) for machines providing service on TCP port 1. This generally identifies SGI machines because they are the only vendor that enables this service by default. The port scan identified a machine at 11:23:19 and at 11:24:05; a user attempted to log into the machine via telnet. The attacker first tried "lp," which was password protected, then tried "demos," which was not password protected. The attacker grabbed the password file and the NIS password file then created a "hidden" directory (.a) and ftp'd their exploit and hack tools from several locations, including their "home base." The attacker exploited a bug in the "df" command and became root, started a sniffer program, and cleaned up the log files to remove their login records. The sniffer program was named "diag," and it logged to a file named /usr/spool/lp/.a/err. The attacker set a password on the "lp" account.

The port scan was detected by our "TCP connection logger." The "telnet" sessions were logged by our "keystroke logger." Access to the machine was disabled at the firewall. The system administrator was notified. All passwords were changed, and the operating system was reinstalled. This machine was on a switched Ethernet port, so the sniffer didn't log anything. After analysis of log files and files left by the attacker and in cooperation with remote sites (that were affected by the hack), it was determined that the attacker was in Russia.

Case 2 - Unpatched System

Our address space was scanned by a machine in Brazil beginning at 22:07. This was an "mscan" type scan. Beginning at 22:16, a second machine from Brazil did a "follow-up" scan of selected machines. At 23:35 a bug in "named" was exploited on one machine, resulting in a "root shell xterm" being sent to that second machine in Brazil. The attacker created the directory "/tmp/.a" and ftp'd their hack tools (a sniffer and a "smurf" program) into it. The attacker named the sniffer program "update" and the log file "blah.log" and installed a trojan /bin/login to allow rewt/lamer! to log in without being logged. The attacker also created an account "blah" without a password but did not clean up the system log files. The sniffer proved fruitful, and at 00:07, the attacker logged into another machine and tried to exploit known vulnerabilities to become root. This attempt was unsuccessful.

The port scan was detected by our "TCP connection logger." The login sessions were logged by our "keystroke logger," and access to the machines was disabled at the firewall. The system administrator was notified. All affected passwords were changed, and the operating system was reinstalled. We got no cooperation from Brazil and were not able to determine any further information about this attacker.

Case 3 - Sniffed password.
Our TCP connection logger indicated an unusually large increase in network traffic from one of our machines. The keystroke log indicated "inappropriate" activity on four machines. Analysis of the four machines revealed that an IRC "bot" was running. (An IRC bot is a program that allows a user to appear to be logged into IRC even after they have logged off. It has several features including allowing the user to keep their IRC nickname and channels so that no one else can take them.) There also were other hacker tools left on some of the systems. Network activity for the four machines was monitored for a couple hours before closing off access to those machines. Although the attacker attempted to exploit known vulnerabilities with these systems, he was unsuccessful. Review of the log files revealed all the remote sites visited by the attacker; those sites were notified of the activity. It was clear from our logs (and verified later) that an authorized user had his login information "sniffed." It was uncertain at first where it had been sniffed. The following day, unauthorized logins occurred on five other machines, using a different login name. The attacker was able to gain root access on two of the machines and installed a sniffer program on each machine. He changed passwords on some accounts and also installed an IRC "bot" on one of the machines.

Unfortunately, our keystroke logging machine had a disk failure during the critical time, and we were not able to log all of the activity. Fortunately, though, our other logging machines were running. Using them, along with system logs from some of the compromised machines, we were able to piece together the full attack. Like the previous user, it was clear that this user had his login information sniffed. Both authorized users were at the same remote location. We surmised (and later verified) that the accounts were "sniffed" at that location. We closed off access to the five computers and began reviewing log files and files left by the attacker. We contacted remote sites that were affected and requested information. Piecing this information together revealed the exact identity of the attacker, who was from a U.S. city. Anticipating a return visit to our machines, we replaced one of the hacked machines with a special-purpose machine, which appeared to grant access. It kept the attacker "busy" while the machine notified us. While the attacker was "on," we contacted the ISP from which he was coming. That ISP verified that the attacker was coming from the same location as the previous attacks. The attacker returned several times later to see if his "holes" were still open.

The "unusual" network traffic was detected by our "TCP connection logger." The login sessions were recorded by our "keystroke logger." Access to the machines was disabled at the firewall. The system administrators were notified. All affected passwords were changed, and the operating systems were reinstalled on the machines that were "root" compromised.

Case Summary
Having multiple computer systems for intrusion detection allowed us to detect an intrusion, even when one of those monitoring systems was down. Diligent review of log files enabled us to detect the intrusions early. Quick response by security personnel contained the "damage" and permitted a short down time for the compromised systems. In all cases, we continue to monitor recently "hacked" machines and accounts for at least several weeks following an intrusion. This monitoring will often show other "hostile" sources that we can then either watch for or we can notify appropriate authorities. In at least one case, we detected unauthorized login attempts using login names and passwords that had been sniffed five months prior. Fortunately, those passwords had long since been changed.

Summary

An effective intrusion prevention and detection system includes limiting your vulnerabilities, knowing what methods attackers are using, educating your users, implementing hardware and software solutions that detect those vulnerabilities and attack methods. Knowing how to use the IDS and actually using it are critical. Day-to-day log review is boring at best; automating the review process is necessary. Diligent review of logs is paramount to detecting intrusions early and to limiting the damage. Effective computer and network security is very dynamic. Security staff must continually learn and experiment, developing and enhancing tools. Intrusion detection tools are only one part of an effective computer security plan. Trying to prevent the intrusions is of utmost importance and can be accomplished (or at least reduced) by educating end users, training system administrators, and running vulnerability checks on your machines. Cooperation from remote sites is necessary to close all holes.

References

William R. Cheswick and Steven M. Bellovin, "Firewalls and Internet Security," Addison-Wesley, 1994.

Vern Paxson, "Bro: A System for Detecting Network Intruders in Real-Time," Seventh USENIX Security Symposium Proceedings, pp. 31-51, January 1998.

"Network Intrusion Detector," Lawrence Livermore National Laboratory (UCRL-MA-116609 rev.3), November 1997.

Internet Security Systems, Inc., "Internet Security Scanner" and "RealSecure," http://iss.net, 1998.

Shadow, SANS Institute, http://www.nswc.navy.mil/ISSEC/CID, 1998.

Computer Incident Advisory Capability (CIAC), CIAC Bulletins, http://ciac.org, 1998.

Computer Emergency Response Team (CERT) Coordination Center, CERT Advisories, http://www.cert.org, 1998.

Rootshell, Exploit Information and Hacker Tools, http://rootshell.com, 1998.

BUGTRAQ, Bugtraq mailing list archives, http://www.geek-girl.com/bugtraq/index.html, 1998.

Computer Operations, Audit, and Security Technology (COAST), http://www.cs.purdue.edu/coast/coast.html, 1998.