NERSC Experiences with Security Tools and Monitoring

Tina Butler
National Energy Research Scientific Computing Center
tbutler@nersc.gov

ABSTRACT: *The US DOE's National Energy Research Scientific Computing Center (NERSC) is responsible for providing an open high performance computing environment for the DOE research community. We will describe how NERSC uses Cray MLS-based security tools, third-party tools like* ***ssh*** *and* ***tripwire****, and locally developed tools to secure and monitor its' T3E and SV1 systems.*

Introduction

NERSC, located at Lawrence Berkeley National Laboratory (LBNL) in Berkeley, California, serves a broadly based community of scientists and researchers with a large variety of projects within the U.S. Department of Energy (DOE) Office of Science. DOE and DOE-sponsored users access NERSC machines from across the country and a variety of international locations. Because NERSC is located at a non-classified DOE facility and supports an open research environment, it has the challenge of protecting critical research assets while preserving open access and easy interchange between scientists.

NERSC environment

Currently, NERSC HPC systems include a 696 PE Cray T3E-900, a 96-processor PVP cluster consisting of 3 Cray SV1s and one J90SE, and an IBM RS6000/SP system with 608 processors. NERSC’s, and to a large extent LBNL’s, requirements for long-term file storage are met by an HPSS-based mass storage system with a capacity of 600 Terabytes. To support these systems, and the NERSC staff, there are a large number of smaller specialized servers and workstations. NERSC is connected to the outside world through the DOE Energy Sciences Network (ESnet) via OC12. This paper will focus on the Cray HPC systems at NERSC.

Risks

Although NERSC is an unclassified facility, its’ computing assets are critical to the scientists that use NERSC resources in their research programs. The major issue at NERSC is not revealing classified or sensitive information, but the effects of computer vandalism. The data stored at NERSC, while not “sensitive”, is vital to a number of research programs. Any system compromise that leads to data loss or corruption could seriously affect the progress and validity of important research projects.

NERSC HPC systems are heavily used; all available cycles are committed to DOE and DOE-sponsored users. A system compromise that led to loss of cycles through damage to the system or illicit use of computing resources would deprive those legitimate users of their rightful access; in effect, a denial of service.

A third risk is damage to NERSC and DOE credibility. If NERSC does not diligently protect the resources given to it by the Department of Energy and Congress, it is unlikely to get more funding for further resources.

Perceived and demonstrated dangers

Although attacks may come from either inside or outside, this paper will deal only with the external risk. NERSC is a widely accessible site, with a large university-based presence; a very public and attractive target. Because NERSC must remain open to researchers, and promote easy, legitimate interchange of data, it is not possible to isolate or to firewall the production HPC systems. This requires NERSC to walk a fine line between sufficient security to protect computational and data assets, and sufficient openness to allow scientists to do good science.

Probes and attacks against NERSC systems have increased significantly over the last year. A typical incident pattern is as follows:

- First, the DNS servers are queried, system by system, for host information for the entire domain. (No zone transfers are allowed.)
- Next, available services are scanned – the most popular targets have been SunRPC, bind, telnet, ssh, pop, imap, and the rcmds.
- Approximately 6-8 hours later, surgical attacks occur against specific hosts and services.
- After the attacks, attempts to connect and exploit attacked systems follow.

A year ago, the scan rate was about 1 every 2 days; now the scan rate is 2-3 per day. We have also noticed a rise in the incidence of coordinated scans where the scan, attack and exploit all come from different, highly synchronized, systems.

Security strategy

Since NERSC does have important resources to protect, and there have been a significant number of attempts to gain unauthorized access to those resources, the following general security strategy has been put in place.

Each system needs to protect itself as much as possible.

Each major externally accessible system at NERSC is configured using as much host-based security as is available and consistent with the access requirements of the system. Systems do not rely on network monitoring or firewalls for their primary security. Operating system security facilities for hardening, monitoring and auditing are used whenever possible. Systems are configured with unnecessary services disabled and security patches are applied regularly. The minimum number of necessary accounts is maintained on system console workstations.

Critical support systems should be isolated as much as possible (SWS, CWS)

Critical support systems that do not need to be directly accessible to users, like the system consoles, are isolated behind a firewall. Network traffic that is strictly internal to the NERSC HPC systems goes over private networks that are not routed to the outside world.

Administrators should be isolated (firewall)

Administrators, particularly of UNICOS MLS systems, have privileges associated with their user accounts, and sometimes with their workstation. Because of these privileges, it is prudent to provide a higher level of protection for administrator accounts and workstations than those belonging to unprivileged users. At NERSC, administrators live behind a firewall, and privileged access to production and support systems is limited to a handful of hosts and users.

Monitor and filter at the network router level in addition to host defenses

Although individual hosts can be configured to greatly enhance their security, there are a number of attacks that are not susceptible to strictly host-based defenses. Network monitors and routers are a critical component in detecting intrusions, alerting responsible administrators, and dynamically activating enhanced defenses.

What NERSC is using

In order to implement and support the security strategies above, NERSC uses a variety of different tools. Vendor provided software; open source or commercial third-party software and locally developed software are all used as available and appropriate.

Host-based tools and facilities

UNICOS-specific facilities

UNICOS Multi-Level Security (MLS) provides improved security up to B1 levels. It is useful even in completely unclassified/non-sensitive environments like NERSC when configured at a less stringent level of assurance. The facilities for security logging and auditing, monitoring and maintenance of file integrity, and separation of administrative roles are particularly useful in the NERSC environment.

Levels/compartments

Although most commonly associated with true multi-level secure computing, security levels and compartments can be useful in less constrained environments. NERSC runs with levels and compartments “flat”, i.e., all users at level 0 and no compartments. An additional feature available in conjunction with levels and compartments is Secure MAC. This consists of two special security levels **syshigh** and **syslow**, which are used to provide added protection to system binaries and world-accessible directories.

PALs

Another feature of MLS is the use of privilege assignment lists (PALs) for separation of administrative roles and privileges. This has proven to be extremely useful at NERSC.

Enhanced logging and auditing

The UNICOS **spaudit** facility allows a host to collect and log a large volume of data about security-related events that occur on the system. The types of events that are logged are configurable and range from user logins to access control violations to file unlinks. Besides MLS security violations and accesses, many types of events include information that is useful in day-to-day system monitoring. The **reduce** command is used for processing the security logs. It allows for filtering and selection of any of the defined event types.

Open source tools

TCP wrappers

Almost all Unix-based systems come, by default, with the most open network services configuration possible. Many of the services defined are obsolete, insecure, and unnecessary for day-to-day operations at most sites. Turning off unnecessary services is a relatively simple way to greatly increase system security.

In addition to turning off unnecessary services, it is also possible to restrict access to services through the use of the freely available **tcpwrappers** software. This open-source package includes a daemon, **tcpd**, which intercepts traffic to services defined in inetd.conf and only passes on requests that meet site-configured access rules. A library extends the tcpwrappers functionality to other open-source and local applications that do not use the inetd path.

Ssh

One of the biggest holes that any site has to cope with is the lingering dependence on cleartext, persistent passwords as a means of user authentication. Users have been shown to be consistently bad at choosing passwords, thus leading to easy guessing. Even well chosen passwords are subject to sniffing, cracking, and shoulder surfing. Eliminating cleartext passwords is a continuing NERSC priority.

A major step in the elimination of cleartext passwords was the adoption of **ssh** as a replacement for **telnet** and **rcp**. Many sites have gone to using **ssh**. Unfortunately, **ssh** requires a number of modifications to securely mesh with the UNICOS model for user identification and authorization (iauser). An unmodified **ssh** will build and run under UNICOS, but it will also give away root access to the system. A number of mod sets have been generated by different sites to make **ssh** work correctly on UNICOS and UNICOS/mk systems. The most widely distributed UNICOS mods are from SDSC; NERSC has put the following additional changes into **ssh-1.2.26**:

- correct pseudo-terminal reclamation
- different code for UNICOS iauser interface
- extended error checking for failure to create TMPDIR because of quota limitations
- issuing a banner message prior to authentication
- updating udb lastlog and logfails fields
- code to lock the **ssh** daemon in memory, and to set CPU and memory limits to unlimited.

In December of 1999, NERSC turned off **telnet** on all of its’ production systems.

Replacement for ***FTP***

In addition to finding a replacement for **telnet** that does not require cleartext passwords, it is equally desirable to replace standard **ftp** for the same reasons. NERSC is still evaluating several different approaches to providing **ftp** functionality without cleartext passwords. Any or all solutions have to support a variety of architectures, and must be able to integrate with the HPSS mass storage system. Due to the diverse community supported by NERSC, it is likely that multiple solutions will be adopted. Some of the options are:

- **ssh** tunneling of **FTP**. This provides safe **ftp** access through with **ssh** encrypting the **ftp** control channel. It can be complicated to set up for Unix clients, and only a limited number of Windows 9x/NT **ftp** clients properly support it.
- **Scp**. This component of the **ssh** package works quite well with Unix clients, however, performance can be poor since both control and data are encrypted. It is also not a viable solution for file transfer between NERSC HPC systems and the HPSS mass storage system. **Scp** is not supported for Windows 9x/NT and Mac clients.
- **Sftp**. This **ftp** replacement is only available with **ssh-2**. Licensing and UNICOS porting issues for **ssh-2** are still under investigation.

- **SafeTP**. SafeTP provides a proxy that intercepts all FTP client requests and encrypts the control channel, and optionally the data channel. SafeTP clients and servers currently exist for Windows 9x/NT and generic Unix systems.
- **Kerberized ftp**. This requires setting up a Kerberos infrastructure. There are also compatibility issues among the versions of kerberos available on different NERSC platforms.
- **GSI-enabled ftp**. This **ftp** has been modified to use the Globus Grid Security Infrastructure. GSI, based on the GSSAPI and SSL, uses X.509 certificates for authentication.

Host-level monitoring

UNICOS MLS-based tools

The **spcheck** command is a front-end for a constellation of security-monitoring tools, similar to COPS. Using spcheck, a host can be checked for setuid/setgid files, correct permissions and security labeling on files and directories, repeated user login failures, and other security-monitoring functions. The tools can also be used individually.

NERSC uses **spfilck** for monitoring permissions and security labeling on a selected set of system binaries and directories. Although **spfilck** does not monitor content changes in the set of files that it looks at like **tripwire** does, **tripwire** doesn't know about security labels and PALs.

One drawback of **spfilck** is that it does not come with a sample list of files to monitor; generating an appropriately inclusive list can be laborious. A good source of candidates is the privilege database found in /etc/privdb. A list of setuid and setgid files can also be generated with **spcheck.**

Unused userids and user login failures can be tracked with **spcheck** as well.

Open source tools

Tripwire

Tripwire is a tool for monitoring the integrity of files and directories according to their creation, modification and access times, permissions, and detecting any changes to file contents through the use of several different checksum technologies. Tripwire was originally developed by the COAST project at Purdue University, but moved into commercial product development. The original source version of Tripwire, ASR 1.3.1, can still be downloaded from Tripwire, Inc. Although it is advertised to lack some of the more advanced features of the commercial product (Tripwire 2.x), the commercial product is not available for UNICOS and UNICOS/mk.

Tripwire 1.3.1 builds quite straightforwardly for UNICOS and UNICOS/mk. The CRC32 and Snefru signature algorithms did not work correctly under UNICOS/mk, but all other elements have been problem-free.

Configuration requires determining which files you wish to protect, at what level, and at what frequency. Tripwire's monitoring is relative to a database of signatures that are generated by tripwire in an initialization phase. Generating checksums and signatures for the entire system at relatively high frequency can be costly, so tripwire users are advised to consider limiting the number of signatures used, and the frequency at which the file set is monitored. Normal tripwire monitoring activities rely on the accuracy and integrity of the database. It is important to generate the initial database from a system that is known to be clean, and to keep the tripwire database safe from intruders – read-only media is suggested.

Local tools

UNICOS security auditing can generate enormous logs on a daily basis, especially if you do any kind of detailed data collection. Extracting information from the logs and doing any detailed monitoring using it can be daunting. The **reduce** command allows filtering and extraction of security log entries according to a number of criteria, but there are no other tools supplied for further reduction of those entries.

At NERSC, a locally produced set of scripts, **reduce.logs**, is used for post-processing login information from the security logs. This tool extracts login records, and then compares the (user, source, service, status) tuple with a database of previous login-tuples. This comparison can be very useful as a way of monitoring day-to-day login patterns. The login history that is maintained is also useful when checking for deviations from usual user access patterns.

Network monitoring

In addition to tightening the configuration of individual hosts, and monitoring host connections and activities, NERSC relies on real-time network monitoring and intervention as an additional protective measure. The facility used for monitoring NERSC network traffic is **Bro**, a stand-alone system developed by Vern Paxson of the LBNL Network Research Group. NERSC has been using

Bro to monitor network traffic across its DMZ for about 18 months, and has been able to significantly improve the detection of scans and blocking of attacks.

Bro supports real-time monitoring of traffic over FDDI, and 10 Mb, 100 Mb and Gigabit Ethernet, and can generate responses, in the form of router filters, based on that traffic. Currently, **Bro** analyzes traffic for six applications: **finger**, **ftp**, **portmapper**, **ident**, **telnet** and **rlogin**. Packets are chosen and analyzed on a connection basis and examined for predetermined attack fingerprints. Packets of interest are extracted in their entirety and saved. **Bro** can also be run against archived tcpdump files. This has proved useful in tracing and characterizing new attack fingerprints. Because **Bro** separates mechanism and policy, it is readily extensible both in what types of monitoring it can support, and the implementation of responses.

NERSC has **Bro** monitoring the two separate choke points into the internal network. Each **Bro** system handles an average of 100 GB of traffic each day; of this about 90% is **FTP** data, 7% is encrypted **ssh** traffic, and the remaining 3% is analyzed and archived. When a scan is detected, **Bro** generates a router filter to block the scanning host. In addition to blocking all hosts that scan, NERSC blocks all SunRPC traffic from external sources. These policies have reduced the detected attack rate against the NERSC network from 2-3 per week to 2-3 per month.

Because LBNL has a close working relationship with the University of California, Berkeley, the Lab has been allowed to use **Bro** to monitor traffic on the UCB network. The UCB network has a high volume of traffic, and is an early target for many new varieties of attack. **Bro** analysis of UCB network traffic has been very valuable in characterizing these attacks and generating new fingerprints for the **Bro** library.

Bro is available in source form from Vern Paxson at LBNL. Attack fingerprints are distributed to the **Bro** community as well. **Bro** runs on FreeBSD; hardware for a single system would cost approximately $10,000. The human time required for maintaining a **Bro** system – mostly reviewing logs – varies according to the amount of traffic at your site. At NERSC, reviewing **Bro** logs requires about 30 hours a week for analysis and backchecking.

Results

What are the results from these activities? Successful attacks against NERSC have greatly reduced. There have been no known root compromises on any of the large HPC systems, although some user accounts have been compromised through sniffed passwords. Active network monitoring and proactive defenses in response to threatening behavior from external hosts, as well as hardening of individual hosts have been very successful in preventing serious system losses.

Future directions

Of course, the rate of change in the types and numbers of attacks against systems is increasing rapidly. To attempt to keep pace with these increasing risks, NERSC plans to continue to enhance the existing network monitoring and intrusion detection tools and look for better ways to automate the auditing of all those logs.

Changes to host-based tools on UNICOS are somewhat limited by the maturity of the operating system – it seems unlikely that UNICOS will acquire IPSEC functionality in the kernel unless Cray significantly revives UNICOS and UNICOS/mk development. NERSC's IBM SP does have kernel IPSEC support, and it is being used to good effect. Porting and installation of ssh-2 is likely, if licensing issues can be resolved.

Selection of one or more appropriate replacement technologies for FTP, in the near future, will bring further progress toward getting rid of cleartext passwords completely.

Further Information

The main site for ssh is www.ssh.fi. The current state of the UNICOS version of ssh is unclear.

Tripwire is available from www.tripwire.com. The 1.3.1 Academic Source Release is the latest version of tripwire that is compatible with UNICOS.

To get more information on **Bro**, look at www.aciri.org/vern/bro-info.html. There are links there for Vern Paxson's excellent paper describing **Bro**, pointers to the **Bro** mailing list, and information on how to obtain the **Bro** source.

This work was supported by the Director, Office of Science, Division of Mathematical, Information, and Computational Sciences of the U. S. Department of Energy under contract number DE-AC03-76SF00098.