Ace/Agent for the Cray

Erv Kuhnke, Steve Finn, UTA Chris Macneill, RSA
Sponsored by Jacqueline Bell
Defense Threat Reduction Agency

1.0 Introduction and Background

The Defense Threat Reduction Agency (DTRA) is an independent agency of the United States Department of Defense (DoD). The mission of DTRA is to reduce the threat to the United States and its allies from nuclear, biological, chemical (NBC), conventional and special weapons through the execution of technology security activities, cooperative threat reduction (CTR) programs, arms control treaty monitoring and on-site inspection, force protection, NBC defense, and counterproliferation (CP); to support the U.S. nuclear deterrent; and to provide technical support on weapons of mass destruction (WMD) matters to the DoD Components.

DTRA owns a Cray SV1 which is operated by its Weapons Phenomenology and Advanced Computing Branch. The SV1 is located at DTRA's data center in Alexandria Virginia. We have 16 CPU's and 2 Gigawords of memory. We currently use the Unicos 10.0.0.7 operating system. This is a non-sensitive, unclassified system. We support on-site users and those who are remotely located at contractor sites or other government laboratories. Remote users have connectivity through the Internet, DREN, or via dialup PPP.

Because of increased security awareness we wanted to eliminate multiple use passwords to prevent attacks where someone captures a login session and reuses the login sequence. The use of one time passcodes generated by a physical token was selected as a technique to obtain this capability. Use of a PIN number in conjunction with the physical token was desired to restrict unauthorized use of the token if it was lost or stolen.

In addition to replacing use of passwords, we wanted a solution that could be extended to provide software based encryption of communications to/from the Cray to increase privacy and make packet sniffing more difficult.

Our site has a very small systems staff. We do not have an onsite Cray Inc. Analyst and we do not have the source of Unicos or the expertise to do much with it if we did.

2.0 Alternatives Considered

At the time of our evaluation, PKI solutions utilizing digital certificates were not fully developed, and even now are not commonly used for applications such as Telnet and ftp. We considered three alternative tokens and five alternative architectures.

2.1 The Tokens

2.1.1 Smarty Card

Other systems at DTRA use the Smarty Card from Fischer International Systems Corporation (FISC) http://www.fisc.com. This card is unique in that it is shaped like a 31/2 inch floppy disk and can be inserted in a floppy drive. This is good in that a separate card reader is not required and is bad if you don't have a floppy drive, want to use your floppy, or have a laptop with either a floppy or CD-ROM but not both at once. The Smarty SDK supports MS-DOS, Windows 3.x, Windows 95/98, and Windows NT. In addition, several VPN vendors support the Smarty Card. At DTRA, it is used in conjunction with the V-One product. There is not a Unicos client, and the SDK does not support Unix, Linux or Macintosh so we did not choose this device.

2.1.2 CryptoCard

The CryptoCard http://www.cryptocard.com was considered because it is currently used at Los Alamos National Lab. DTRA uses some computers at Los Alamos and owned a Cray YMP-M98, which was located at Los Alamos until its retirement in October 1999. At the time of our evaluation, Los Alamos was still using RSA SecurID cards and in fact still accepts them although CryptoCard's are issued to new users. There is no client for Unicos. At Los Alamos, the CryptoCard is used in conjunction with Kerberos. The vendor has GUI Clients for WindowsNT/98/95 and command line interface for both client and server for Linux, Solaris, FreeBSD, AIX, and NT. No Macintosh client is supported. This was a strong contender but would have required that we implement Kerberos or have someone port an existing client to Unicos. That would have been difficult for use because we did not have Unicos source code or expertise.

2.1.3 SecurID

The SecurID ACE pinpad from RSA http://www.rsasecurity.com/products/securid/ and the ACE Agent software was our choice. SecurID cards were in use at DOD HPC Modernization Program sites, at Los Alamos, and for another system at DTRA with some significant overlap of user base with our system. We knew it worked well because we had used it on the DTRA YMP-M98 although we didn't know exactly how since someone at Los Alamos did the work for us.

2.2 The Software

2.2.1 Kerberos

The first option we considered is Kerberos because of cost considerations. (It's free) We knew that Ken Hornstien at Naval Research Lab had developed a KNSC, which could authenticate to an ACE agent, and Los Alamos had a KNSC that worked with the CryptoCard. Our users need to authenticate to each of these realms. The Los Alamos and DOD HPC versions of Kerberos are incompatible. DOD uses a password and a

SecurID passcode, which requires clients that are modified. Los Alamos uses only the passcode so standard clients can be used. In either case, Kerberos V5 uses the IP address for encryption and does not work with firewalls that use Network Address Translation (NAT). There are versions of Kerberos that can traverse NAT firewalls but the range of client implementations does not cover the range of systems that connect to our SV1.

We have a single Cray and it is doubtful that we will ever have a large network of heterogeneous systems. We have Sun and SGI systems which perform service functions such as controlling a tape silo, functioning as a print server or operator's console, but end users do not login to those systems. It seemed like a nightmare in the making to implement Kerberos at our site when the main functionality we wanted was to have Unicos prompt for a passcode and send it off to the ACE Server for authentication. It's already bad enough switching between the Los Alamos realm and the DOD realm, neither will accept cross realm authentication from the other, and we did not want to add a third realm.

2.2.2 VPN's

We then looked at VPN software that could perform the authentication. The idea was that the Cray could not be seen from the outside world and the only access to it would be through the VPN that would obtain the passcode. Two products were considered. The V-One SmartGate VPN Gateway www.v-one.com has a client supported on Windows 95, 98, NT 4.0 w/SP 4 & 5, MacOS System 8.1 or later, Solaris 2.6, Win CE 2.; P/PC, H/PC, H/PC Pro, and Red Hat Linux 6.0 .

Check Point Software Technologies Ltd. http://www.checkpoint.com provides VPN-1 client, VPN-1 SecuRemote and VPN-1 SecureClient all of which support Windows98/95/NT.

Because we have users with workstations using IRIX, AIX, HP/UX and other unsupported operating systems, these solutions were not selected. Both the V-One and Checkpoint product lines can interface with the RSA family of tokens and we may implement this or some other VPN solution in the future for those who have supported clients. In addition, although Checkpoint now offers a PCI board based hardware accelerator for use with NT or Solaris on the server, that product was not offered at the time we did our selection over one-year ago. Without hardware acceleration, either of these solutions would have required a quite large multi-processor server to support our connection to the Internet.

2.2.3 Front End

Another option that was our main fallback position was to provide a front-end system that had a client for one of the tokens. Users would Telnet or ftp to the front-end, be prompted for the passcode, and then passed through to the Cray. For interactive we would use a captive login that would invoke Secure Shell to complete the login to the

Cray. For FTP, we would NFS or DFS the front-end to the Cray and users would just ftp to the front-end.

2.2.4 ACE Agent

As described by the vendor “RSA ACE/Agents function much like security guards, standing between the user and a protected resource or device to enforce two-factor authentication via RSA ACE/Server.” In our case, each user’s shell is set to /usr/local/ace/prog/sdshell. Sdshell prompts for the passcode and once authentication is successful, control is passed to the user’s shell of choice, as defined in the ACE/Server database. This is the “standard” way of implementing SecurID authentication on UNIX systems without exits in the operating system binaries. To provide backward compatibility with sites migrating from the old ACM7100 standalone software the libacm.a interface with login, su, ftp and telnet has also been provided.

3.0 Implementation

As mentioned in section 2.1.3 above, we used SecurID cards on DTRA’s YMP-M98. Historically interfaces were developed by Security Dynamics/Cray to implement User Exits in a number of UNICOS binaries, e.g. login, ftpd, and su. This interface is provided via static library known as libacm.a. The original software, ACM7100, developed by Security Dynamics (now RSA Security Inc.) was a standalone product with a local database.

When the ACE software needed updating on Unicos someone at Los Alamos just changed the Unicos source themselves along with other changes unrelated to our topic. Because DTRA’s YMP-M98 was the last unclassified Unicos system at Los Alamos, we foresaw a problem in obtaining ongoing support from Los Alamos if we just adopted their existing code on our machine. Another minor detail was that we do not have Unicos Source code. DTRA contacted both Cray and RSA and discussed the issues involved. It happened that RSA was involved in a port of the ACE Agent to Unicos/MK for another customer and was willing to offer DTRA a quote on performing a similar port for us.

At the final stages of the project when investigating why the new ACE/Agent libacm.a linked ftp daemon wouldn’t work, Chris Macneill discovered that the PASSCODE string was being passed to the libacm.a API in encrypted form. According to the interface specification this PASSCODE should have been passed in clear text.

It appears that at some point in the past Cray had received a change request to encrypt PASSCODE, rather than pass it in clear text. Changes were made in the way the Cray binaries interfaced with the libacm.a that probably made sense to those implementing them. Unfortunately the implementor does appear to have not fully understood the interface and did not appear to have discussed the changes with Security Dynamics/RSA Security prior to implementation.

This change worked OK for the local database in the old ACM7100 product as the user's credentials can be encrypted using the same algorithm and a binary compare performed. The ACE/Agent uses a different encryption/hashing method before passing the data to the ACE/Server. Passing an encrypted PASSCODE to the ACE/Agent causes an authentication failure, as the ACE/Server does not expect to encrypt the PASSCODE before validating the supplied PASSCODE using the ACE/Agent authentication protocol. Consequently NQS cannot currently be supported in the ACE/Agent implementation. If there is sufficient demand, it is a relatively simple job to enhance the ACE/Agent protocol to allow the ACE/Server to validate a DES encrypted PASSCODE. Whilst the work is fairly simple, development scheduling means that this could take 6 to 9 months to be delivered, after the need has been identified.

The problem with ftp resulted in SPR 717278, which was resolved by Cray in less than six weeks.   Thanks to David Ecale, Dave Turgeon and whoever else at Cray they got to actually make the changes.

Since the login, su and telnet binaries can also be linked to libacm.a, it is possible the same problem exists with them. These weren't tested at the DTRA site as the sdshell method was chosen. This provides protection for login, su and telnet without having to relink object code and was a new install rather than a migration from ACM7100.

The sdshell methods occurs following successful operating system Username/Password authentication. Thus the prompts are Username, Password and PASSCODE. The libacm.a method replaces the operating system password for those users with Tokens. Since the database is no longer local, with this method it is necessary to have a local configuration file (/etc/aceusers) to determine which users have Tokens. This can be ALL, NONE or a mixture of Token and Non-Token users using an include or exclude list.

4.0 Results and Conclusions

It works. We were able to import the token records provided by Los Alamos for existing SecurID cards into DTRA's system. For interactive logins there is a standard Unicos prompt for password, then the RSA sdshell invokes the necessary software to prompt for and authenticate the passcode. For ftp the user enters the passcode at the password prompt and no password is used. Use of the passcode only for ftp allows use of standard ftp clients. A Sample screen dump of the login process is shown below:

```
Cray UNICOS (Q) (ttyp014)

*******************************!!!!!!!!***************************************
This is a private computer facility.   Access for any reason must be
specifically authorized by the owner.  Unless you are so authorized,
your continued  access and any other use may  expose you to criminal
and/or civil proceedings.
******************************************************************************
```

```
login: sf
Password:

Active label set to : level0,none

Last successful login was : Fri Apr 28 17:38:15 from 208.237.197.39
Enter PASSCODE:
PASSCODE Accepted
```

Current work involves a transition from use of Telnet and ftp to use of Secure Shell (SSH) to provide encryption of traffic to/from the SV1.