Cray Security and the Distributed Computing Environment

Brian Gaffey and *Stephen Lord*, Cray Research Inc.

ABSTRACT: *This paper describes a number of security related enhancements that Cray is implementing for the Open Software Foundation (OSF) Distributed Computing Environment (DCE). Specifically we will cover DCE 1.1's security related features and how they are implemented in UNICOS. CRI has extended many of these features to provide security solutions for SuperCluster. We present an overview of CRI's Multi-level security (MLS) and how DCE supports it. Finally, we cover CRI's Network Queuing Environment's (NQE) use of DCE security services.*

1 Overview

For many years the computer industry has searched for an open, comprehensive solution to network security. This search has been frustrated by a lack of standards and general agreement in the industry. The Distributed Computing Environment set out to solve this problem in the early 1990s by standardizing on Kerberos version V from MIT. Now, nearly five years later, DCE is poised to address this problem.

In this paper we will discuss aspects of the problem and present our solutions to them. We will also discuss the additional requirements of integrating the DCE solutions into the UNICOS and UNICOS/mK environments.

First, we will discuss the components of the CRI environment. This includes DCE, the DCE security service, CRI's multi-level security and CRI's NFS id mapping. Next, we will present the security problems related to the CRI environment. Finally, we will present our extensions to DCE, UNICOS and UNICOS/mK which address some of the problems. We will also share some of our ideas related to the remaining problems.

1.1 DCE

DCE is a comprehensive software package that lets users develop, execute, and maintain distributed applications. It is operating system and network independent, enabling you to take maximum advantage of your software investment.

The Distributed Computing Environment is a layer of software that resides between a computer's operating system and application programs. Masking the physical complexity of the multi-vendor networked environment, it enables an application to be segmented and executed on the system best suited for each segment. Application users and programmers alike can take advantage of the power a Cray Research system brings to this open environment.

The following are the components of the DCE core services:

Threads. Threads are the fundamental building block of DCE. They provide the capability of concurrent execution on a multi-CPU computer. All DCE components are multi-threaded.

RPC. DCE's remote procedure call is an advanced programming tool. The RPC supports "at most once" semantics, variable length arrays and an internal protocol which allows efficient execution over wide areas networks (WANS). All DCE components are built using the RPC. For example, the DFS client requests data from the DFS server by sending an RPC to the server.

Time. DCE's Distributed Time Protocol (DTP) ensures that all nodes in the DCE cell have a consistent view of the current time. This allows the security service to "expire" privileges consistently throughout the cell. It also allows logs from various nodes to be viewed in a consistent manner. DCE allows for the use of other time protocols such as NTP to be used instead of DTP.

Directory Services. DCE's Cell Directory Service (CDS) provides name to location mapping in the cell. For example, when an RPC is first issued, the RPC runtime library sends a request to CDS to identify the server that is to handle the RPC. CDS is an advanced directory service which allows for caching of information on each node, replicating of CDS servers, and partitioning of the name space. CDS is integrated with the security service. All objects in the CDS namespace can be protected by access control lists which define the types of access granted to different users and groups. DFS makes use of CDS to provide a global, uniform namespace. The namespace model used provides for all file names to be cell relative and within the cell to be location independent. This means that the name of the file does not change even if the file moves or the user moves. Also, if DFS servers are added to the cell, there are no administration changes required on the DFS clients.

Security Service. The security service is based on Kerberos version 5 with extensions to support delegation and other features. The security service provides a network wide consistent set of user ids and accounts. All users in a cell must first be authenticated before using any service in the cell. Once authenticated, access to various objects can be limited (authorized) by access control lists (ACLs). The Security Service also provides for cryptographic checksumming of headers and data (subject to international exportation laws).

1.2 MLS

Multi Level Security or MLS is an extension to Unix provided by Unicos which provides a much higher level of security than traditional Unix implementations. It allows data and users on a system to be partitioned from each other so that they cannot communicate or exchange data in anyway, It breaks the special permissions usually associated with the root account down into a set of privileges thus providing much finer grain control over who can perform administrative tasks.

1.3 ID Mapping

ID mapping is a feature of Unicos used to mask different account to uid mappings between hosts. This is currently an extension to NFS which allows an NFS server to map between uids and gids supplied by clients to those in the local UDB.

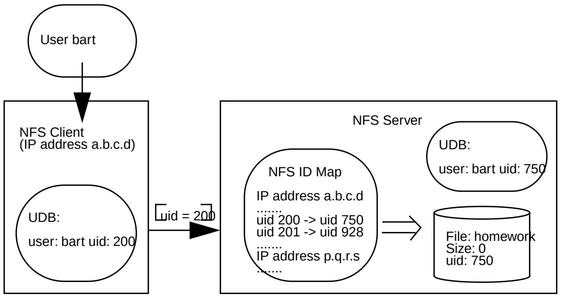

Figure 1: NFS ID Mapping

1.4 Kerberos Version 5

Kerberos 5 is the last release of Kerberos from MIT. MIT's project Athena never released version 5. However, as we said already, DCE chose version 5 as the base for the DCE security service. Once the reference implementation was created, the DCE vendors made extensions and corrections to the base code.

During this same period the Internet Engineering Task Force (IETF) took up the effort to standardize version 5. They used as a base the last release from MIT. Needless to say, the IETF version and the DCE version have diverged.

Many of our customers are interested in using the IETF version since it is in the public domain. DCE 1.2 will support interoperability with version 5 but we don't expect 1.2 products to appear until 1997.

2 Requirements

The requirements for security fall into two broad statements: A distributed computing environment should have all of the security assurances as a single computer system. Within a distributed computing environment, one system can not be allowed to compromise the security of any other system.

2.1 Integrated login and single sign on

A problem which people often expect DCE to solve for them is that of having to log onto multiple hosts in a network as they access them, and having to maintain multiple user databases. DCE has the user registry which should be able to act as a central repository for user information. However, DCE provides no utilities to integrate conventional unix network utilities with DCE security. In addition, the Unicos UDB associates many new fields with a user which are not available in the DCE registry

CRI's DCE product provides a feature called integrated login. When this feature is enabled, all authentications in UNICOS or UNICOS/mK will acquire DCE credentials. The process can then use these to access DCE services or DFS file systems.

Integrated login is the first step toward "single sign on". With single sign on a user only provides a password once. After that all accesses for services within the environment use the initial set of credentials to pass authentication. In Kerberos terms this requires ticket forwarding and ticket refreshing both of which are features of Kerberos version 5.

2.2 Intercell Authentication for DFS

Multiple DCE cells can be configured to communicate with each other in a trusted manner so that a user from one cell can be authenticated in another cell. This allows different administrative domains to share resources and information. However, when it comes to DFS there is a restriction in the way this is implemented. The DFS user in an inter-cell file access is referred to as a *foreign user*. A foreign user's identity is held as a set of universal unique identifiers or UUIDs, one of which represents the DCE cell they came from, the rest represent their identity (uid and gid) within that cell. Conventional unix filesystems do not have the ability to store, or perform access control checks on this representation of a user. Unless a DCE local filesystem (LFS) is used, all users from a foreign cell are effectively mapped to nobody.

An existing mechanism within Unicos, NFS ID mapping, is used to allow users from a foreign cell to access data via DFS whilst maintaining their individual identities.

2.3 Integrating DFS with Unicos MLS

Unicos Multi Level Security is an expansion of conventional Unix security. One of the features of MLS is the ability to compartmentalize data and users, and to propagate this information over a network. DFS does not currently support these concepts.

2.4 Accessing DFS files from an NQE job

There are a number of scenarios where an NQE job will require DCE credentials. In particular, if a user's home directory is accessed via DFS then the job will need DCE credentials in order to start-up. Since DCE credentials have an expiry time built into them, there needs to be a way of refreshing them without the user being present.

3 Intercell Authentication without LFS

Conventional Unix filesystems, Unicos's NC1 included, specify file ownership and access permissions in terms of a file's owner and group. NC1 also supports a form of access control list, but this is quite different from that provided by the DCE LFS. A DFS user is represented as one of three classes:

- Unauthenticated. The user is treated as nobody with a uid and gid of -2.
- Authenticated within the cell. The user has an account in the local DCE cell's registry. This registry contains a uid and a gid list for the user.
- Authenticated from a foreign cell. The user has an account in a different DCE cell which has been configured with secure keys to exchange information with the local cell.

Each DCE cell is a separate administrative domain with its own set of users. It is possible to create users in each cell independently without fear name space overlap between cells. When a user identity is passed between cells it is represented as the user's identity within the originating cell and the cell's Universal Unique Identifier or UUID. The combination of all three is guaranteed unique.

The problem with this representation is that it does not map directly onto the information which a traditional unix filesystem can store. DFS ACLs include the concept of foreign users, and can enforce file ownership and access control based upon them. This relies on the underlying filesystem (LFS) being able to store the data. In the case where DFS is accessing a non-LFS filesystem, all users from a foreign cell are mapped onto a single uid/gid pair resulting in only very coarse grain access control.

This clearly represents a problem for legacy filesystems, or for vendors, such as Cray, who do not have LFS support.

Unicos NFS has a facility called ID mapping which allows a remote user's uid to be mapped onto a local uid before any filesystem access occurs. Mapping is based upon the originating IP address of the NFS request. This means that two uid spaces have to be maintained, but in the case where client and server are under separate administrative control this may happen anyway.

This same mapping technique can be applied to DFS. Before any access to the filesystem, the user's identity obtained from their DCE credentials is mapped onto a local account. In the case of DFS the mapping can be thought of as being from accounts in a foreign registry to accounts in the local registry.

Whilst this does mean that a user has to have an account in both registries, it does allow files to be uniquely identified as belonging to a user from a foreign cell. Because DFS maintains state information about a user's identity, the mapping is not performed on a per request basis and is fairly efficient..

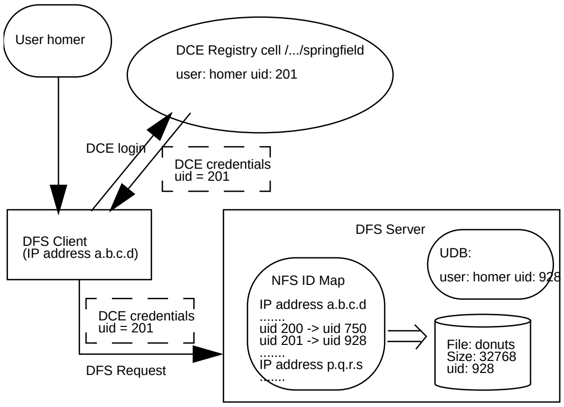

Figure 2: DFS ID Mapping

Mapping is still based upon the IP address, in future it could be based upon the cell UUID leading to simpler administration. IP based id mapping does however mean that it is possible to map from users in the local registry to user ids on a specific host should this be required.

Since the solution is wholly contained within the DFS server, clients from any DCE vendor can be supported.

4 Multi Level Security for DFS

Unicos MLS can extend the identity of a user and the ownership of a file to include Mandatory Access Control (or MAC) labels. A MAC label consists of a level and a compartment. A file exists at a particular level or in a particular compartment, a user may have permission to run at several levels or in several compartments. A user in one compartment or level cannot pass data to a user in another compartment or level except under a few very restricted circumstances.

The NC1 filesystem stores MAC labels with inodes on disk, and the Unicos UDB associates a MAC label range with a user's account. Using extensions to the IP protocol it is also possible to propagate this label information between hosts.

Our initial Unicos DFS implementation did not pass MAC labels between machines. When a user accesses a file via DFS they do so at the lowest level and with no compartments active. This means that any file with an active MAC label cannot be accessed via DFS.

In the Cray DCE 1.1 release, DFS has been extended to support the following on MLS systems:

- A user's MAC label on the client is propagated to the server and used to control file accesses.
- File MAC labels are visible on the DFS client. They cannot be changed via DFS.
- DFS will enforce the network access list (NAL) and will work through a NAL with a restricted MAC label range.

- By using the ID mapping facility, users will be restricted to the MAC label range configured in the UDB.
- PAL only configuration of Unicos is supported.

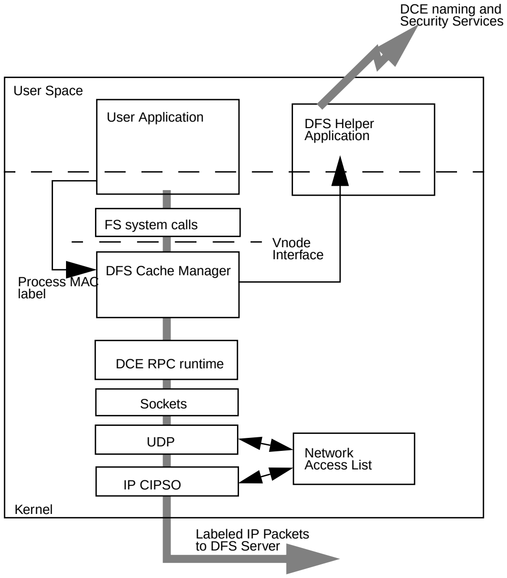

Figure 3: Passing MAC labels via DFS - the client

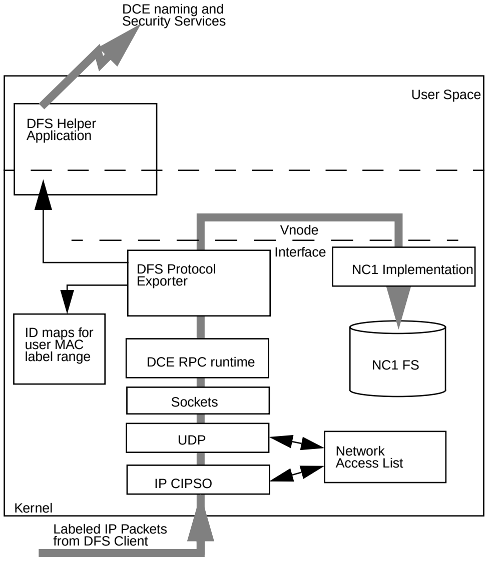

Figure 4: Passing MAC labels via DFS - the server

5 Kerberos 5 integration with DCE security

DCE Security is based upon Kerberos version 5, the DCE security server includes the kerberos servers. However, DCE passes kerberos information via its own protocols, not those defined by kerberos. The typical use of Kerberos is to implement secure versions of telnet, rlogin etc., the typical use of a DCE security server is to gain credentials for DFS access.

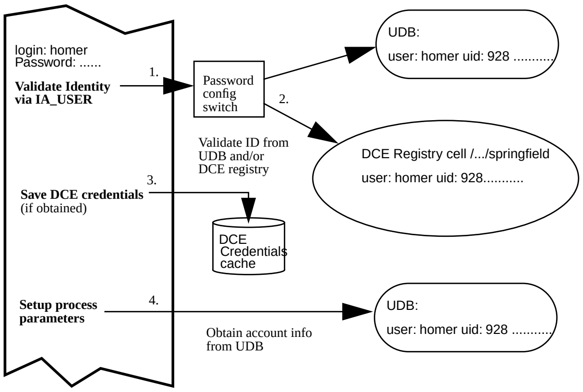

Figure 5: Integrated Login

In an ideal world these two are combined, a user would log into their workstation authenticating themselves using the DCE security server and obtaining a set of credentials suitable for accessing DFS files. Subsequently they would use kerberized telnet to log into a remote host where they would also obtain a set of DCE credentials. The only password supplied was the initial one to log into the workstation.

5.1 Integrated Login

The initial step in integrating DCE security and kerberos is to provide an integrated login mechanism whereby when the user logs into a host and types in their password, password validation is performed by the registry rather than by the local password file (the UDB). A configuration file controls how the password is validated, from the DCE registry, from the UDB, or from both.

If the registry was used for authentication then the DCE credentials are saved and associated with the user's session. The same technique works for telnet, rlogin with a password, ftp and nqe batch jobs.

Finally the UDB is used to set up all the remaining parameters of the user's session. The UDB is still required as it contains many extensions to the account information which are not available from a DCE registry.

5.2 Single Sign on

Integrated login always requires a password to function. Single sign on adds the ability to take a set of previously

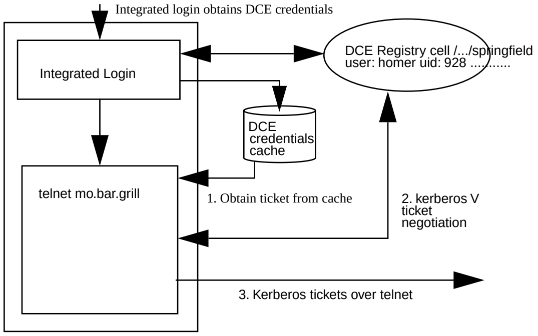
Integrated login obtains DCE credentials
Integrated Login
DCE Registry cell /.../springfield
user: homer uid: 928 ....
DCE
credentials
cache
telnet mo.bar.grill
1. Obtain ticket from cache
2. kerberos V
ticket
negotiation
3. Kerberos tickets over telnet

Figure 6: Single Sign On Leaving a Host

obtained DCE credentials and use them to gain access to remote services or systems and have DCE credentials for use on the remote systems.

Single Sign on makes use of Kerberos V (beta 5) versions of utilities such as telnet to ship tickets from the DCE credentials cache to remote systems. This kerberized version of telnet **forwards** the ticket to the daemon on the target system.

A ticket for the requested destination is obtained from the DCE This included a ticket granting ticket for the user at the destination host. The ticket is forwarded over the telnet protocol to a kerberos aware telnetd.

telnetd creates a a kerberos V ticket file for the user and then runs /bin/login The login binary executes a new DCE binary krb2dce which converts the kerberos V ticket file into a DCE credentials file. Finally the UDB is used to set up the session information before starting a shell process.

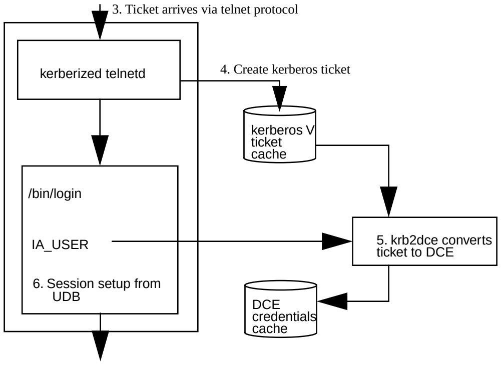
3. Ticket arrives via telnet protocol
kerberized telnetd
4. Create kerberos ticket
kerberos V ticket cache
/bin/login
IA_USER
5. krb2dce converts ticket to DCE
6. Session setup from UDB
DCE credentials cache

Figure 7: Single Sign On Authentication to Remote Service (telnet)

5.3 *The Network Queuing Environment*

CRI’s NQE supports access of DFS files today from any system. NQE also uses the normal mechanisms for authentication on CRI systems. Therefore, when the integrated login feature is enabled, NQE jobs will obtain DCE credentials and can access DCE services.

In future releases of NQE, tickets or credentials will be obtained at submission time and forwarded to a kerberos or DCE aware NQS. There, the execution server or the load balancer will periodically refresh the ticket and subsequently use it to initiate the job.

6 Conclusions

DCE can be integrated with an operating system’s services to provide an integrated solution to security. It is possible for a user to login only once and have secure access to services. It is also possible to protect each system in a distributed environment from threats from other systems. Finally, it is possible to extend DCE to provide support for MLS systems.

7 Future Direction

The current implementation goes a long way towards addressing distributed security. However, there is more to do. When the DCE 1.2 release is available CRI will support that implementation of kerberized utilities and its forwarding mechanism. This will provide secure non-password access between DCE systems.

DCE 1.1 supports MLS’s MAC and DAC. Later releases will add support for auditing.