HIERARCHICAL TRUST RELATIONSHIPS FOR INTER-CELL AUTHENTICATION

1. INTRODUCTION

DCE 1.0 provides peer to peer inter-cell trust relationships. This requires that a pairwise exchange of keys between cell administrators for each pair of cells that are to communicate. While the DCE administration tools make this exchange simple, it remains an exchange that must be performed periodically.

This RFC restates the specification for hierarchical trust of the DCE 1.0 privilege server [Pato 90], which greatly reduces the number of keys that must be exchanged to cover a given set of communicating cells. A more formal presentation of the trust issues involved can be found in [GLP 92].

2. AUTHENTICATED INTER-CELL COMMUNICATION

Inter-cell access to target services is provided by two features: the ability to place principals from foreign cells on access control lists and the ability to construct a PAC certified for presentation to a service in a foreign cell.

An ACL contains a list of entries that identify principals and their associated rights to an object. Identification of a principal consists of two components: the principal's identifier (UUID) and the identifier (UUID) for the principal's certification authority (cell).

Reference monitors grant access to objects by comparing an identity sealed in a PAC with the identities in ACL's. The PAC represents the initiator's identity and group privileges with identity pairs of principal/group ID and authority — analogous to the way these identities are represented in ACL's.

2.1. Cross-Cell Trust

When allowing principals from foreign cells to be included in an ACL, an application is forced to trust services other than just the local cell's authentication and privilege server. Minimally the application must also trust the authentication and privilege server from the initiating cell to certify the initiator's identity. In addition the cells' authorization services must be able authenticate each other so that applications can rely on the foreign certificates.

Trust in the local cell is dependent on a key shared between the authentication service and the principal. A foreign cell's authentication service does not possess this secret, instead it shares a different secret with the local cell's authentication service. This secret, the inter-cell key, is used to allow principals from the foreign cell to obtain tickets to the local cell. Trust is established between cells that either share inter-cell keys, or between which there exists a sequence of cells that share inter-cell keys. In the DCE transitive trust is limited to those cells which directly share inter-cell keys and to inter-cell sequences that form a well defined hierarchy.

2.2. Transitive Trust

Allowing cells to establish authenticated communications by sharing a key is comparable to the mechanism used to allow principals to establish authenticated communications. And, as is the case with principals, it is too cumbersome for each cell to have a shared secret with every other cell with which it may need to interact.

The authentication service provides the solution to the scaling problem for authenticated communication between principals in a cell, but no comparable service can resolve the problem for cells. If such a service existed, it would be difficult to imagine cell administrators from independent organizations agreeing to a common repository for the key distribution service. Even in public key based systems, it is hard to imagine cell administrators agreeing to an external agent controlling the key for the global certification authority.

Transitive trust allows the use of a chain of point-to-point authentication steps using shared keys between each link. This avoids both the space complexity of managing keys for every cell a given cell may communicate with and the unacceptable creation of a global authority. Unrestricted transitive trust, however, is as dangerous as a global authority. Subverting a single cell will allow an intruder to masquerade as any other cell — the intruder need only construct a dummy cell for the intended victim and then use a transitive trust path that passes through the compromised cell.

2.3. Hierarchical Transitive Trust

Imposing a hierarchical structure on the transitivity of trust allows potential damage to be contained and provides an algorithmic description of trust between cells. The Privilege Server uses this structure for determining the trustworthiness of cells involved in certifying a foreign principal. The following diagram depicts a portion of the naming tree. The nodes `/.../hp.com`, `/.../dec.com`, and `/.../apple.com` are cell roots that correspond to the root cells for the corresponding companies. Subordinate to these nodes are other cell roots that correspond to organizational structures of each company.

```
          hp.com                dec.com               apple.com
             |                     |                      |
    ----------------        --------------         --------
    |    |        |         |      |     |         |      |
   hpl  ma      boise      lkg    src   crl       mac    II
         |                  |
       apollo              rpc
```

The goal is to allow cells to act as intermediaries for other cells while preventing this extension of trust from subverting the entire global environment. Compromising a cell should not result in compromise to cells that are unrelated to the cell. Damage should be restricted to objects subordinate to that cell and possibly to objects in other cells that are open to access by principals in the subverted cell or its subordinates. For example, compromising the `apollo` cell should not result of compromise to any data in the `apple` cell except possibly for data that is available to principals in the `apollo` cell and for data stored in the `apollo` cell. A small number of rules governing the transitivity of trust provide these features:

(a) A peer cell is trusted to certify principals from that cell.

(b) Cells are trusted to certify descendent cells.

(c) Ancestor cells are trusted to certify any cell they trust.

(d) Cell traversal must first travel up the hierarchy to an ancestor which will serve as an articulation point. Traversal may then proceed across 0 or 1 peer links. Traversal may then continue down the hierarchy to descendents. Once a peer link or a down link has been followed, no subsequent ancestor links may be followed.

These rules allow us to construct an environment as depicted in the following figure:

```
          hp.com - - - - - - - - dec.com              apple.com
             ^                     ^                      ^
    ----------------        --------------         --------
    ^    ^        ^         ^      ^     ^         ^      ^
   hpl  ma      boise      lkg    src   crl       mac    II
         ^                  ^
       apollo              rpc
```

This figure depicts ancestral/descendant relationships with the token ‘‘^’’ and peer trust relationships with a dashed line ‘‘- -’’. In this figure any principal in any cell at HP can trust other principals at HP and at DEC; any principal at DEC can trust principals from DEC and HP; principals at Apple can only trust other principals at Apple. Trust between principals at DEC and HP exists if the HP and DEC cells were involved in the certification process. Subversion of the HP cell allows an attacker to impersonate any legitimate HP principal to any cell inside HP or DEC except for the principal’s actual cell. It also allows the attacker to impersonate any principal from any other cell (e.g., an Apple cell) to cells inside HP. This attack, however, does not allow the attacker to impersonate another cell (e.g., Apple) to DEC.

The following diagram shows a different trust model between the companies. Instead of a high degree of trust at the upper level of the organization, there is a more limited degree of trust established. The

`/.../hp.com/ma/apollo` cell shares keys with the `/.../dec.com/lkg` and the `/.../apple.com/mac` cells. In this scenario principals at `apollo` can interact with principals at `lkg`, `lkg/rpc` and `mac`. Principals at `lkg` and `lkg/rpc`, however, cannot interact with principals at `mac`.

```
              hp.com                   dec.com              apple.com
                 ^                        ^                      ^
        ----------------         --------------           --------
        ^    ^        ^          ^      ^     ^           ^      ^
      hpl   ma      boise    / - lkg    src   crl         mac     II
             ^             /     ^                       /
         apollo - - - - -       rpc                     /
            |                                          /
            + - - - - - - - - - - - - - - - - - - - - -
```

Subversion of the `apollo` cell creates havoc inside the `apollo` cell and allows the intruder to impersonate `apollo` principals to `lkg`, `lkg/rpc` and `mac`. This attack however does not allow principals at `lkg` or `lkg/rpc` to interact with principals at `mac`. This is because neither `lkg` nor `mac` are descendents of `apollo` — they are simply peer cells.

Subversion of the `lkg` cell will allow the attacker to impersonate any cell and principal combination to its descendent `lkg/rpc`, but it will only allow attacks on objects protected for `lkg` and `lkg/rpc` principals in the `apollo` cell.

REFERENCES

[GLP 92] Virgil Gligor, Shyh-Wei Luan, Joseph Pato, ‘‘On Inter-Realm Authentication in Large Distributed Systems’’, in *Proceedings of the 1992 IEEE Symposium on Research in Security and Privacy*, IEEE Computer Society, 1992.

[Pato 90] Joseph N. Pato, ‘‘DCE Authorization Services — Privilege Server’’, OSF DCE Specifications, 1990.

AUTHOR'S ADDRESS

Joseph N. Pato
Distributed Object Computing Program
Hewlett-Packard Co.
250 Apollo Drive
Chelmsford, MA 01824
USA

Internet email: pato@apollo.hp.com
Telephone: +1-508-436-4350