QuickSilver: Root Password Rotation as a security measure for Small/Medium Scale LANs

ARJUN VENKATRAMAN

arjun.dhanush@gmail.com

1.1 Security Threats to small/medium scale LANs

In an academic/small scale development environment, where development and training are taking place on a regular basis, it is not always feasible to prevent the users from obtaining the administrator level passwords for individual systems in order to modify configuration to suit individual project needs. This is especially true in the case of academic project development. In such a scenario if the administrator passwords are common for all the systems are common the following threats emerge

a. In the case of the password leaking through one user to multiple users, malicious or accidental harm to the system could occur.

b.If the passwords are not changed frequently, data which is meant to be confidential may be accessible by unauthorized users.

c. Passwords chosen by humans are typically insecure, since most users choose passwords which are easy to crack by brute force or standard dictionary hacks.

The primary attack pattern in such a situation is one of Social Engineering wherein the attacker takes advantage of the user's lack of knowledge to gain security sensitive information about a system.

1.2 Introduction to QuickSilver

The aim of QuickSilver is to change the administrator/root passwords for the individual systems across the network, using random password generation and assignment functions. This is done in order to circumvent the Social Engineering attack, by using its own method against it.

The password for one level of the network reside on the server one level up. The user who needs the password is required to approach the administrator of that level and verbally obtain the password. The passwords are changed every time the server boots and the individual systems download the password for the current session at each boot time.

The Superserver generates and manages the passwords for the entire network.

The passwords are randomly generated strings, to prevent cracking using simple methods.

1.3 Design Principle

In designing QuickSilver, emphasis has been placed on keeping the design simple to understand, while ensuring that it will not be easy to break into the system.

Care has also been taken to preserve the hierarchy to ensure an efficient chain of command. The purpose here is to maintain accountability. Since the passwords are unique to the individual systems, the administrator can keep track of which user has requested the password for which system.

In addition to this, the passwords are randomly generated, which ensures that common passwords are not used.

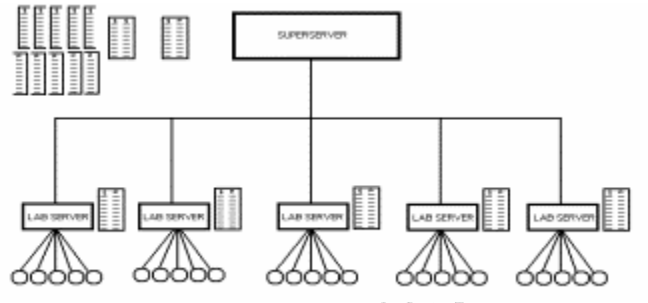
The password assignments are also random, which means that users cannot guess at the passwords.

1.4 High Level Design

QuickSilver is designed as follows

1.4.1 Tree Structure

QuickSilver segregates the entire network into a tree structure as shown below



At the top level is the Superserver. Below this, at the intermediate level are the Intermediate Servers, and finally at the lowest level are the Client Nodes. Each of these is described below.

1.4.2 Superserver

The Superserver is the controlling entity for the security of the network. It is responsible for the generation of the random passwords and storing them in groups called P-Lists as well as for the assignment of P-Lists to the Intermediate Servers for further assignments to the individual systems. The individual P-Lists are identified by unique identifiers.

The Superserver also maintains a complete map of the entire network tree.

1.4.3 Intermediate Servers

The Intermediate Servers are responsible for keeping track of each of the systems attached to them. This is done by means of data structures

called S-Lists, which are basically lists of IP addresses of the Client Nodes attached to the Intermediate Servers. Each S-List is also stored on the Superserver and is identified by the IP address of the Intermediate Server to which it belongs.

1.4.4 Client Nodes

The Client Nodes are the user level machines, which obtain their administrator password for the session at boot time from the Intermediate Server one level above.

1.4.5 Generation of S-Lists

The generation of the S-Lists takes place as follows:

1. Each node registers with the Intermediate Server directly above it.

2. The Intermediate Server enters the IP address of each of the registering nodes into a list called the S-List.

3. Each S-list is stored in a plain text file with the name of the file being the IP address of the Intermediate Server

4. The Superserver maintains an S-List of all the Intermediate Servers. In addition to this, the Superserver also maintains a copy of all the S-Lists maintained on the Intermediate Servers. This serves a dual purpose. It provides backup as well as a means for the Superserver to be aware of the entire tree below it.

A sample S-List is shown below:

```
192.168.4.2

192.168.4.1
192.168.4.3
192.168.4.4
192.168.4.5
192.168.4.6
192.168.4.7
192.168.4.8
192.168.4.9
192.168.4.10
192.168.4.11
```

1.4.6 Generation of P-Lists

The P-Lists are generated as follows:

1. The Superserver generates a set of random strings of fixed length.
2. These strings are then organized into groups called P-Lists.
3. Each P-List is then stored in a plain text file with a unique ID.
4. The random generation uses a number generation algorithm based on both the current system time as well as the process ID of the instance of the Superserver that generates the P-Lists. This is to ensure randomness.

A sample P-List is shown below:

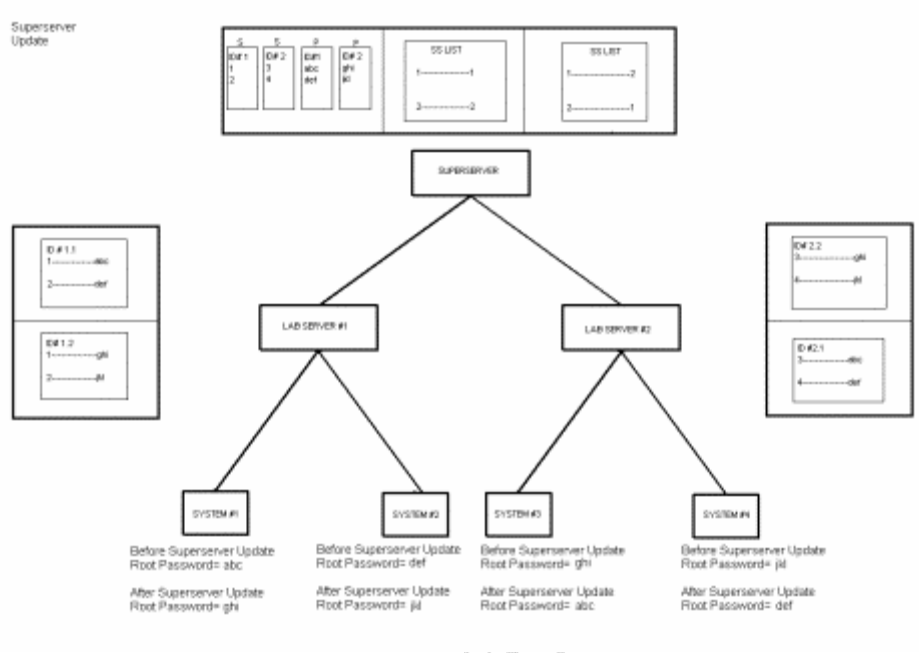
```
qpdhrwltgp mmlccbuane jkfxahrvig siaglandej otvbjsjkyl
yqqjvlgfgn hblstrpabc dzgasxmijf nkbvdcvprh wvweoiskmj
ssqmmobeuy cqyilukmpn  ybgqwmhhxp hloyusdcsf ewjgslzwnu
nuepedwevw wsmxcwfzdy gphsacctyb  pabbzvlogd ylwjxnujof
uweevgedih rgzngznldj arhvffkgzm jccqqkgagb tnkmbqcioq
cxshawmqjf yvmpycvyeh ithlwhefmj eectunbauz aokotgxvpb
xzfxezhckq gxascrqxsf chinokmrah mspjmqjmvx idxrkwsudz
roszvbcoyb byniuhlwtq xjiqfahrbg ghqlqfryii pfygcyntqx
zptcarxoym vnaklxtito ryvfwcpqod bjqovimyjt xgywtoigei
gegeruemx dcbnqmbvhm mawibskqcc ikddzyukxr silmkeqffg
```

1.4.7 Superserver Update

The Superserver Update follows the following pattern:

1. The Superserver discards the current set of P-Lists.
2. It then generates a new set of P-Lists using the method described above.
3. Each of the S-List ID's (i.e. the IP addresses of the Intermediate Servers) is entered into an array.
4. Each of the unique P-List ID's is entered into another array.
5. Both arrays are rotated by a different random number. This results in a unique mapping from S-List to P-List at each rotation.
6. This mapping is stored in a file called an S-P file.
7. The Intermediate Servers contact the Superserver at boot time to download the appropriate P-List for that session.

A schematic of the Superserver Update is shown below:



1.4.8 Intermediate Server Update

The Intermediate Server update procedure is as follows:

1. The Intermediate Server downloads the appropriate P-List from the Superserver.

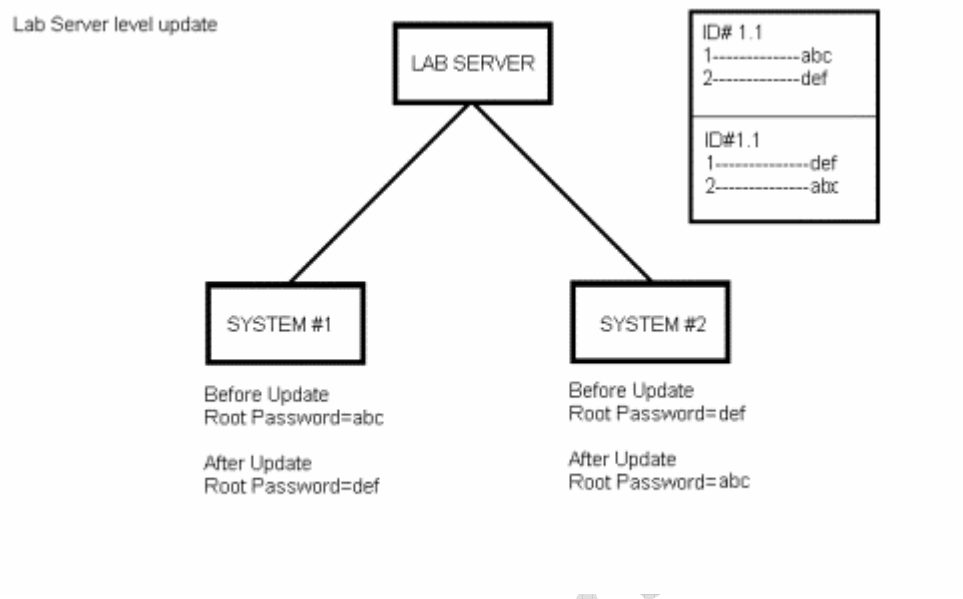
2. Each of the passwords on the P-List is entered into an array.

3. Each of the entries on the Intermediate Server's S-List (i.e. the IP addresses of the individual nodes) is entered into a second array.

4. Both the arrays are rotated by a different random number. This results in a unique mapping from nodes to passwords for each rotation.

5. The passwords are downloaded by the individual nodes at boot time.

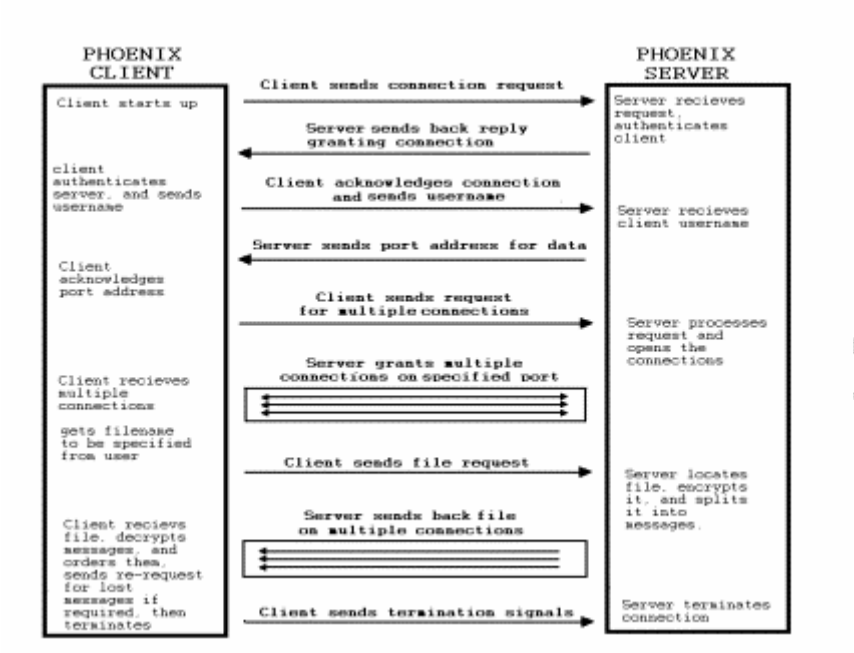
A schematic of the Intermediate Server Update is shown below:



1.4.9 Messaging and Data Transfer

The messaging and data transfer in QuickSilver takes place using a connection oriented TCP protocol, protected by the SSL encryption method. This protocol is called Phoenix and was also developed in-house. Each message is broken down into a number of parts, each of which is sent over one of three connections. The connection over which to send a particular chunk of message is selected at random. In addition to this both parties in the communication are required to have a certificate, which may be generated in house, or by a trusted third party.

A Schematic of the Messaging Module is shown below:



1.4.10 Management of Passwords

The management of passwords is the key to QuickSilver. Instead of attempting to prevent a social engineering attack, QuickSilver uses the very mechanism of the attack to provide protection. When a user requires the password to a particular system, he is required to verbally obtain it from the systems administrator. This ensures that the administrator knows which system's password has been released and also the fact that passwords are unique for each system, prevents the misuse of a common password.