....collateral....

SECURITY WHITE PAPER

Introduction

In today?s increasingly connected business environment, corporations are rushing to extend the availability of important information to mobile workers. The desire for anytime, anywhere access must be tempered with a proper appreciation of the vulnerabilities created when a user is not physically present. The problem of security can be broadly divided into two areas:

1) assuring that only legitimate users can access a system

2) making sure data is neither looked at nor tampered with as it travels across the network.

Oracle Mobile Agents Version 2.0 elegantly addresses both these issues by providing robust, flexible security features that are very easy to deploy, manage and configure. These features include:

· endpoint authentication with variable key lengths and automatic time-outs

· data stream encryption using a variety of algorithms and key lengths

· tamperproofing

· replay protection

This white paper outlines how Mobile Agents implements these points. The first is covered in the section titled, "Authentication". The other three are discussed together in the section "Data Security".A brief glossary of terms is presented first to familiarize the reader with some common security terms and how they are used in this document.

Glossary of Terms

DES

"Data Encryption Standard", an encryption algorithm developed 20 years ago by IBM and the U.S. National Security Agency. Currently in wide use as a standard. DES comes in two variations: DES-40 uses 40-bit keys, and is available for use worldwide. DES-56 uses the maximum size 56-bit keys, and is available only inside the U.S.

Fingerprint (digest)

A value produced by a digest function given a message as input, to be thought of as a compact, unique representation of the message data. Fixed-size and small (compared to the input size, which is unlimited), but designed such that even single-bit changes in the input produce radical changes in the digest. Like a checksum but better. Typically MD5 is used as a digest function, producing 128-bit digests.

RC4

A symmetric encryption algorithm owned by RSA Data Security Inc. RC4 is extremely simple and very fast, but relatively new to many security administrators.

Replay

The process whereby an attacker copies a message as it travels over the network, and then re-sends it. The attacker need not know the contents of the message or tamper with the contents for damage to be caused, so encryption and tamper-proofing do not automatically protect against replay.

RSA Public Key Cryptography

An asymmetric encryption algorithm in wide use for authentication. An endpoint encrypts with a private key that is never disclosed to anyone. Data is then decrypted with a public key which is made available to everyone. Data encrypted with the private key can only be decrypted with the public key. Similarly, data encrypted with the public key can only be decrypted with the private key.

Session key

A random pattern exchanged during initial login sequence, each time a session between two parties is established, used as a key for data encryption. Not to be confused with basic or public/private keys which are fixed unless explicitly reset.

Signature

A value included with a message, intended to validate the contents of the message. Typically, the cleartext message is digested, a session key is appended to the digest, and the concatenation is run through an encryption function (usually DES). The result is the digital signature of the message.

Signing

The process by which you create a digital signature and attach it to an authentication message. A message is "signed" using an <encryption algorithm> and a <key>.

Authentication

Overview

Authentication is the means by which Oracle Mobile Agents? endpoints (clients, agents and the Message Gateway) prove their legitimacy to one another. It is independent of data security. The algorithm used depends on the authentication policy. There are several authentication policies supported in Version 2.0, which can be broadly described as follows:

Strong Authentication

Strong authentication requires clients, agents, and the Message Gateway to mutually authenticate themselves . This authentication scheme is based on public-key cryptography, specifically RSA with 256, 512, 768 or 1024-bit keys. Each side has a public/private key pair. Public keys are exchanged, and messages are sent encrypted with the receiver’s public key. Such messages can then be decrypted with the matching private key (which is known only to the receiver). Strong authentication requires all clients and agents to be running Version 2.0 of Oracle Mobile Agents. Clients or Agents running Version 1.0 (or 1.1) are denied service.

Basic Authentication

Basic authentication also requires clients and agents to authenticate themselves to the Message Gateway. However, the Message Gateway is assumed to be legitimate and does not have to authenticate itself to the clients and agents. This authentication scheme is based on symmetric DES ("full" DES with 56-bit keys are used for authentication; the 40-bit key restriction placed on DES for use outside the U.S. and Canada applies only to encryption). One key, known to both parties, is used for both encryption and decryption. With basic authentication, all clients and agents can communicate with the gateway regardless of which Version of Mobile Agents they are running.

Compatible Authentication

Compatible authentication is the same as strong except that it allows clients and agents running Version 1.0 of the Message Manager or Agent Event Manager to connect to the Message Gateway. Version 1.0 clients and agents use basic authentication while Version 2.0 clients use strong. Compatible authentication is provided so that customers running Version 1.0 of Oracle Mobile Agents have an easy migration path if they wish to implement strong authentication when they upgrade to Version 2.0.

Configuration and Setup

Because of the different algorithms used, basic and strong authentication differ in the amount of setup required. Each Message Gateway has a security configuration. This contains information on the authentication policy, available encryption algorithm, session key, strong key length, tamperproofing status, and the Gateway?s public key. The first time a client/agent is run against a Message Gateway, it requests information about the this information and sets up security. Subsequent logins do not require an exchange of this information (unless the authentication policy has changed).

Basic Authentication Setup

Each user and agent is assigned a password, which is stored in the gateway’s user database, encrypted with DES using the username as the key. If they wish, a user can choose to store their encrypted password locally so they don?t have to enter it every time. Agent passwords are always stored since no user is present to enter a password.

When a client or agent first connects to the Message Gateway, it sends a cleartext message requesting configuration information. The Message Gateway responds with a message confirming the security configuration (basic), signed using DES and the encrypted user password. The client verifies this message, reads the configuration information, and setup is complete.

Strong Authentication Setup

When strong authentication is first set up, neither the client/agent or the Message Gateway trust each other, so it is necessary to establish trust before exchanging public keys. To do this, each client user and agent owner is assigned a one-time password, which is stored in the Message Gateway database encrypted using DES and the username. This password is given to the user by the system administrator. The setup process then commences:

: 1) Client sends an unsigned, cleartext message requesting configuration information from the Message Gateway.

2) Message Gateway returns a response, signed using DES and the one-time password, containing authentication policy, strong key length, available encryption algorithms, and the Gateway?s public key.

3) Client verifies the response and saves the Message Gateway?s public key to disk. It then generates a public/private key pair of the length specified by the Message Gateway. The private key is encrypted in a local password (referred to

as a Security Code - see below) and kept on the user's local machine. (For agents, the private key is encrypted using a password defined for the agent?s owner and stored in the agent?s directory.)

4) Client sends a login message containing their public key to the Message Gateway. The public key is attached to a login message, signed using DES and the one-time password, and sent to the Message Gateway.

5) Message Gateway validates message, the one-time password is discarded and the client?s public key stored. Setup is complete. All subsequent communication uses public key encryption.

The setup process is identical for agents, with one important difference; public/private key pairs are set up on a per agent owner, not per agent basis. In Oracle Mobile Agents, agents are owned (managed) by users, and a single user can ?own? many agents. As a result, agents share security contexts with one another, according to the users to which they are assigned. This means that agents managed by the same owner will use the same keys.

Whenever a key is imported into the gateway, distributed to the gateway, or verified by the gateway, it is checked against the minimum strong key length. If it is less than that length, the key is rejected, even if it correctly authenticates the sender.

The Security Code provides an additional level of protection for the client. It is never stored anywhere. Users must supply it every time they are authenticated. The security code prevents an attacker from stealing a user's laptop and automatically gaining access to that user's private key. If you forgets your security code, your public/private key pair becomes useless and a new key pair needs to be generated.

Logging in

When a client or agent wishes to connect to the Message Gateway, it sends a login request. All login requests are signed. The signature consists of a fingerprint of the login message appended with the session key. The algorithm and key used to encrypt the signature depends on the authentication policy. The login process is as follows:

1) client/agent generates session key

2) client/agent sends login message to Message Gateway Basic - signed using DES and the encrypted user/agent password Strong - signed using RSA and the Message Gateway?s public key

3) Gateway verifies signature, extracts session key, and sends registration response. Basic - registration response not signed. Strong - registration response signed using user/agent?s public key

4) Client/agent verifies registration response

5) Security context established and login complete (see below)

Sessions and Security contexts

The registration response contains all the data security parameters (session key, chosen encryption algorithm, tamper-proofing status, encryption status, and encryption algorithm state) for the Message Gateway. Collectively, these parameters form a "security context". The exchange of this security context establishes an implicit "session"; the session remains established until the gateway explicitly terminates it, the client shuts down, or the gateway shuts down. Note that the session is not a connection in the traditional network sense, it is simply some state shared between the gateway and the client or agent.

When a security context disappears, it must be re-established. Since security contexts are stored in memory, they are terminated when either endpoint (client, agent, or gateway) shuts down. A security context can also be explicitly terminated when its owner's authentication lifetime expires.

If the gateway receives a message but is unable to decrypt or tamper-verify it because the required security context has disappeared, it will automatically reject the message and ask that the sender authenticate themselves. After authentication, the client re-sends the message using the newly-established security context.

If the gateway wishes to send a message to an endpoint for which it has no security context, it will send a network control request to the endpoint asking it to authenticate itself. After authentication, the message is sent to the endpoint, using the newly-established security context.

Authentication lifetimes

In a connection-oriented world, a session is equivalent to a network connection; when the connection is broken, the session is terminated and the user needs to re-authenticate when they next establish a network connection.

In Oracle Mobile Agents, there are no connections. A session is a more abstract concept and it is much longer lived than in a connection-oriented world. If the client or agent and the gateway both stay running for a year, the user remains authenticated for that period of time, continuing to use the same session key. This has obvious security implications.

Oracle Mobile Agents addresses this problem by setting an "authentication lifetime". When a security context is established, its authentication lifetime timer starts ticking. When the timer reaches zero, the context is destroyed. The client or agent must then re-authenticate itself as described in the previous section. Time-outs are set in one hour increments.

Periodic re-authentication has two benefits. The first is that a new session key is generated every time the user re-authenticates. The more often a session key is changed, the harder it is to break data security. The second benefit is that someone who steals a mobile computing device will be unable to impersonate the device's owner after the authentication lifetime expires.

Data Security

Data security in Oracle Mobile Agents consists of tamper-proofing, i.e. detecting when messages are tampered with or replayed, and encryption, i.e. hiding data from view while it is in transit. Data security is transparent to the user, and incurs only minimal administrative overhead. These features are enabled or disabled as part of an overall encryption level enforced by the Message Gateway. The encryption level can be set to one of three values:

· "disabled" means neither tamper proofing nor encryption are used

· "tamper-proof" means tamper proofing is performed and encryption is disabled

· "encrypt" means both tamper proofing and encryption are performed

All data security in OMA is point-to-point. The Message Gateway, Message Manager and Agent Event Manager will always verify and/or decrypt incoming messages, store, and then tamper-proof and/or encrypt again prior to forwarding the message to its final destination. In addition, data security is enable on a per-driver basis. There are a number of reasons for per-driver and hop-to-hop data security; the most important one being that wireless networks are in general less secure than corporate LAN?s. Security may be essential for a public wireless network, but unnecessary for a secure corporate network.

Both tamper-proofing and encryption work with the Oracle Mobile Agents message compression algorithms. Compression is performed prior to encryption.

Encryption

Encryption provides the highest level of security for data in transit across a network. The encryption algorithms available depends on where the software will be used. Oracle Mobile Agents supports both DES and RC4 encryption, with key lengths suitable for use either internationally (40 bit keys) or within the U.S. and Canada (56 bit DES or 256 bit RC4). The session keys used to drive the encryption algorithms are exchanged during authentication and are randomly generated. By using authentication lifetimes, the administrator can force session keys to be re-generated at specific intervals, preventing attackers from using brute force to guess a session key.

The gateway administrator can preference-order the gateway's encryption algorithms such that the algorithm used when communicating with a particular endpoint is the first one on the gateway's list that is supported by the other endpoint. This allows a mixture of domestic and export clients to communicate with a domestic gateway, for example.

Tamperproofing

Tamper-proofing provides the next highest level of data security. It is faster than encryption because data is sent over the network in cleartext rather than encrypted form. Before sending a message to the network, the endpoint generates a fingerprint of the data. This fingerprint is then encrypted (using the session key) and sent along with the message. The receiving endpoint decrypts the fingerprint and then runs the cleartext through the same digest used by the sender. The receiver compares these two fingerprints and only accepts the message if they are the same.

Replay Protection

To protect against replays, each message is assigned a unique sequence number. Any message with a duplicate number is rejected.

.Summary

There are many degrees of flexibility built into Mobile Agents V2.0 security, allowing the system administrator to choose which security components are important to the corporation, and enforce them as required. The administrator has control over:

* allowed authentication models - the administrator can choose to between:

- Basic Security - password-based authentication.

- Strong Security -RSA Public Key Cryptography authentication.

- Compatible Security - hybrid of basic and strong based on what Version of the Message Manager the client is running.

* session key lifetime - where administrators are concerned about a session key being in use for an extended period of time, they can choose to set an authentication key lifetime.

* encryption algorithm - the administrator can decide which security algorithms are supported for a given gateway.

* encryption key length - inside the United States, the DES algorithms can support 40 or 56 bit key lengths, and RC4 algorithm can support 40 or 256 bit key lengths. Outside the United States, both algorithms are available only in 40 bit key lengths.

* minimum authentication key length -key length is critical to the integrity of the authentication process, and as such should be very large. The administrator can enforce a minimum key length anywhere between 256 and 1024 bits (512 for export).

* network links encrypted - the administrator can choose to only encrypt over selected communications links, such as radio or dialup links, and not over others, for example, an internal LAN or WAN known to be secure.

* tamper-proof protection - the administrator can choose whether messages should be protected from tampering, or being altered during transmission.

In almost every case, the choices available to the administrator come down to a trade off between increased security at the expense of performance or administrative costs. With the flexibility provided by Mobile Agents V2.0, administrators can set and enforce a security policy that meets their specific needs.