CA TECHNOLOGIES

UNIFIED INFRASTRUCTURE MANAGEMENT - NIMBUS PROBE SECURITY OVERVIEW

The NimBUS monitoring bus is comprised of three essential elements:

1. Hubs, which control monitored event aggregation and bus authentication
2. Probes, which provide various services such as event monitoring and control. Probes can both access as well as provide bus services.
3. Robots, which control probes on the system being monitored
   (At the lowest layer, all bus participants are considered probes. From a systems-design standpoint the word 'agent' can be used interchangeably with 'probe')

Identity and Authentication

Services on the bus require access privileges. In order of increasing privilege they are OPEN, READ, WRITE, ADMIN and SUPER. Each privilege assumes the previous levels. Services that are 'open' do not require authentication and include (predictably) the probe login request as well as various directory services. All other requests require bus authentication (login).

READ privilege grants access to all services protected by that privilege. WRITE is higher. ADMIN allows administrative control of all, but bus security settings. SUPER allows user creation and other high level security functions.

For probes to become full bus participants they need to authenticate to the hub. Session tickets are assigned by the hub after authentication has taken place. If the probe was installed and started by the robot, the robot first verifies the integrity of the probe binary using an HMAC. If the binary was not tampered with, a one-time session "cookie" is created by the robot and passed to the probe at startup. If the initialized probe then connects to the robot and presents the same cookie back the robot, the robot considers it to be valid probe. If the probe decides to login to the bus it contacts the local robot and the robot uses this cookie to assert the identity of the probe to the hub. The hub then uses the cookie, the probe's IP and other information to create a unique Security ID (SID) for it.

User probes are those that are not started by the robot. They authenticate to the hub using twofish message encryption over the network and on-disk twofish + MD5 / SALT techniques against a user/password database on the hub. Optionally, the hub can also be configured to route these authentication requests to an LDAP server (Active Directory and others).

Encryption

Low-level Encryption is performed at the message level on authentication and administrative requests. SID encapsulation and on-disk encryption using a combination of a symmetric algorithm (Twofish) and MD5 hashes.

Public key encryption is used in the application-layer Tunnel within the hub probe and optionally between the robot and the hub using a statically compiled version of OpenSSL.

The bus hub-to-hub Tunnel layer proxies bus requests across untrusted networks using an OpenSSL-based connection pool between trusted hubs.

Tunnel end-points as well as robot/hub connections use self-signed client and server certificates. Third party certificates can also be used (for hub-to-hub tunneling) by replacing the self-signed certificates with third-party certs in the standard OpenSSL certificate store. Please see the hub probe documentation for more info.

OpenSSH: Some of the probes (eg: RSP probe) use OpenSSH to install modules or determine system status on UNIX/Posix systems. The SSH key settings and authentication options are largely determined by the OpenSSH server daemon running on the monitored operating system. The version of libssh NimBUS uses will be generally compatible with the server version. Please verify the capabilities in the libssh version listed in the addendum with versions installed on your operating systems.

The optional LDAP authentication service uses the Novell NDK LDAP library to transmit LDAP requests between the hub and LDAP server over LDAPS connections. These connections confirm both the server and client certificates.

Library Versions: As of Robot/Hub 7.63 NimBUS uses the following library versions:

OpenSSL: 1.0.0m

OpenSSH: (libssh 0.5.3)

James Burnes

Principal Software Engineer / Security

CA Technologies