CA Unified Infrastructure Management

CA UIM Global Support Team

HUB
Performance Optimization
&
Troubleshooting Guide

January 3, 2017

Author: Gene Howard
Advisors/Editors:

Jason Allen, Steve Danseglio, David LeDeaux,
Kathryn Maguire, Paul Breheny, Daniel Groen

Legal Statement

Table of Contents

Background

This UIM HUB performance optimization and troubleshooting guide fosters a deeper understanding of the CA UIM HUB and how it operates. It also covers how to optimize performance, and how to troubleshoot HUB issues. The troubleshooting sections include a standard troubleshooting checklist of tasks as well as offering up over ***10*** scenarios describing HUB issues and how to resolve them.

This guide also contains some extracts from key sections of the CA UIM Help documentation, internal documentation, backline and development resources, or Knowledge Articles that are useful for understanding the HUB and how to optimize performance and troubleshoot issues. That said, it is also important to note that all of the information in this document has been reviewed, tested, reassessed, and vetted for accuracy.
For the complete CA UIM HUB help documentation, search for the HUB documentation at:

https://docops.ca.com/dosearchsite.action?queryString=HUB&startIndex=0&where=UIMPGA

This guide is based on the HUB version for UIM 8.4. The Help documentation and Release Notes contain important information and should be reviewed in full. Moreover, valuable information on HUB installation, scalability and compatibility can be found in the Downloads section of the CA UIM support site at the links below:

https://docops.ca.com/ca-unified-infrastructure-management-probes/en

Please also refer to the Compatibility Support Matrix for the latest information on supported OS, databases and platforms.

Introduction

The Hub is the core of UIM communication and Security. When the hub is not working properly we can see issues from being unable to login to IM and Admin Console or UMP, to messages, alerts, QOS, communication not getting processed as they should. Because of the distributed nature of the hub architecture this could mean all of the UIM / Nimsoft infrastructure is down or just one or more segments being down.

The hub has 3 Main responsivities.

- **Handling security**
  - This includes the validating the 3 different types of users
    - NMS users
    - Account Users
    - LDAP users
  - Probe validation
  - Login for IM, Admin_console and UMP

- **Handling communication and messages between robots, probes and other hubs.**
  - Keep alive checks on robots
  - Collect QOS and Alarm information from robots
  - Forward and receive message to and from other hubs
  - Processing QOS and Alarms messages from local probes.

- **Queue management**
  - Processing of incoming and outgoing messages through queues

Technical Overview
Communication

The hub probe itself uses the following ports and protocols:

Port	Protocol	Description	Details
48001	TCP	Spooler	Receives bus messages from local robots and probes.
48002	TCP/UDP	Hub	Main control port for hub communication - TCP is used for callbacks and UDP is used for hub-to-hub broadcasts
48003	TCP	Tunnel Server	Control port for Tunnel server (configurable by end-user) - tunnel clients must connect to this port

Ports and Protocols

The HUB makes TCP connections and requires certain ports/protocols including the following:

Primary and Secondary hub communication:

Hubs discover other hubs by sending out broadcast (UDP) messages. Non-primary hubs that are separated from the primary hub by routers or firewalls cannot discover other hubs over UDP. To allow hub to talk to each other that are on separate networks that do not all broadcast forwarding but have ***NO port / Communication restrictions*** a static route can be setup in the hub GUI.

Static routes are used to:

- Connect two hubs that are in the same environment, that reside on different network subnets.
- Connect to a hub outside a firewall so that you can create a secure tunnel to the hub.

Important! Do not connect a hub with both a tunnel and a static route.
In some situations, data can be transmitted over the insecure static route rather than over the secure tunnel. Delete static routes that are used to configure a tunnel after the tunnel has been established and is working. Do not retain static routes when tunnels exist.

The hub listens on ***port 48002*** for local request coming into the hub. These request come from the following sources

- robots checking in with the hub or establishing and sending messages.
- Connection from Infrastructure Manager, Admin_Console and UMP for login request.
- Hub to Hub communication not involving a tunnel
    - Security file synchronization
    - Hub.sds routing table information.
- Acts as the spooler probe for local probes installed on the hub.

When the hub is part of Tunnel connection additional ports are used to traffic over the SSL tunnel. By default, this is **port 48003** but can be changed to a custom port. The hub still uses port 48002 for local communication to robots and probes but to access remote sites all traffic is channeled over the Tunnel port.

NOTE: *It is **NOT** expected that you will have both direct connection AND tunnel connection between the same two hubs. This will cause unexpected behavior and is not a support configuration.*
When using LDAP the hub will communicate with the LDAP server on ports 389 or 639. These can be configured in the HUB.cfg

When the hub is communicating with Robots and Probes it is expected to be able to make direct connection to the **IPaddress:Port** for each Robot/ Probe unless Robot Proxy mode is in use.

Hub and Robot Communication

Set the Hub Communication Mode

Hub-managed components use the hub SSL mode. The hub SSL mode is primarily used for *robot-to-hub* communication. When hubs are *not* connected by tunnels, the hub SSL mode is also used for *hub-to-hub* communication.

SSL communication is enabled through the `UIM_HOME/robot/robot.pem` certificate file. The controller creates this file during startup. The file contains the key to decode encrypted CA UIM messages.

Important: The robustness of SSL communication is improved in v7.70.

Before, in a non-tunneled domain, hubs that are configured for unencrypted communication can decode encrypted messages.

In a multiple hub domain, upgrading to v7.70 does not allow this scenario. See, Impact of Hub SSL Mode When Upgrading Nontunneled Hubs in the Hub (hub) Release Notes.

Note: Any tunnels set up between hubs remain after you upgrade, and communication will continue.

We strongly recommend that you connect all hubs with tunnels.

To set the communication mode:

1. In Infrastructure Manager, expand the hub robot, and open the hub probe in the configuration GUI. Select the **General** tab, and click **Settings** in the lower right corner.
2. Select the **SSL** tab.
3. Select the mode. UIM hubs have three communication mode options:
    a. **Normal** SSL mode 0 — Unencrypted The `OpenSSL` transport layer is not used
    b. **Compatibility mode** SSL mode 1 — The hub and robot to communicate without encryption or with `OpenSSL` encryption. Components first attempt to use SSL. If a request is not acknowledged, the component sends unencrypted requests.
    c. **SSL Only** SSL mode 2 — `OpenSSL` encryption only
4. Save the configuration.

Wherever possible, we recommend that you use mode 1, compatibility mode (See Note 2 below for performance impact). Compatibility mode enables secure communication between the components that support SSL, and unsecure communication for any other components.

Note 1: Hub v7.80 supports the TLS protocol by using TLS cipher suites for tunnels between hubs, and hub-to-robot SSL settings.

- To restrict tunnel communication to TLS cipher suites, upgrade the hubs to v7.80. Select a cipher suite that resolves to TLS. CIPHER_SUITE_NAMES
- To use TLS with hubs that are at v7.71 and earlier, use a cipher suite resolving to TLS *and* SSLv3.
- To use a TLS cipher suite for hub-to-robot SSL settings, use a cipher suite resolving to TLS *and* SSLv3.

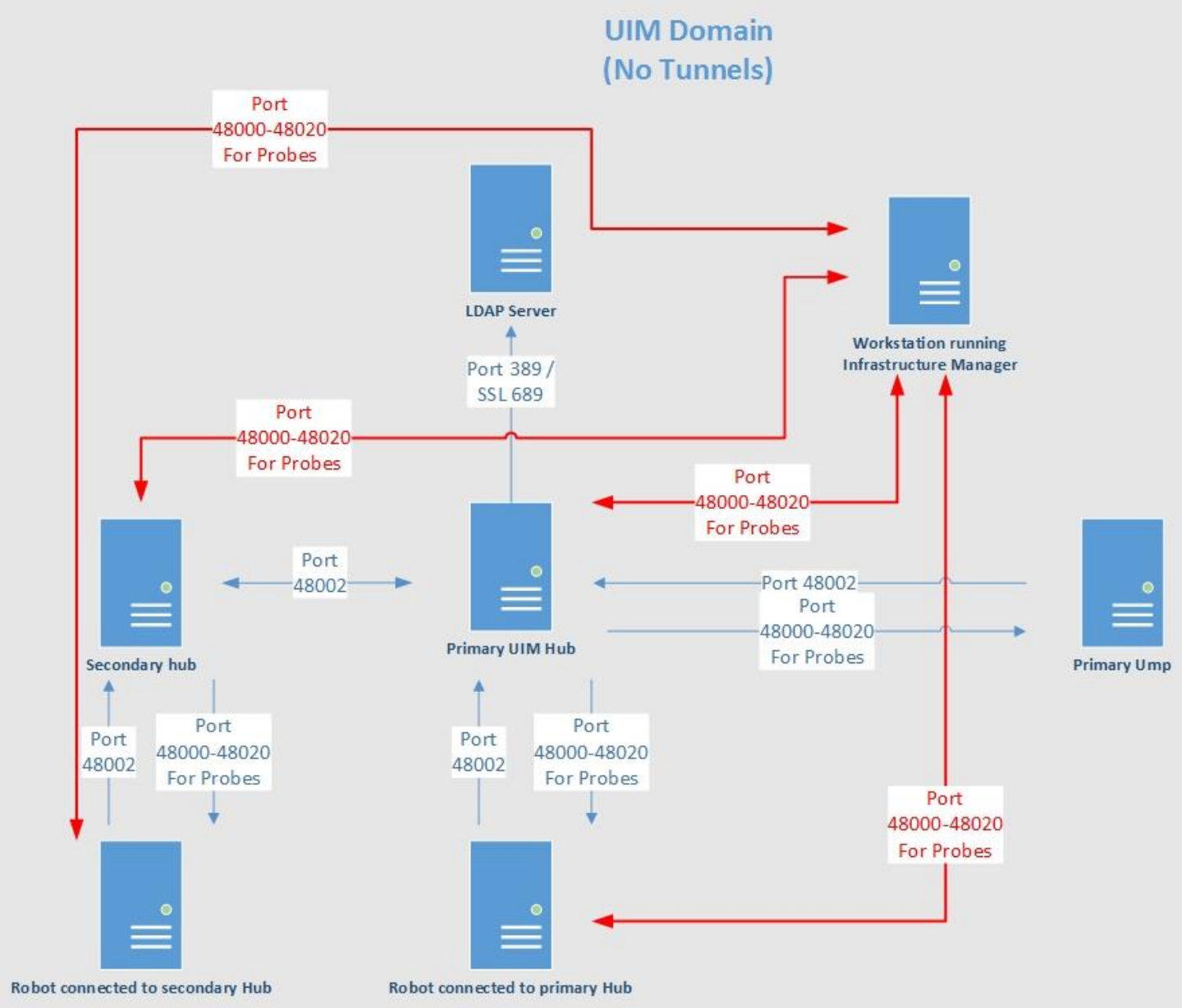
Restart the tunnel server and tunnel clients when:

- The tunnel server cipher suite is changed
- The tunnel server hub is reverted to a prior release and the tunnel clients are using a TLS cipher suite

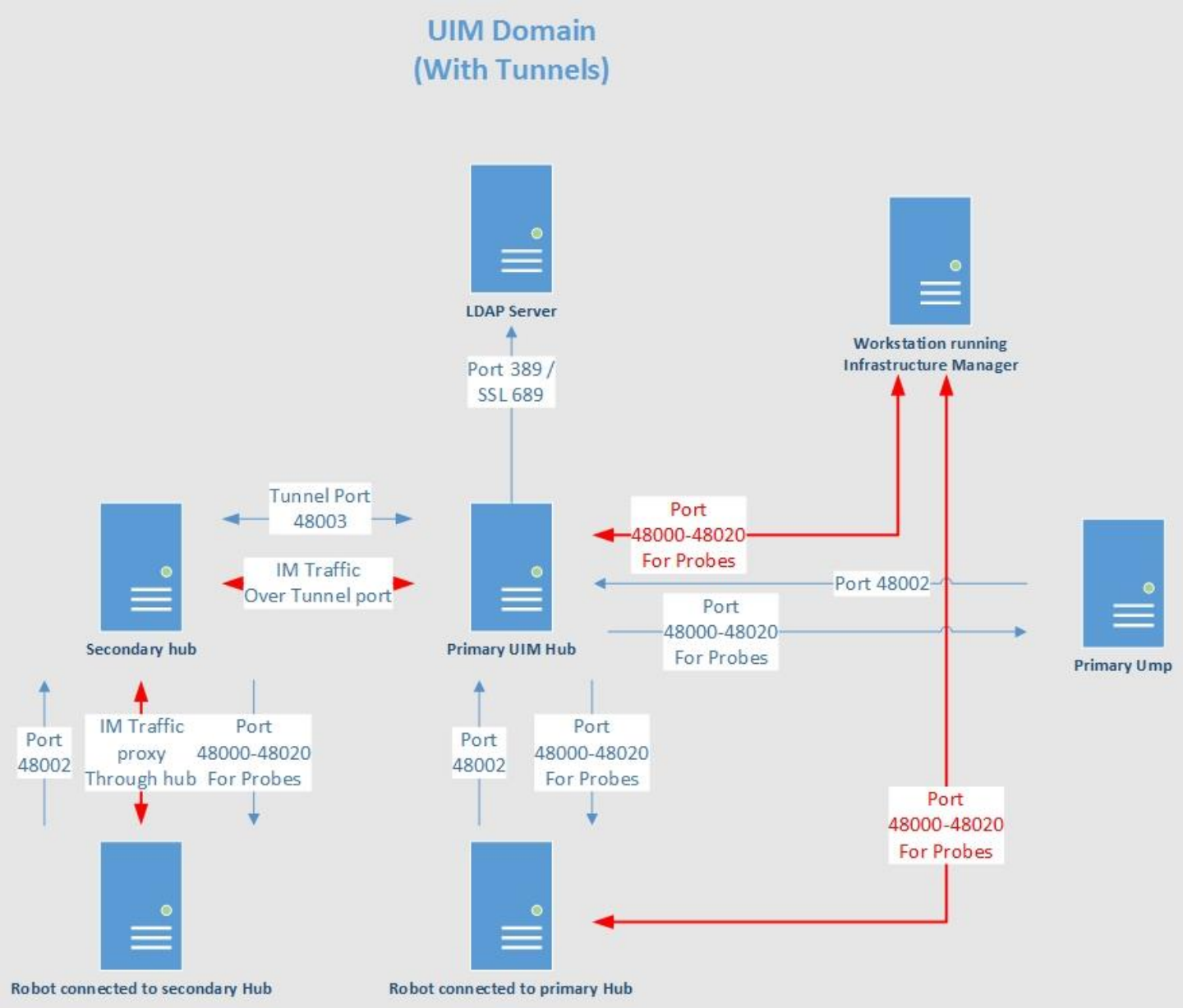
Note 2: The ***higher*** the encryption level the ***slow*** the hub will be able to transfer messages.

Hub Connections and Ports Diagram

Hub Communication and Ports with no Tunnels.



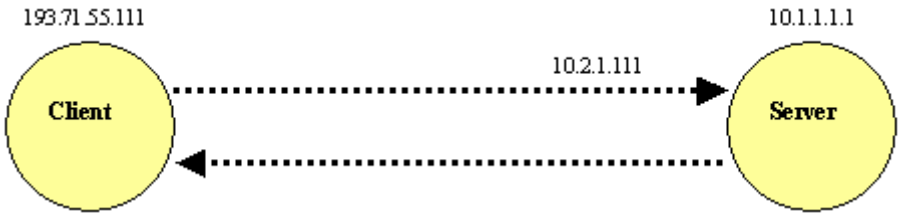
Hub Communication and Ports with Tunnels.



Hub Communication with Tunnel in a NAT Environment

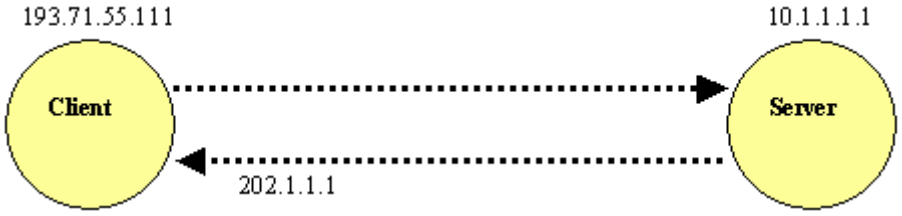
Networks that use Network Address Translation (NAT) affect how a tunnel is configured.
The following scenarios describe three possible configurations.
Important! When a tunnel is configured, the tunnel replaces the static hub and NAT setup in the hub configuration.

Client address in NAT environment



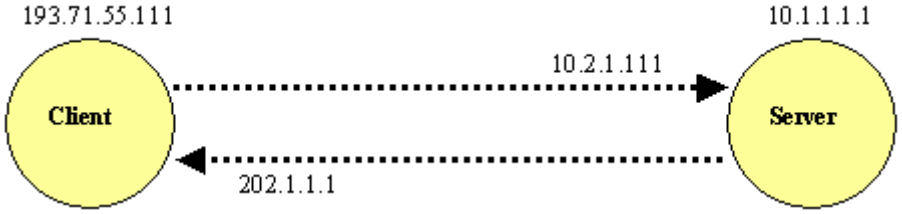
The client certificate must be issued to the common name that is visible to the server. In this case, that is `10.2.1.111`, and *not* `193.71.55.111`.

Server address in NAT environment



Clear **Check Server Common Name** in **Tunnel Client Setup** to disable server common name checking. The client sees `202.1.1.1`, but the server certificate contains the common name `10.1.1.1`. If server common name checking is enabled, the communication fails.

Server and Client addresses in NAT environment



Combine the two previous methods:
The client certificate must be issued to the common name that is visible to the server.
Clear **Check Server Common Name** in **Tunnel Client Setup** to disable server common name checking.

Firewall rules

Ports needed to be open between Primary UIM Hub and UMP Bidirectionally

- From Hub to ump
  - Ports 48000 – 48020 (all ports used by probes)
  - Ports 80 /443 (Port used by UMP)

- From UMP to HUB
  - 48002 (Hub probe)
  - 4334 (Udm_manager)
  - 8182 (Relationship_service)

Additional ports needed to be open on the primary hub

- 8080, or 8443; (Service_host / admin console)

Ports needed between primary hub and local robots

- From hub to robot
  - Ports 48000 – 48020 (all ports used by probes)
- From robot to hub
  - Port 48000-48002

  Additionally, if there are internal firewalls between hubs,robots, we recommend opening the port range 48000-49000 locally (internal traffic.) This will allow the robots and hubs to freely communicate with each other and allow for future probe expansion and tunnel session traffic.

*Ports needed to be open between hubs **without** tunnels*

- Ports 48000 – 48020 (all ports used by probes)

*Ports needed to be open between hubs **with** tunnels*

- Ports 48003 (The tunnel port setup on the Tunnel Server.)

On a tunnel server:

TCP Port 48003 must be opened for inbound traffic. It is important that the port is not opened only to HTTPS or SSL - it must be raw TCP.

On a tunnel client:

No inbound TCP ports must be opened, but outbound traffic to the tunnel server on port 48003 must be allowed, and again this must be raw TCP, not limited to HTTPS or SSL.

Firewall General Information:

Firewall configuration can be a complex topic which is sometimes difficult to understand and troubleshoot. At the core, the configuration required for hub communication (including tunnels) is relatively simple; however, firewalls themselves are not always so simple and there is a lot of potential for confusion.
It is generally the policy of CA Support that we are not able to advise on most specifics of firewall configuration, as that is beyond our area of expertise. However, we can attempt to assist to the best of our ability with general configuration advice which should be carried out by a customer's network team.
Additionally, if there are internal firewalls between hubs,robots, we recommend opening the port range 48000-49000 locally (internal traffic.) This will allow the robots and hubs to freely communicate with each other and allow for future probe expansion and tunnel session traffic.
We have also observed some communication issues related to protocol inspection engines and SSL proxies. It is absolutely imperative that such devices be disabled for the tunnel traffic. The UIM tunnel traffic looks like 98% correct SSL and so the better network devices will recognize that and after a couple seconds to a minute or two generate enough detected errors in the formatting that it will see the connection as an intrusion attempt and shut it down. This is especially common if a customer is using port 443 for their tunnels, as many firewalls apply an extra level of scrutiny to port 443 by default.

A note on Stateful Firewalls

Many firewalls in use today are known as "Stateful" firewalls, indicating that they keep track of the state of TCP conversations/sessions.
These firewalls can cause instability with the hub (especially interacting with it in AC/IM) when they are set to automatically close idle TCP sessions, or block traffic which is considered to be part of the "wrong" TCP conversation.
Such "stateful" firewalls should be configured to have the longest possible session timeout (unlimited, if possible) and to be "non-application-aware", meaning that they do not try to associate specific sessions and conversations with specific applications.

SSL Decryption:

SSL Decryption is an option on many firewalls; it works by creating separate encrypted connections to the client and server so that the encrypted traffic can be decrypted and scanned before being re-encrypted and passed on. This option will cause UIM tunnels to fail and must be disabled.

Specific Firewalls and Configuration Options

The following section contains information regarding specific firewall configurations that we have encountered in the field, and will be updated as more information is gained.

SonicWall:

Some SonicWall firewalls offer "port scan protection" features. This should be disabled where possible, as when the hub starts up, it immediately creates several local sessions on consecutively-numbered ports, and sometimes the firewall will mistakenly detect this activity as a port scan.

Juniper SRX:

Juniper SRX firewalls allow an "inactivity-timeout" setting to be specified per application; for the UIM hub traffic this timeout should be set to "never" in order to avoid the firewall expiring sessions that the hub was intending to re-use.

Palo Alto:

Palo Alto firewalls are stateful/state-aware firewalls and should be configured as follows:
Idle connection timeout: 7200 -- this should be considered a minimum, higher is always better
Max connection timeout: 7200 -- this should be considered a minimum, higher is always better
TCP Connection only : No application awareness (important)

Stonesoft:

Stonesoft firewalls are similar to Palo Alto firewalls in this regard and should be configured as described in the Palo Alto section above.

Sophos:

"Stateful Packet Inspection" should be disabled for the UIM hub traffic. This may apply to other firewall types as well.

How Communication works for Infrastructure manager:

Suppose that a user who is using the Infrastructure Manager client logged into the Primary Hub wants to configure a probe on the robot with IP 192.168.0.4, which reports to the secondary hub at 192.168.0.2; the hubs are not connected via a tunnel, but a static hub connection (or they are on the same subnet and have discovered each other via the hub broadcast mechanism.)

The sequence of events that will occur when a user attempts to configure this robot will be as follows:

1. The IM Client/GUI performs a "nametoip" callback against the primary hub, providing the address of the desired remote robot/probe as a parameter.
2. The primary hub checks its routing table to see if it has cached the requested address.
3. If not, the hub breaks the address down into its component parts, e.g. /Domain, /Hub, /Robot.
4. It checks to see if the given Domain/Hub is one that it knows about, and if so, forwards the nametoip request to that hub.
5. The given hub returns the IP and Port of the robot and the desired probe (e.g. 192.168.0.4:48002)
6. The IM Client/GUI attempts to connect directly to the given IP:port to communicate with the desired probe.

In other words, in our example, the TCP connection will be from the IM Client itself directly to the appropriate port on 192.168.0.4, and the hubs will be bypassed entirely. This is a bit counterintuitive – a common assumption that is often made is that if a robot reports to a secondary/remote hub, the communication to that robot must pass through that hub; this is, however, an incorrect assumption as when tunnels are not in play, the hub actually only provides a location for the client to connect to, it does not route the information itself.

NOTE: Please see diagram above showing communication paths for Infrastructure manager!

Special Notes about LDAP

- Currently the hub ***requires*** that the LDAP ***allow anonymous binds*** to do query / lookups
  - If this is disabled LDAP integration will fail.
- The Container for the LDAP groups used in UIM should have **less than 100 Groups** in the container or you will get an error
- User must be direct members of a group UIM does not support nested groups.
  - Authentication will fail if in a nested group
- User can only belong to ***ONE LDAP*** UIM group. UIM currently does not have any order of precedence.
  - As the order in which the ldap groups cannot be guaranteed the resulting ACL rights may very

Hub Security

Authentication order Process for Nimsoft –> LDAP

When a login attempt comes into the hub the following checks are done

- Can the user be found in the security.cfg?
- Can the user be found in the cm_account table?
- Can the user be found in the LDAP directory?

Once the user is found then a check is done to see what ACL is used and what rights are to be provided.

Authentication order process for LDAP –> Nimsoft

When a login attempt comes into the hub the following checks are done

- Can the user be found in the LDAP directory?
- Can the user be found in the security.cfg?
- Can the user be found in the cm_account table?

Once the user is found then a check is done to see what ACL is used and what rights are to be provided.
If the user is an account user or an LDAP user tied to an account ACL the origin information is restricted based on the account setup.

Queue Types, Queue Management and Messages

Type of Queues

Get / Attach Queues

- The Get queue should be done upstream from the hub so that if there is a problem connecting an alarm is generated and received.
- Get queues retrieve information from attach queues on other hubs
  - This is the side that set the bulksize on to adjust the message flow rate.
- Attach queues collect messages that either a get queue will get or a probe will get.

Post Queues.

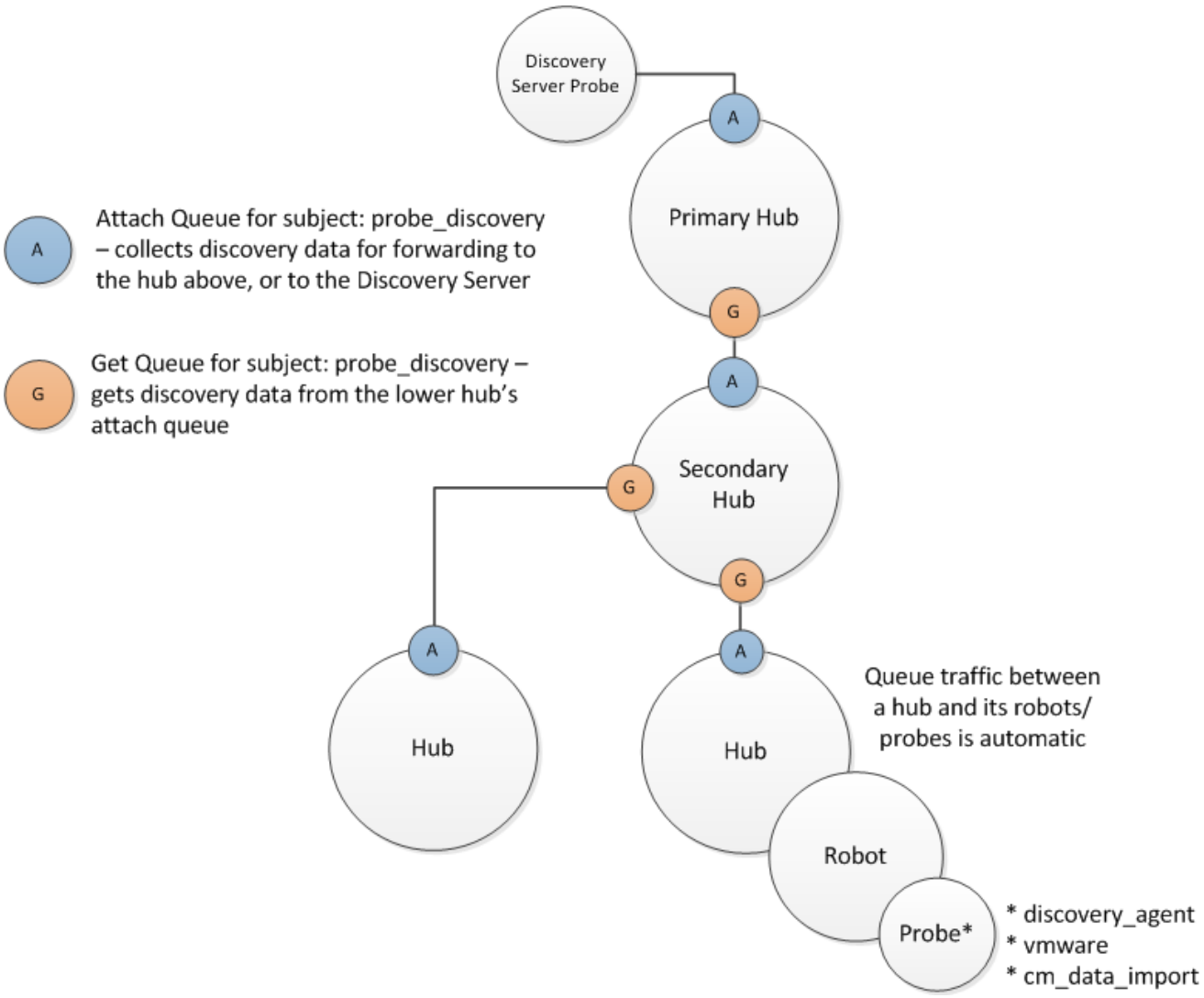
- Generally, not recommended by support as these are similar to UDP Communication, in that there is no confirmation that message was received.
- Only supported way to send messages between UIM domains as of 7.80 HUB.

Queue Setup

- Support suggest that on each upstream hub there be ***two GET queue*** to pull Alarms from one queue and ***ALL other messages*** from another attach queue on the downstream hub.
  (except when using NAS replication. In this case there should be no get queue for alarms.)
- By default, as of hub 7.8X, most queues now default to 1000 bulk size. This can be over written and changed as needed in the hub.cfg. If there is a backup in a downstream attach queue this number can be raised as high as 9999.
  *(Rising it this too high **may negatively impact** your primary server and back end database server. It is better to raise this number slowly until the downstream hub is draining and then allow it to finish draining over time rather than over burdening your primary server and or database)*
- Probes that deal with alarms messages and updates usually can only retrieve one message at a time and process them. Examples of these are:
  - NAS
  - Smtpgtw

  (Currently there is no way to increase the speed at which these probes process messages)

Example Queue setup:



Default Message Type

- alarm (Alarm messages sent directly from probe)
- alarm1 (used by EMS /Alarm Routing services)
- alarm2 (messages forwarded from Alarm_enrichment to nas)
- alarm_close(used by fault_correlation_engine)
- alarm_new (used by fault_correlation_engine)
- alarm_update (used by fault_correlation_engine)
- audit (Message sent by controller when changes are made to configurations)
- BASELINE_CONFIG (QOS message used by Basline_engine)
- event (used by the new EMS probe)
- enriched_events (used by the new EMS probe)
- interface_poller(used by fault_correlation_engine)
- legacy_alarm (used by EMS /Alarm Routing services)
- PREDICTION_CONFIG (used by prediction engine)
- probe_discovery (messages sent by new CTD graph probes such as snmpscollector and vmware)
- qos_message (QOS messages sent by all proves
- qos_definition (QOS definition messages sent by all probes)
- QOS_BASELINE (QOS message used by Basline_engine)
- TOT_RULE_CONFIG (used by the new EMS probe)
- TOT_RULE_CONFIG (used by the new EMS probe)
- udm_inventory (Used by discovery_Server and udm_manager)

Hub Subscriber limits:

This is covered in details in the following KB article that is kept up to date on this topic: tec000004446

Windows OS

The hub can only handle a maximum of 64 subscribers in a Windows environment due to Windows OS (winsock) limitations on the number of open sockets. So, if a hub has more than 64 subscribers (i.e. more than 64 queues, (which includes temp queues) then the hub will start to service the queues in a "round robin" fashion. This means one queue will get temporarily dropped while another one is picked up, then that one gets dropped and another one is picked up, etc. This results in two undesirable effects. One effect is that the hub's overall performance will be seriously degraded. Another effect is that if the queue from a probe gets dropped and the probe notices this, it would cause the probe to restart itself in an attempt to re-establish its queue.
Windows only allows you to WaitForMultipleObjects on 64 sockets, after which you have to split this information in two or more lists and WaitForMultipleIObjects on each of them. This causes an unacceptable delay in the message flow of messages on sockets not in the first list. Unix select() does not have this limitation, and so the limit is not relevant in Linux/UNIX.

Linux/UNIX OS

The number of subscribers for Linux/Unix hubs is only limited by available resources. Note that in Linux/UNIX, once again, temp queues count as subscribers. Note that subscriber_max_threshold doesn't really matter for UNIX/Linux OS's as it does not suffer from the Windows limitation of WaitForMultipleObjects on 64 sockets. Though, on Linux it's been seen before that if it runs out of sockets and or open files, open files must be increased to mitigate.

You can increase the open file limit on the server. Normally open files limit is set to 1024. On Linux open files configuration can be checked with 'ulimit -a'

ulimit -a // to check the system limits especially on open files. default is 1024 which hub could be exceeding

lsof // get total number of open files while issue is occurring to make sure it's not the open file limit issue

Consult with a Linux admin if you need to increase the number of open files.

Checking the number of subscribers

Select the hub probe and hit Ctrl-p to open the hub probe utility and press the green arrow button to execute the callback list_subscribers to see the current total subscribers number at the bottom of the subscribers section of the resultant output - Note you MUST add 1 more subscriber since the list starts at 0.

For example:
Probe : /NMS/NMS-Server/NMS-Robot/hub
Address: 10.xxx.xxx.xx
Command: list_subscribers

```
subscribers    <TABLE START>
  0         -
    name      10.xxx.xxx.xx/64248
    subject   _$HUBALL,alarm_stats
    count     1
    inq       0
    establishe1395840520
    postroute t_41
  1         -
    name      10.xxx.xxx.xx/64227
```

```
   subject   alarm_new,alarm_update,alarm_assign,ala
   count     1
   inq       0
   establishe1395840519
   postroute t_40
2          -
 etc
 etc...
```

Note that the list of subscribers ends right <u>before</u> the section:

postroute <TABLE START>

Alarming on the number of subscribers
Using the hub Raw Configure you can add a new key and value to set the configuration parameter in the hub section to alarm when the number of subscribers reaches a specific value (see below).

subscriber_max_threshold
Specify the number of subscribers that must be reached to generate an alarm, e.g., 54.

subscriber_max_severity
Specify the severity of the max subscriber alarm above. This value is numeric.
Alarm severity values:

5 is **Critical**
4 is **Major**
3 is **Minor**
2 is **Warning**
1 is **Informational**

Hub configuration settings:

<hub> Section:

postroute_reply_timeout

This value is also in seconds, and decides how long the hub will wait for a reply from any queue/subscriber after sending messages. **The default** = 180. Suggested setting for low volume traffic/queues: 300. Note that if the hub bulk size for the data_engine is high, e.g., 2000, then you can set ***'postroute_reply_timeout'*** to **300** to make sure the hub waits long enough to send all of its messages. Otherwise if the timeout is reached, the queue is disconnected and reconnected but may enter a loop trying to process the same messages, and no alarm is thrown warning of this condition. Instead the queue may change to yellow status (unknown) and stop processing QOS messages.

postroute_passive_timeout

This value is also in seconds, and decides how long the hub will let the queue be passive, e.g., there's no traffic before disconnecting it. **The default** = 300. Suggested setting for low volume traffic/queues is 420 but check the nas Status window->Alarm History transaction and filter on *Queue* and apply the filter. Then double-click on each queue alarm to see how long it is taking between when its generated and when it clears and then consider adding some more time (seconds) to the value. For example, if it the Queue alarm is generated and cleared within 3 minutes (180 secs) then set '***postroute_passive_timeout'*** to 300 seconds.

nametoip_forward_timeout

You can set the amount of time it takes for a nametoip request to timeout in the <setup> section of the hub.cfg: nametoip_forward_timeout, **the default** is 2.

subscriber_max_threshold

Specify the number of subscribers that must be reached to generate an alarm, e.g., 54.

subscriber_max_severity

Specify the severity of the max subscriber alarm above. This value is numeric.

<Tunnel> Section:

protocol_mode

Default value is 0 when key is not present. When set to 1, the hub will revert to hub 7.71 tunnel behavior; by default, the "fixed" tunnel algorithm from hub 7.72 is used.
hub 7.80HF9 defaults to 7.72 tunnel behavior.
It can be put into a slightly better version of the original 7.80 behavior by setting

protocol_mode *= 1*

in the /tunnel section of raw config.
delete the key entirely to revert to 7.72 behavior.

<ldap>\<server>

retries_count

Retry count when proxying via another LDAP enabled hub. Default: 3

retries_sleep

Sleep time in ms between each retry when proxying via another LDAP enabled hub Default: 1000

Timeout

Timeout in seconds on LDAP requests or Nimbus requests when proxying via another LDAP enabled hub. Default: 15

Best Practices for optimizing Hub performance

General hub / tunnel performance tuning.

- Update to the latest version of robot and hub on **ALL** of your hub servers
  (We strangle recommend keeping all hubs the same version)
  - Check the Nimsoft Archive here.
  - Check the Hotfix page here.
- Make sure you have applied the hub 7.x hub and tunnel settings:
  - postroute_interval = 120
    postroute_reply_timeout = 180
    postroute_passive_timeout = 300
    hub_request_timeout = 120
    tunnel_hang_timeout = 300
    tunnel_hang_retries = 3
  - Full details are here.

Queue tuning:

Queues are controlled by two things:

- Speed of the network connection
- Bulk size on the get or post side of the queue.

Speed of the network connection

To check your network speed in the hub GUI you can right click on a hub and to two options

- Response check
- Transfer check

Notes:

- The higher the number the better.
- When using tunnels this number will go down as all of the traffic is encrypted.
- If the transfer rate is less then 250KBs this may cause a problem with message transfer

Client will need to work with network team to address bandwidth and latency issues.

Bulk size on the **GET** or **POST** side of the queue

- Pre 7.X hubs had a default Bulk size of 100 Post hub 7.x this was raised to 1000
- In most cases this can be left at 1000.
- In busy environments this can be raised incrementally up to 9999
  (There is no hardcoded limit but support has seen performance dope off after this point)
- The bulk size can be set when creating GET or POST queues are edited after the fact using raw config.

General UIM troubleshooting steps:

For the given UIM version, please also check the version of the Hub and check associated probe docs such as, What's New, Release Notes, Known Issues and perform a search for the issue/error/return code, using relevant keywords at searchit.ca.com, findit.ca.com and/or check the CA community.

https://docops.ca.com/display/UIM84/What's+New
https://docops.ca.com/display/UIM84/Known+Issues
http://searchit.ca.com
http://findit.ca.com
http://communities.ca.com
https://docops.ca.com/ca-unified-infrastructure-management-probes/en/alphabetical-probe-articles/data_engine/data_engine-versions-8-0-8-3/v8-0-data_engine-im-configuration
https://docops.ca.com/ca-unified-infrastructure-management-probes/en/alphabetical-probe-articles/hub/hub-troubleshooting

Hub Specific trouble shooting steps:

NametoIP check:

The most useful tool for troubleshooting IM connectivity is the hub callback 'nametoip'. What you want to do is point the probe utility at the hub probe on the hub that you are logged into via IM (e.g. the primary hub). Then do a "nametoip" callback and put in the full /Domain/hub/robot/controller address for the controller probe on the 3rd-tier hub. You will get back an IP:port combination -- if there is a tunnel connection being used you'll get the ip:port of a tunnel server/client endpoint. If it is direct connectivity you will get the ip:port of the actual controller e.g. port 48000. Got to make sure you can telnet to the ip:port that is returned. Then repeat this test, but use /Domain/hub/robot/hub and ensure that it resolves to the same tunnel endpoint, or same IP but port 48002 and test connectivity to that. If this all comes out as expected, and telnet works, the next thing I'd try is putting them on hub 7.72 all around. Usually this is only necessary with tunnels but may help.

Log level and settings:

Generally, for the first round of trouble shooting the following should be done:

- Hub loglevel set to 3 and logsize set to 25000
- Controller loglevel 3 and logsize to 5000
  (if the problem is between two hubs both hubs should have this set)
- Reproduce the issue and collect the following from both hubs if needed.
  - robot.cfg
  - controller.log
  - _controller.log
  - Hub.cfg
  - hub.log
  - _hub.log

Telnet is your Friend!

The use of telnet between robots and hubs can be a very quick way to narrow down a communication / tunnel issue.
For example, making sure you can only connect on the tunnel port 48003 and not 48002.
This will usually lead you to a problem with configuration and or firewalls causing an issue.

Example commands:

Format of command

telnet <IPADDRESS _Connecting_Too> <Port_Conneting_on>

For direct connection between two hubs or from robot to hub:

telnet 192.168.1.1 48002

For connection to robot

telnet 192.168.1.2 48000

For connection between two hubs using a tunnel.

telnet 192.168.1.3 48003

Troubleshooting scenarios

Scenario 1: Tunnel will not connect due to bad password

Problem:

New tunnel will not connect between hubs or after updating a certificate the tunnel will not connect.

Example log:

May 28 06:42:14:231 [3492] hub: Could not read private key file
May 28 06:42:14:231 [3492] hub: [1] error:0x06065064:digital envelope routines:EVP_DecryptFinal_ex:bad decrypt
May 28 06:42:14:231 [3492] hub: [2] error:0x0906A065:PEM routines:PEM_do_header:bad decrypt
May 28 06:42:14:231 [3492] hub: [3] error:0x140B0009:SSL routines:SSL_CTX_use_PrivateKey_file:PEM lib
May 28 06:42:14:231 [3492] hub: CORE failed to create SSL context for client

==========
May 28 06:42:14:231 [3492] hub: internal alarm - Password error. Invalid or missing certificate password. Reason: Failed to decrypt certificate, 4, 121.244.218.14
May 28 06:42:14:231 [3492] hub: CORE added SESSCTRL for 121.244.218.14/48003
May 28 06:42:14:231 [3492] hub: CORE - setup complete, starting to work
May 28 06:42:14:731 [2072] hub: dist_route - no receiver dropping RN38167534-00001 (sub:alarm s:671)

Solution:

Try entering the password again created during the Cert creation.
If that does not work, you will need to recreate the cert and password.

Scenario 2: Tunnel will not connect due to bad cert file

Problem:

After creating a tunnel cert and putting it in place the tunnel will not connect.

Example log:

Jul 26 14:37:45:078 [2380] hub: Could not read private key file
Jul 26 14:37:45:078 [2380] hub: [1] error:0x06065064:digital envelope routines:EVP_DecryptFinal_ex:bad decrypt
Jul 26 14:37:45:078 [2380] hub: [2] error:0x0906A065:PEM routines:PEM_do_header:bad decrypt
Jul 26 14:37:45:078 [2380] hub: [3] error:0x140B0009:SSL routines:SSL_CTX_use_PrivateKey_file:PEM lib
Jul 26 14:37:45:078 [2380] hub: CORE failed to create SSL context for client

Solution:

Create a new Cert and password and setup the tunnel again.

Scenario 3: Tunnel will not stay connected to do SSl error 5

Problem:

After setting up the hub tunnel and certificate we cannot get the tunnel to connect.
Or the hub will connect for a short time and then disconnect or show red in Infrastructure manager:

Example log:

May 9 15:23:33:849 [2832] hub: SSL handshake start from 69.176.98.24/48003: before/connect initialization
May 9 15:23:33:849 [2832] hub: SSL state (connect): before/connect initialization
May 9 15:23:33:849 [2832] hub: SSL state (connect): SSLv3 write client hello A
May 9 15:23:33:880 [2832] hub: ssl_connect - SSL_connect error (5) on new SSL connection
May 9 15:23:33:880 [2832] hub: SSL_connect error occured
May 9 15:23:33:880 [2832] hub: TSESS could not connect to tunnel 69.176.98.24:48003 (0)
May 9 15:23:33:880 [2832] hub: CTRL could not connect to server 69.176.98.24/48003

- You are able to telnet to the proper IP address and port
- The wire-shark trace looks normal.
- Nothing wrong with trace route that you can see

The server side Hub logs show:

May 9 15:09:38:129 [140089799726848] hub: TSESS-A-47-124 session looping (60) wait_time is now: 605
May 9 15:09:38:199 [140090869274368] hub: SSL handshake start from 209.249.244.5/63959: before/accept initialization
May 9 15:09:38:199 [140090869274368] hub: SSL state (accept): before/accept initialization
May 9 15:09:38:205 [140090051352320] hub: Sent heartbeat on queue route 'Audit_to_On-Prem'
May 9 15:09:38:211 [140090869274368] hub: SSL alert (write): fatal: handshake failure
May 9 15:09:38:211 [140090869274368] hub: ssl_server_wait - SSL_accept error (1) on new SSL connection: 209.249.244.5
May 9 15:09:38:211 [140090869274368] hub: [1] error:0x1408A0C1:SSL routines:SSL3_GET_CLIENT_HELLO:no shared cipher

Solution:

- The problem is external to Nimsoft and could be seen on any hub version using a tunnel
- We found out that there was a setting on the Client firewall called 'SSL Decryption' that was taking the certificate that we were sending out, trying to decrypt it and then re-encrypt it and sending it out.
- As soon as the customer removed that setting or added an exception for our two servers we were able to establish a tunnel connection.

Scenario 4: LDAP user cannot login to Infrastructure manager.

Problem:

When an LDAP user ID and password are used instead of an NMS user they cannot login to Infrastructure manager. The username and password have been confirmed.

Example log:

APR 28 huhti 2015 21:06:42,602 ERROR [NmsAuth:401] Login failed for ***user@domain.com***: javax.security.auth.login.FailedLoginException: (12) login failed, Received status (12) on response (in sendRcvLogin) for cmd = 'login'
hub
Apr 28 21:13:09:236 [1852] hub: verify login - cmd=gethubs frm=xxx.xxx.xxx.xxx/64309 failed
Apr 28 21:13:09:280 [1852] hub: login [LDAP] - invalid credentials
Apr 28 21:13:09:281 [1852] hub: login [NimBUS] - wrong password user= ***user@domain.com*** ip= xxx.xxx.xxx.xxx
Apr 28 21:13:09:291 [1852] hub: verify login - cmd=gethubs frm=10.40.191.66/64314 failed
Apr 28 21:13:09:335 [1852] hub: login [LDAP] - invalid credentials
Apr 28 21:13:09:336 [1852] hub: login [NimBUS] - wrong password user= ***user@domain.com*** ip= xxx.xxx.xxx.xxx
Apr 28 21:13:24:452 [1852] hub: login [LDAP] - (logon_user) ***0 user found*** for (&(objectClass=person)(|(userPrincipalName= ***user@domain.com***)(sAMAccountName= ***user@domain.com***))), do not know which to use.

Solution:

Usually this can be fixed by using Raw configuation on the primary hub to change the following in the ldap section:

From:

format = $username@$domain

To:

format = $username

Scenario 5: LDAP users' groups cannot be listed

Problem:

When editing ACL and trying to connect the ACL to and LDAP group you get a message such as:
"cannot list LDAP groups"

Example log:

N/A this is a popup message seen in the User interface.

Solution:

Make sure that the group you add to AD is truly a FLAT AD group. There cannot be more than 100 groups in the groups container pointed to by the following key in the hub.cfg:

<ldap>
<server>
base = ou=groups,ou=domain,o=com

Nested groups are currently NOT supported. 'Groups within groups' or AD referrals are not supported in any manner. A *nested* group in this context means:

- LDAP Group A Exists
- Individual User is a direct member of LDAP Group A
- LDAP Group B Also Exists
- LDAP Group A is added to LDAP Group B's membership, so that users who are members of A are now also 'indirectly' members of B.

IMPORTANT: Currently you cannot use *nested* groups/users within an AD group otherwise it will not work correctly. You MUST use just one 'FLAT' group with your Nimsoft 'admin' users in it - don't 'nest' sub-groups with users in it as it will not work.

When nested groups are used, associating the ACL with a given login user will not work since the hub doesn't see/treat the user as a direct member of the group. The user must be a direct member of a single flat group meaning that the group would be listed in the Active Directory "MemberOf" attribute

Scenario 6: Cannot connect to LDAP server

Problem:

Cannot connect to LDAP server for some reason. Test fails

Example log:

Line 72258: Nov 9 08:51:18:491 [11220] hub: do_ldap_query [LDAP] - query failed: ldap_search_ext_s: 'Can't contact LDAP server' (81)
Line 77988: Nov 9 08:51:48:895 [9484] hub: do_ldap_query [LDAP] - query failed: ldap_search_ext_s: 'Can't contact LDAP server' (81)
Line 77997: Nov 9 08:51:48:903 [10972] hub: do_ldap_query [LDAP] - query failed: ldap_search_ext_s: 'Can't contact LDAP server' (81)
Line 78004: Nov 9 08:51:48:913 [10728] hub: do_ldap_query [LDAP] - query failed: ldap_search_ext_s: 'Can't contact LDAP server' (81)
or
Jun 9 10:37:53:913 [6180] hub: (nim_ldap_get_connection) host 'USCDCDCU02.us.cbre.net' port '3268'
Jun 9 10:37:54:020 [6180] hub: login [LDAP] - basic login took 111 ms
Jun 9 10:37:54:020 [6180] hub: login [LDAP] - auth failed: 'Invalid credentials' (49)

Solution:

Check loglevel 3 logs for the hub.
Check for entries such as:

- *'Can't contact LDAP server' (81)*
- *auth failed: 'Invalid credentials' (49)*

These are codes returned directly from the LDAP server.
You can check for a list of LDAP error code here
Work with your LDAP admin to check the LDAP configuration information setup in the hub.

Scenario 7: Queue(s) backing up.

Problem:

Queues backing up will usually mean the upstream hub or Probe is not getting the information quick enough to keep up
With the incoming message rate.

Example log:

N/A

Solution:

It is best to check the following for attach queues used between hubs.

- Check the hub scream hub get queue for errors on loglevel 3
- Check hub transfer rate between hubs and address network speed with network team if at or below 250KBs
- Check the hub bulk size on the get side and increase as needed.
- Check using dr. nimbus to see if a probe has been misconfigured and is flooding the system with QOS or alarms

For queues that are emptied by probes such as the probDiscovery queue and the emailgtw queue it is best to do the following:

- Set the problem probe loglevel to 5 and logsize to 25000
- Set the hub loglevel to 3 and logsize to 25000
- If the probe has a memory setting, try increasing it.
- Deactivate the probe and activate the probe and check the probe and hub logs for errors to try and narrow down the problem.
- Search the Knowledge Base for any errors found.

Scenario 8: Cannot see directly connected hub in IM.

Problem:

We have installed a new hub that is on the same network as the primary but it is not showing up.

Example log:

NA

Solution:

This is usually caused by the hub not being able to communicate with UDP broadcasting.
Connect to each hub via IP address from Infrastructure manager and set up the named IP on the named services tab
In the hub GUI.

Also check both hubs robot.cfg and hub.cfg and make sure the domain is exactly the same. All sections of the UIM address space are case sensitive.

/domain/hub/robotname/probe

Scenario 9: Multiple Tier Environments - instability and probe GUI problems

Problem:

Customers who have 3 or more "tiers" of hubs in their environment - meaning a top-tier hub which has a tunnel connection to a secondary hub which in turn has a tunnel connection to a third hub - may experience instability. This usually manifests as hubs unexpectedly restarting themselves or briefly disappearing/turning red in Infrastructure Manager.

Example log:

None

Solution:

Hub 7.72 resolves this issue for many customers; however, it will introduce a new problematic behavior in some circumstances which should be noted. Specifically, for probes such as CDM, interface_traffic, database-related probes, and other probes where the GUI in Infrastructure Manager populates a lot of "current" data and checkpoint status indicators, these GUIs may fail to open, giving an error like "Communication Failed (80040402)." Hub 7.80 will resolve that problem, but at the expense of stability, so there is a trade-off.
As of this writing (Oct 2016), the development team has not yet released a hub build that fixes both issues, but it is in progress.
NOTE: Updating to the latest 7.8X Hot fix may also help.

Scenario 10: File Descriptor Usage Spikes

Problem:

In some Linux environments (usually those with a large number of tunnel clients), a hub (usually a tunnel server) may stop responding and freeze or hang until the Nimbus Service is restarted. In such cases, the output of "lsof" or similar tools will show that the hub is using a large number (over 1024) of file descriptors, and it releases these descriptors when restarted.

Example log:

None

Solution:

Work is currently (February 2016) in progress to investigate and resolve this problem, which has to do with the way the hub accounts for its active tunnel sessions. For now, the therapeutic restart option to restart the hub on an automated/timed basis is a good workaround - many customers find that setting the hub to auto-restart every 24 hours will mitigate the problem.
There is also an issue that has been discovered with baseline_engine 2.60 (and prior) and prediction_engine 1.31 (and prior) which will manifest in a similar way - these probes do not properly manage their queue subscriptions which leads to the hub consuming file descriptors over time. This is corrected in the baseline_engine and prediction_engine probes that will be released with UIM 8.4; meanwhile, hotfixes are available on the UIMfiles website.

Scenario 11: UIM hub tunnel disconnects after a very short time and will not reconnect until the hub is restarted

Problem:

After successfully connecting a new UIM hub (7.x or later) with an SSL tunnel, the hub will be seen to turn red in Infrastructure Manager, and cannot be communicated with. The hub will not recover until the entire service is manually restarted.

Example log:

NA

Solution:

Cause:
The root cause is a session inactivity timeout set at the firewall level.

For the Juniper SRX firewall, this is controlled by the "inactivity-timeout" keyword in the firewall's application configuration rules. The default (if no inactivity-timeout) is 30 seconds, but this may be configured to a higher value. Other firewalls may have similar default values.

The UIM hub manages the suspension and timeout of its own sessions, and session management at the firewall level can interfere with this process.

Resolution:
The resolution is to set the inactivity timeout to "never" for the UIM-related sessions.

An example of this configuration for the Juniper SRX would be:

```
#
# Allow UIM Tunnel Server Traffic
#
application uim-tunnel {
   protocol tcp;
   destination-port 48003;
   inactivity-timeout never;
}
```

tec1729447

Scenario 12: Hub 7.X Crashes periodically.

Problem:

The hub is crashing you can see the hub crash in the event logs on windows:

Example log:

From the windows application log:

Information 9/6/2016 7:04:02 AM Windows Error Reporting 1001 None "Fault bucket , type 0
Event Name: APPCRASH
Response: Not available
Cab Id: 0

Problem signature:
P1: hub.exe
P2: 0.0.0.0
P3: 573b222a
P4: StackHash_69ee
P5: 6.2.9200.21815
P6: 56eafa87
P7: c0000374
P8: PCH_86_FROM_ntdll+0x0000000000002C6A
P9:
P10:

Attached files:

These files may be available here:
C:\ProgramData\Microsoft\Windows\WER\ReportQueue\AppCrash_hub.exe_f9496090287a75333e5dd357a99c6729ba41f118_252e6958

Analysis symbol:
Rechecking for solution: 0
Report Id: 9e271386-7421-11e6-9434-00505698293a
Report Status: 2
Hashed bucket: "
Information 9/6/2016 7:04:02 AM Windows Error Reporting 1001 None "Fault bucket , type 0
Event Name: APPCRASH
Response: Not available
Cab Id: 0

Problem signature:
P1: hub.exe
P2: 0.0.0.0
P3: 573b222a
P4: StackHash_69ee
P5: 6.2.9200.21815
P6: 56eafa87
P7: c0000374
P8: PCH_86_FROM_ntdll+0x0000000000002C6A
P9:

P10:

Attached files:

These files may be available here:
C:\ProgramData\Microsoft\Windows\WER\ReportQueue\AppCrash_hub.exe_f9496090287a75333e5dd357a99c6729ba41f118_252e6958

Analysis symbol:
Rechecking for solution: 0
Report Id: 9e271386-7421-11e6-9434-00505698293a
Report Status: 4
Hashed bucket: "
Error 9/6/2016 7:04:02 AM Application Error 1000 (100) "Faulting application name: hub.exe, version: 0.0.0.0, time stamp: 0x573b222a
Faulting module name: ntdll.dll, version: 6.2.9200.21815, time stamp: 0x56eafa87
Exception code: 0xc0000374
Fault offset: 0x00000000000eac95
Faulting process id: 0x1f68
Faulting application start time: 0x01d2082dafe21725
Faulting application path: f:\Program Files (x86)\Nimsoft\hub\hub.exe
Faulting module path: C:\Windows\SYSTEM32\ntdll.dll
Report Id: 9e271386-7421-11e6-9434-00505698293a
Faulting package full name:
Faulting package-relative application ID: "
Information 9/6/2016 6:59:04 AM Windows Error Reporting 1001 None "Fault bucket, type 0
Event Name: APPCRASH
Response: Not available
Cab Id: 0

Solution:

This issue was resolved with the following:

- Update to the latest hub hotfix version
- Disable antivirus / exclude the entire Nimsoft directory from ANY kind of scan.
- Set the hub timeout settings higher:
    - remote_nametoip_ttl = 60 --> set this to 300
      postroute_interval = 30 ---> set this to 60
      postroute_passive_timeout = 30 --> set this to 300
      postroute_reply_timeout = 60 --> set this to 180

Scenario 13: I am unable to open any probe GUI on the robot through IM.

Problem:

I am unable to open any probe GUI on the robot through IM

No direct connection is available in between the Primary HUB Server and the robot.
Direct connection is available in between one secondary HUB Server and the robot.
The robot is visible and status is green in IM application.
IM application log in to the Primary HUB Server.

Example log:

Unable to reach controller, node
/aaa/bbb/ccc/ddd/controller
error message: communication error

Solution:

Please configure HUB Tunnel communication in between the Primary HUB Server and the secondary HUB Server so that the secondary HUB server will proxy all the necessary communication to the robot.

Important: The steps will vary depends on your environment.

Note: IM is attempting to communicate the robot directly if there is no HUB Tunnel communications for routing.

Advanced Hub probe troubleshooting

How to setup nas to dump log files when an error occurs

http://www.ca.com/us/support/ca-support-online/product-content/knowledgebase-articles/tec000003963.aspx

The article mentioned above represents two intermittent and random issues a) an example of a data_engine issue reported by a customer who received data_engine alarms (ORA-00604 and ORA-04031), every Saturday evening between 10:50 and 11:30 PM, e.g., Mar 16 23:01:28:245 [4632] de: data_engine [LSV] [LSV] - Oracle_Database::ExecuteSP [LSV] data_engine [LSV] status: -1
OCI_ERROR - ORA-00604: error occurred at recursive SQL level 1, Oracle was used as the backend database for CA UIM; and b) another situation where the customer needed to capture hub logs when their remote tunnel dropped its connection randomly.

Note that the script and Auto Operator profile included in the article can be used to capture log files for any probe and any reason, especially when it is difficult to catch the problem when it occurs. Since the trigger is the alarm message text in this case, defining an accurate message filter (regex) is the key.

Appendix

Hub Broadcasts

The hub, by default, attempts to aggressively broadcast (via UDP) its presence in order to locate other hubs. In most cases this leads to desirable behavior - hubs installed locally in a domain will automatically be found by other hubs and begin sychronizing their security files. However, in some cases it can lead to undesirable behavior - for example, hubs which are installed locally but have tunnels between them due to specific security/routing requirements may receive broadcasts from other local hubs and attempt to bypass the tunnels or use unexpected routes for hub traffic.

In such cases, broadcasts can be disabled via the Advanced Settings dialog in the Hub GUI, or through a raw configure key located in the <hub> section of the hub.cfg:

broadcast_on = no

Note that if broadcast is disabled via this option, new hubs will not be automatically discovered and customers will need to use Static Hub entries in the Name Services section of the hub configuration.