

## Instant Messaging Diagnostic Scenario

1. *Header information.* The industry segment is the broad category of instant messaging, which includes one-to-one, chat, image and file transfers, and some with voice and/or video services (person to person and person to automated agent, bots for various functions – in some ways just an alternative to information lookup, but many have used it). Most instant messaging systems are deployed on a wide array of client platforms. Initial implementations were created in the academia, but now the majority comes from AOL, MSN and Yahoo. There are a few other substantial efforts such as ICQ and IRC. The adoption and use of instant messaging has exploded during the last 4 years and has become a valuable productivity tool although there is some dispute. The issues of cross platform integration are in the forefront of any present activity, but once there is a single interoperability standard, operation and diagnostics will be important issues.

The following scenario focuses on the aspects of diagnosing a problem within a campus-wide messaging infrastructure. Items to note are, no common information base that all tools can glean information from, authorization problems with key staff, lack of data that could reduce the problem resolution time, and a void of focused reporting and diagnosis tools that a wide range of users can operate.

A major educational institution is operating a large scale messaging system for a campus population of thirty thousand faculty, students and staff spread over 4 campuses. Instant messaging has become a staple of the computational environment and is widely deployed. The primary users are students, but it is also widely used by faculty and staff. One segment of the staff population, the medical center, has begun to use instant messaging in a pilot deployment with the emergency desk and a chat channel to physicians with enabled PDA's on call within the hospital. The client software was developed by the campus IT staff, and can also communicate with other major messaging systems as well.

<i>User:</i>	Campus IT administration or operation group.
<i>Technical buyer:</i>	Campus IT research and development group.
<i>Economic buyer:</i>	CIO, VP, Provost or Director of the IT wing of campus.

## 2. *A Day in the Life (Before)*

*The idea here is to describe a situation in which the user is stuck, with significant consequences for the economic buyer. The elements you need to capture are five:*

- *Scene or situation:* The campus NOC is operating normally indicating no major problems on a morning during the middle of the fall term. Its responsibility is not only monitoring the network, but core services such as

the email, file sharing, VoIP, video, Kerberos, virus and worm activity, web, library and instant messaging services.

- *Moment of frustration:* At approximately 11 am, the help desk begins to see a steady increase of users from the medical center that are experiencing instant messaging problems. The problem manifests itself as end users dropping their client connection, and then reconnecting every sixty seconds. Also, the “buddy lists” seem to come and go as well as members of the lists both connect and reconnect. Since the messaging infrastructure consists of five front-end servers and two backend LDAP servers, the NOC staff tries to connect using a remote shell (SSH) into one of the front-end servers and begins to examine the log files from the messaging daemons, but does not have access to these files. The NOC begins to try the same process on the remaining four servers but does not have access. The NOC tries to use the instant messaging service itself and it seems to be operating normally. Thirty minutes have elapsed since the incident was first reported. Still the help desk continues to log problems and now notices that others besides the medical staff are seeing similar problems as well, therefore the help desk decides to escalate the problem to the network staff, since they believe it is isolated to the medical center. The instant messaging service is a highly interactive service, which usually is the first to indicate a network problem. Sixty minutes have elapsed since the incident was first reported. The network engineers begin their diagnosis process which involves running iperfs(1) and inspecting Netflow audit logs between the medical center networks and the front-end servers. When completed, all tests are within operating specifications. The network switch fabric and front-end host and LDAP server network interfaces are inspected using SNMP, but all are within specifications. The help desk continues to log problems from the medical staff who are now are highly frustrated. Ninety minutes have elapsed since the incident was first reported. The key instant messaging architects are called in to help solve the problem. They begin their diagnosis by isolating testing to one front end server at a time, and find that front-end #2 is the culprit, and take it out of rotation immediately, which resolves the help desk issues of users calling, but the service now has slight performance problem due to one server being removed. All logs are inspected thoroughly and still a problem cannot be found. When front-end #2 host is tested while not in rotation, it seems to be operating normally. It is decided to increase instant messaging debugging levels and return front-end #2 back into rotation to try to get more detailed data. This was initially turned off because of the large amount of data that it produced. It has been 150 minutes since the problem was first discovered. Now with increased logging detail on front-end #2, it now suggests that the response time from when a client first attaches to the front-end server to the time it gets the buddy connectivity information varies greatly. The nominal average is 40ms, and it is now 20sec, ±10 sec. Front-end server #2 is now removed

from rotation again and diff(1) is run on all files pertinent to the instant messaging system with a reference configuration. It is discovered that the file /etc/im/local.conf had the host entries for the backend LDAP servers changed from their IP addresses to hostnames, which caused the system to do getserverbyname(2) calls to the DNS servers for each reference. This not only caused large delays since this part of the code was referenced hundreds of times each second, but also overloaded the DNS servers. It was also discovered that the modification time of the file change coincided with a new novice system administrator who was logged in to the system doing maintenance. The server file management system is configured to notify staff when it detects a change in the configuration files, not to overwrite them, but it was not seen since the owner of this does not arrive at work until late in the afternoon. The file is modified, the server is put back into rotation and a meeting is set up with the system administrator who modified the file.

- *Desired outcome:* The problem is detected quickly, managers of the service fix the problem and begin a postmortem process to decide if changes in the management process of the service need to be made. The first line of support (NOC) needs to have enough data to take the culprit server out of rotation quickly, then give the experts of the service enough data to solve the problem quickly and return the server into rotation. This will reduce the frustration of the medical staff and the students.
- *Attempted approach:* Ad hoc stabs at guessing at the problem with out access to key information or knowledge of the service. Trying to retrieve log data they didn't have access to. Bringing experts in to solve the problem.
- *Interfering factors:* Key staff do not have access to vital data. Problems with both authorization based and detailed debugging data does not exist due to space concerns. Also, other subsystems do not aggregate events to a common area.
- *Economic consequences:* Key employees diverted from their normal tasks to solve the problem, which costs real dollars. The trust of the medical centers staff and administration is lost because it doesn't seem that the campus central IT staff can maintain a messaging service. The medical center may opt to build and maintain their own system, which has its own set of economic consequences.

### 3. *A Day in the Life (After)*

*Now the idea is to take the same situation, and the same desired outcome, but to replay the scenario with the new technology in place. Here you just need to capture three elements:*

- *New approach:* At 8:30am the NOC gets alarm from a tool that monitors the transaction/second from the DNS servers indicating that they are experiencing a 20% increase in load. The NOC

operator begins to analyze the problem. Then at approximately 9am, the help desk begins report a steady increase of users from the medical center that are experiencing instant messaging problems. The problem manifests itself as end users dropping their client connection, and then reconnecting every sixty seconds. Also, the “buddy lists” seem to come and go as well as members of the lists both connect and reconnect. The NOC operator then accesses a reporting tool that gleans its information from a wide variety of data sources (system, application and network logs). They first use a graphic tool that looks at the historical transactions/second of the service of a week ago, and note a 20% reduction in transactions. They then notice that the problems started during the early morning hours, and then became steadily worse. The operator then compares each of the five front-end servers and notices that the transactions/second of front-end host #2 is greatly reduced. They take the host out of rotation, and then call the administrator of the messaging service. The administrator of the service brings up an event query tool and queries it for any warnings or errors of any network-based events with regards to front-end host #2, which returns nothing. They then modify the query to show transaction times from the front-end #2 server to the back-end LDAP server within the last hour and compare them with yesterdays data. *Note: even though this type of information a large amount of data, the log archive system maintains 4 days worth before removing it. Also, an administrator has access to this data even though it has sensitive user data but has been anonymized.* The NOC calls the administrator of the system again and mentions that they have discovered two more artifacts since the first phone call. One; the campus DNS servers are getting an unusually high number of requests from the front-end host #2 and there was a warning last night that the file distribution system (Depot) did not update the file /etc/im/local.conf because it was modified manually. The administrator looks at the file and notices the problem, edits the file and notifies the NOC to put the host back into rotation. The NOC verifies that the load to the DNS servers has subsided and the normal transaction rate of the service is nominal again. The administrator has verified that the transaction times to the back-end LDAP servers are normal again. The NOC then sends mail to the managers of the service outlining the event. Forty-five minutes has elapsed since the help desk logged the first user call.

At the weekly meeting of the operations managers, a postmortem of the event is performed. The administrator that made the change to the file did insert an informational event into the log system to track his changes. It is suggested that CMS should log changes into

the event system. The managers conclude that the content distribution system should force a copy, the event should be logged as an error and that a new monitoring tool should be created to graph and alert staff if backend transaction times are out of spec. Since all tools pull from a common data store and similar tools exist to monitor other systems, building both tools take about an hour.

- *Enabling factors:* Large data set to pull information from, that can be queried with a single method and a rich API to build tools from?
- *Economic rewards:* Highly efficient operations staff that is proactive. Trust from other groups within the organization.