User Interface - Feature #3236

implement port range support for the web client's embedded web server

02/01/2017 03:10 PM - Greg Shah

Status:	Closed	**Start date:**	
Priority:	Normal	**Due date:**	
Assignee:	Sergey Ivanovskiy	**% Done:**	100%
Category:		**Estimated time:**	0.00 hour
Target version:			
billable:	No	**vendor_id:**	GCD

Description

Related issues:

Related to User Interface - Feature #2683: reverse proxy implementation	**Closed**	
Related to User Interface - Bug #3304: Eliminate denial of service in virtual...	**New**	**06/26/2017**

History

#1 - 02/01/2017 03:12 PM - Greg Shah

I am looking into enabling the "port range" idea for launching web clients. Today we support a hard coded client web port or a dynamic mode (port==0) where the port is assigned randomly. Supporting a port range would be an additional mode where the client's web port would be selected from specific pre-set range of ports. This would enable a firewall to be used for better security and should also make it easier to implement a transparent reverse proxy.

To accomplish this, we would need to track the ports from that range that are in use. This means we need to know when the client's embedded web server is shut down. A good place to do this is in ClientCore.start() where we execute this code:

```
if (!running)
{
            driver.shutdown();
        }
```

The problem here is that this is called after we call session.terminate() (closing the session with the server) which means we cannot notify the server that the port is free.

I would like to move the embedded web server shutdown to be before the session.terminate(). Logically, I think it is safe to do. My simple test shows this works OK. The only thing I have found is that the client log now has some extra entries:

java.lang.InterruptedException
at java.lang.Object.wait(Native Method)
at java.lang.Object.wait(Object.java:502)
at com.goldencode.p2j.ui.client.driver.web.PushMessagesWorker.waitForMessages(PushMessagesWorker.java:139)
at com.goldencode.p2j.ui.client.driver.web.PushMessagesWorker.run(PushMessagesWorker.java:96)
java.lang.InterruptedException
at java.lang.Object.wait(Native Method)
at java.lang.Object.wait(Object.java:502)
at com.goldencode.p2j.ui.client.driver.web.WebTaskWorker.run(WebTaskWorker.java:62)
Feb 01, 2017 12:43:17 PM org.eclipse.jetty.server.AbstractConnector doStop
INFO: Stopped ServerConnector@27fe3806{SSL,[ssl, http/1.1]}{localhost:0}
Feb 01, 2017 12:43:17 PM org.eclipse.jetty.server.handler.ContextHandler doStop
INFO: Stopped
o.e.j.s.h.ContextHandler@4d1b0d2a{/common,jar:file:/home/ges/projects/app/poc/deploy/lib/p2j.jar!/com/goldencode/p2j/ui/client/driver/web/res,UNAVAILABLE}
Feb 01, 2017 12:43:17 PM org.eclipse.jetty.server.handler.ContextHandler doStop
INFO: Stopped
o.e.j.s.h.ContextHandler@5c072e3f{/client,jar:file:/home/ges/projects/app/poc/deploy/lib/p2j.jar!/com/goldencode/p2j/ui/client/gui/driver/web/res,UNAVAILABLE}
Feb 01, 2017 12:43:17 PM org.eclipse.jetty.server.handler.ContextHandler doStop
INFO: Stopped o.e.j.s.h.ContextHandler@10a035a0{/,null,UNAVAILABLE}

#2 - 02/01/2017 03:13 PM - Greg Shah

From Constantin:

> What if the client dies unexpectedly and the server is never notified the port is free? This can lead to exhaustion of the avail ports pretty fast...
>
> Why not track at server-side which ports are in-use, and when the client's session terminates (on server-side), add that port back to the pool?

#3 - 02/01/2017 03:13 PM - Greg Shah

> From Constantin:
>
> > What if the client dies unexpectedly and the server is never notified the port is free? This can lead to exhaustion of the avail ports pretty fast...
> >
> > Why not track at server-side which ports are in-use, and when the client's session terminates (on server-side), add that port back to the pool?

That is my plan. The problem with the current order is that the session would go away first, so we would put the port back into the pool before the client's embedded web server has a chance to shutdown. I'm trying to eliminate that race condition by inverting the order of shutdown on the client side.

#4 - 02/01/2017 03:14 PM - Greg Shah

Constantin wrote:

> Ah, OK, then it makes sense.

#5 - 02/01/2017 03:15 PM - Greg Shah

Sergey wrote:

> I think these logs show the correct sequence of shutdown() method invocation:
>
> ```
> @Override
> public void shutdown() {
> // close the emulated windows first
> super.shutdown();
>
> // notify the remote client about application exit
> ```

```
websock.quit();

// since the application will exit shutdown the server
shutdownServer();
}
```

where websock.quit() invokes System.exit(0) on receiving MSG_QUIT from the websocket client and two thead PushMessageWorker and WebTaskWorker are interrupted their WAIT states.

#6 - 02/01/2017 03:16 PM - Greg Shah

where websock.quit() invokes System.exit(0) on receiving MSG_QUIT

I don't see that WebClientProtocol.quit() calls System.exit(). Where do you see this code?

#7 - 02/01/2017 03:18 PM - Greg Shah

Sergey wrote:

I didn't want to confuse this shutdown use case. I meant that getting MSG_QUIT from the websocket client invokes that method

```
public boolean processBinaryMessage(byte[] message, int offset, int length) {

boolean handled = false;

if (length  1 && message[offset]  MSG_PAGE_LOADED) {
.............................................

else if (message[offset]  MSG_QUIT && length  1) {
// exit the client session
System.exit(0);
}

return handled;
}
```

#8 - 02/01/2017 03:19 PM - Greg Shah

It is a good point. I had forgotten we did that. For now I'm going to stop working on this idea.

#9 - 02/01/2017 03:20 PM - Greg Shah

Sergey wrote:

> Another thought is that we don't start watch dog timer here.
>
> // notify the remote client about application exit
> websock.quit(); should invoke WatchdogTimer with the server shutdown on receiving MSG_QUIT or by timeout.
>
> // since the application will exit shutdown the server
> shutdownServer();
>
> Thus I think the current shutdown logic is incorrect.
>
> Do you agree?

#10 - 02/01/2017 03:24 PM - Greg Shah

> Do you agree?

I think we do need to consider it, but I seem to recall that we deliberately don't use the watchdog in that case. Consider that when the embedded web server is being stopped, it is expected that there will no longer be anyone connecting. Notifying the JS client is a "best efforts" thing. There are many possible race conditions.

As you note: the JS client can trigger a System.exit() while we are trying to close the session with the FWD server. It will need some research and if we went forward with it, much testing.

#11 - 02/01/2017 03:27 PM - Greg Shah

On further thought, we cannot implement the server-side termination hooks as the only solution for managing the available port list. The issue here is that on CTRL-C, we "restart" the client without exiting. The web server stays running but the connection to the FWD server is terminated and then restarted (running == true in ClientCore.start()). We might be able to use it with a kind of FWD server-side watchdog timer. Since there are so many complications with the session management here, I'm postponing work on this.

#12 - 03/23/2017 01:34 PM - Greg Shah

- Related to Feature #2683: reverse proxy implementation added

#13 - 06/18/2017 01:13 PM - Sergey Ivanovskiy

Greg, please help to understand the remote case. There are two cases: 1) the web client is spawned locally and 2) the web client is spawned remotely. In the first case if we specify a range of ports, then these ports are allocated for this server host. But in the second case there are remote broker clients that are used to start the web client remotely. These brokers are assigned to a user's name and from the set of assigned brokers the server spawner selects the one that has the minimal workload on the system. It seems that the port range adds complexity to this logic. Are these brokers can be started for different hosts? If there are different hosts, then we should get at least their IPs addresses or their domain names. The reverse proxy should know an exact IP address or a domain name of the hosts on which the web client will be spawned, depending on the private network architecture. The reverse proxy configuration depends on the port range and on the hosts range. In 2683 we develop this mapping

```
1) https://proxy/gui <-> https://backend:7443/gui
2) https://proxy/7449/index.html <-> https://backend:7449/index.html
3) https://proxy/7449/* <-> https://backend:7449/$1
```

In the case of different backends we should classify backends somehow. We can use the same idea and add server to backend mapping

```
1) https://proxy/server1/gui <-> https://backend1:7443/gui
2) https://proxy/server1/7449/index.html <-> https://backend1:7449/index.html
3) https://proxy/server1/7449/* <-> https://backend1:7449/$1
```

#14 - 06/18/2017 01:26 PM - Sergey Ivanovskiy

Actually, 2683 and 3236 can be developed as a one task due to interdependency of these tasks on the reverse proxy configuration. At least 3236 should be developed first.

#15 - 06/19/2017 08:01 AM - Greg Shah

The highest priority is getting the local spawning working with port mapping. Please work on that first and we will try to get that into the trunk ASAP so that we can deliver a production reference VM that could be used in a cloud environment like AWS.

As a second phase (in a different task branch) we can work on the remote spawner case.

- Each remote spawner will need its own configuration including port (possibly a range, but also it could be 0 or a fixed port).
- We have a user account affinity issue. If we randomly pick a remote spawner (like today) we must know that the OS user account being launched exists on all of the remote spawners.
- The proxy configuration needs to be enhanced to handle the remote spawning case as you mentioned.

#16 - 06/21/2017 04:02 PM - Greg Shah

Code Review Task Branch 2683a Revision 11156

Overall the approach is OK. Is it working in your testing?

1. The WebClientRegistrator usage in ClientCore should not be used if the client is not a web client.

2. Change the name of WebClientRegistrator to WebClientRegistrar.

3. The WebClientsManager.getFreeLocalPort() needs to be atomic. Right now it is a race condition. Two requests to that method that are close enough can collide. When it allocates a port, it needs to add that port to the usedLocalPorts before returning so that the port can never be "found" twice.

4. What happens if the ChUI user executes CTRL-C or the GUI user executes ESC? Does the client properly reopen the session using the same port?

5. What happens of the MSG_QUIT comes back from the JS side before the driver.shutdown() has completed? This seems like a race condition where the client JVM may trigger System.exit() before the notification is sent to the FWD server.

6. The current FWD design for session listeners is pretty weird. This is not your fault. Since SessionManager.listen() can only accept a single listener, that listener can now only ever be used for this web client use case. That doesn't make much sense. And if web clients are not used, then we have the listener there when it is not needed. For now, it is OK, but the use of it highlights the problem we already had.

7. I guess that the hard coded port 7449 is just for testing purposes. The allocation needs to be bounded in a range and it must be configurable.

#17 - 06/22/2017 10:20 AM - Sergey Ivanovskiy

> 4. What happens if the ChUI user executes CTRL-C or the GUI user executes ESC? Does the client properly reopen the session using the same port?

If the JS web client is not closed or logged out, then the same port is used, otherwise new web client should be allocated if a user tries to logged in. But there is a problem with this requirement now. If a user's session is terminated, then this port can be given to new client that is about to be logged, but this port is still in used by the web client that gets CTRL_C or ESC. Is my view correct?

#18 - 06/22/2017 10:25 AM - Sergey Ivanovskiy

Does it mean that we should check if the session is terminated, but the web client is still running?

#19 - 06/22/2017 10:55 AM - Sergey Ivanovskiy

Greg Shah wrote:

> Code Review Task Branch 2683a Revision 11156
>
> Overall the approach is OK. Is it working in your testing?

Yes, it works now, but the current implementation is incorrect (has many issues to be resolved).

#20 - 06/22/2017 11:31 AM - Greg Shah

> If a user's session is terminated, then this port can be given to new client that is about to be logged, but this port is still in used by the web client that gets CTRL_C or ESC. Is my view correct?

Yes, that is correct. This is a problem.

#21 - 06/22/2017 01:22 PM - Sergey Ivanovskiy

Greg, if we have the case when the ports are given and all of them are used by the current clients, then new local users that are in the server's private network can not connect to the server too. Is it a correct requirement?

```
<node class="container" name="portsRange">
  <node class="string" name="namePrefix">
    <node-attribute name="value" value="client"/>
  </node>
  <node class="integer" name="from">
    <node-attribute name="value" value="7449"/>
  </node>
  <node class="integer" name="to">
    <node-attribute name="value" value="7549"/>
  </node>
</node>
```

#22 - 06/22/2017 02:42 PM - Greg Shah

> if we have the case when the ports are given and all of them are used by the current clients, then new local users that are in the server's private network can not connect to the server too.

Yes, correct.

#23 - 06/22/2017 03:54 PM - Sergey Ivanovskiy

Committed revision 11159 added portsRange node under webClient parent node and new business logic to release allocated resources. Working to fix the opened 1) and 5) issues.

#24 - 06/23/2017 09:36 AM - Sergey Ivanovskiy

Greg, does it make sense to create a configuration that provides a mapping of a user id to its web client port number? In that case each user has its own dedicated port number for the web client. And the web client allocation will be succeeded only if this user will be authenticated. Now the client process is spawned and only then the client is authenticated. Why the system don't authenticate the user before the spawn process is started?

#25 - 06/23/2017 09:38 AM - Sergey Ivanovskiy

> Why the system don't authenticate the user before the spawn process is started?

I understand, because the system is checking its OS credential.

#26 - 06/23/2017 09:41 AM - Sergey Ivanovskiy

Greg, don't matter my questions. It seems that one user must be able to create several web clients sessions with the server.

#27 - 06/23/2017 09:47 AM - Greg Shah

You've got it. It is always nice when someone answers their own questions. :)

#28 - 06/23/2017 11:10 AM - Sergey Ivanovskiy

I encountered the following major security risk as Denial of Service attacks at the application level, not at the tcp/ip level that simply don't give anybody to connect to the server via the web client. The fluid of several simulated login requests to log into the server with a fake user/password pair leads to denial of service since all ports will be concurrently allocated and then will be returned to the pool and then will become again allocated. At the moment the current schema makes this possible.

#29 - 06/23/2017 11:26 AM - Sergey Ivanovskiy

It seems that it is required to check the provided user/password pair and only then try to allocate the web client. Greg, what do you think? The user/password checking functionality can be similar to the one implemented for GWT Administration client.

#30 - 06/23/2017 11:46 AM - Sergey Ivanovskiy

This denial of service for non restricted ports is not possible, since the ports is not allocated before the user's OS credentials are verified and they are verified before the web client http server is started. But in the restricted ports case we allocate port number first and then try to spawn the web client.

#31 - 06/23/2017 01:58 PM - Sergey Ivanovskiy

It seems that Spawner interface should be changed to be able to allocate clients resources if it is required. Planning to test this way.

#32 - 06/23/2017 02:16 PM - Greg Shah

Sergey Ivanovskiy wrote:

> It seems that it is required to check the provided user/password pair and only then try to allocate the web client. Greg, what do you think? The user/password checking functionality can be similar to the one implemented for GWT Administration client.

Agreed. This means the client will have to "up call" to the FWD server to allocate and return the port.

#33 - 06/26/2017 06:22 AM - Sergey Ivanovskiy

I added new methods to allocate and to release system resources to the Spawn interface. It fixed only a part of that issue, the flow of external requests can still impact on the host that spawns processes. The system creates a one new OS process per an external request. Does it need to implement CAPTCHA for the login screen?

#34 - 06/26/2017 09:54 AM - Greg Shah

> The system creates a one new OS process per an external request. Does it need to implement CAPTCHA for the login screen?

For now, I don't think this is needed.

This problem should only occur with the virtual desktop mode. This is not expected to be used for production. It is useful for testing.

If we get to the point where a customer expects to use virtual desktop mode for production, we will revisit this issue. Please create a new task ("eliminate denial of service in virtual desktop mode") to describe this work.

I do think we should have a configuration option that will disable virtual desktop mode. This way, a production server would be more locked down. Please add this.

#35 - 06/26/2017 12:53 PM - Sergey Ivanovskiy

I don't understand how to detect if it is a POST request for the virtual desktop mode or the embedded client mode. We have these directory settings for the virtual desktop mode

```
<node class="container" name="webClient">
  <node class="boolean" name="enabled">
    <node-attribute name="value" value="TRUE"/>
  </node>
  <node class="boolean" name="embedded">
    <node-attribute name="value" value="FALSE"/>
  </node>
...............................................
</node>
```

and embedded option to enable the embedded mode.

#36 - 06/26/2017 12:54 PM - Greg Shah

If virtual desktop mode is disabled, can't we just refuse to respond to all requests for path /gui?

#37 - 06/26/2017 01:00 PM - Sergey Ivanovskiy

It seems that if the server refuses GET and POST requests for /gui, then the web client session can be established and the embedded client can't be started.

#38 - 06/26/2017 01:44 PM - Sergey Ivanovskiy

- Related to Bug #3304: Eliminate denial of service in virtual desktop mode added

#39 - 06/26/2017 01:56 PM - Greg Shah

Then in this mode can you just refuse non-embedded mode sessions in WebHandler?

#40 - 06/26/2017 02:18 PM - Sergey Ivanovskiy

Greg Shah wrote:

> Then in this mode can you just refuse non-embedded mode sessions in WebHandler?

It seems that there are no differences at this level between requests to start embedded or non-embedded clients. Embedded option is used for the JS client only, but the JS client is loaded only after the web client process is started. Please explain more thoroughly what are non-embedded mode sessions.

#41 - 06/26/2017 02:57 PM - Greg Shah

The embedded mode comes through the WebClientLauncher remote object which has the spawn() entry point. It does not connect through a web POST or GET request.

#42 - 06/26/2017 03:53 PM - Sergey Ivanovskiy

Thank you, I missed this entry point and didn't implement the port range for clients created by WebClientLauncher. Please help me with examples where this interface is used.

#43 - 06/26/2017 03:56 PM - Greg Shah

See hotel_gui/embedded/src/com/goldencode/testcases/embedded/CustomP2JClientApp.java. This is the embedded mode client that connects to the FWD server and launches embedded mode web clients.

#44 - 06/26/2017 04:05 PM - Sergey Ivanovskiy

Thank you, I see it now. Is it required to implement the port range for this type of clients too? Now they are skipped.

#45 - 06/26/2017 04:09 PM - Greg Shah

Yes, this is actually the most important type of client because most (if not all) customers will be using this approach as their production web GUI.

#46 - 06/26/2017 04:13 PM - Sergey Ivanovskiy

Ok, understand, I will add the existing functionality for this case too.

#47 - 06/26/2017 06:44 PM - Sergey Ivanovskiy

Committed revision 11164 added new options to the directory: forwardedHost, forwardedProto and virtalDesktopEnabled. The virtual desktop is disabled by default. These options forwardedHost and forwardedProto override corresponding HTTP headers: "X-Forwarded-Host" and "X-Forwarded-Proto". forwardedHost and forwardedProto are required if there are several proxies in the route to the server. This directory fragment lists all new additional options.

```
        <node class="container" name="webClient">
.............................................................
          <node class="container" name="portsRange">
            <node class="string" name="namePrefix">
              <node-attribute name="value" value="client"/>
            </node>
            <node class="integer" name="from">
              <node-attribute name="value" value="7449"/>
            </node>
            <node class="integer" name="to">
              <node-attribute name="value" value="7549"/>
            </node>
          </node>
          <node class="string" name="forwardedHost">
            <node-attribute name="value" value="www.goldencode.com"/>
          </node>
          <node class="string" name="forwardedProto">
            <node-attribute name="value" value="https"/>
          </node>
          <node class="boolean" name="virtualDesktopEnabled">
            <node-attribute name="value" value="FALSE"/>
          </node>
...................................
```

#48 - 06/27/2017 12:05 PM - Sergey Ivanovskiy

Greg,

> 5. What happens of the MSG_QUIT comes back from the JS side before the driver.shutdown() has completed? This seems like a race condition where the client JVM may trigger System.exit() before the notification is sent to the FWD server.

It seems there are 2 scenarios depending on how MSG_QUIT is initiated
1) MSG_QUIT is initiated by a user. It happens if a user reloads the current page. Now in this case a user is redirected to the login page.(?)
2) MSG_QUIT is send by the web client. In this case we have the same scenario the JS client gets MSG_QUIT and a user is redirected to the login page.

Please recall what functionality is expected to be if the current page is reloaded by a user. In the current implementation a user is redirected to the login page.
And if the web client is under the reverse proxy and the JS is out of this network, then the reload takes more than a second, on my environment is 2945 ms. But the redirect logic depends on timing now (it is supposed that it takes no more than one second) and in the worse case a user isn't redirected. Please look at this logs that displays the reload use case now. I increased this interval up to 4 seconds.

We can see that index.html , claro.css, dojo.js are loaded from the remote server. Now if a user is not redirected, then the web client is alive, the web socket chanel handles ping and pong messages, but the web client doesn't send the current application windows. It seems that this task is not implemented yet.

```
GET
https://127.0.0.1/server/client1/index.html [HTTP/1.1 200 OK 46ms]
Connection closed with the returned code 1001 p2j.screen.js:3227:7
GET
https://127.0.0.1/server/client1/dojo-toolkit/dijit/themes/claro/claro.css [HTTP/1.1 200 OK 38ms]
GET
https://127.0.0.1/server/client1/dojo-toolkit/dojo/dojo.js [HTTP/1.1 200 OK 77ms]
GET
https://127.0.0.1/server/client1/common/p2j.js [HTTP/1.1 304 Not Modified 28ms]
GET
https://127.0.0.1/server/client1/common/p2j.logger.js [HTTP/1.1 304 Not Modified 48ms]
GET
https://127.0.0.1/server/client1/common/p2j.sound.js [HTTP/1.1 304 Not Modified 73ms]
GET
https://127.0.0.1/server/client1/client/p2j.strokes.js [HTTP/1.1 304 Not Modified 73ms]
GET
https://127.0.0.1/server/client1/client/p2j.canvas_renderer.js [HTTP/1.1 304 Not Modified 69ms]
GET
https://127.0.0.1/server/client1/client/p2j.virtual_desktop.js [HTTP/1.1 304 Not Modified 89ms]
GET
https://127.0.0.1/server/client1/client/p2j.mouse.js [HTTP/1.1 304 Not Modified 108ms]
GET
https://127.0.0.1/server/client1/client/p2j.screen.js [HTTP/1.1 304 Not Modified 117ms]
GET
https://127.0.0.1/server/client1/common/p2j.socket.js [HTTP/1.1 304 Not Modified 143ms]
GET
https://127.0.0.1/server/client1/common/p2j.keymap.js [HTTP/1.1 304 Not Modified 125ms]
GET
https://127.0.0.1/server/client1/common/p2j.keyboard.js [HTTP/1.1 304 Not Modified 116ms]
GET
https://127.0.0.1/server/client1/client/p2j.clipboard_helpers.js [HTTP/1.1 304 Not Modified 122ms]
GET
https://127.0.0.1/server/client1/common/p2j.clipboard.js [HTTP/1.1 304 Not Modified 145ms]
GET
https://127.0.0.1/server/client1/common/p2j.fonts.js [HTTP/1.1 304 Not Modified 148ms]
GET
https://127.0.0.1/server/client1/common/p2j.remote.js [HTTP/1.1 304 Not Modified 144ms]
GET
https://127.0.0.1/server/client1/ajax [HTTP/1.1 101 Switching Protocols 24ms]
Elapsed: 2945 ms  p2j.socket.js:3177:10
Connection closed with the returned code 1005 p2j.screen.js:3227:7
https://127.0.0.1/gui
GET
https://127.0.0.1/gui
```

#49 - 06/27/2017 01:31 PM - Hynek Cihlar

Obviously the time condition in isRequiredToRedirect() is there to distinguish the case when the page loads right after F5 and the case when the reconnection timer fires.

Sergey, as you point out the time condition is unreliable as the time the page loads varies depending on client or server state, network conditions, etc. As such it should be removed and I believe the following steps are the way to go:

1. isRequiredToRedirect() should be removed.
2. In createWebSocket() the onopen handler: when the message to the client web server MSG_PAGE_LOADED fails (timeout, unrecoverable netowrk error, session does not exist (if this can happen), etc.) then the redirect to login page should be performed.
3. MSG_QUIT should be send in onpagehide.

#50 - 06/27/2017 01:58 PM - Sergey Ivanovskiy

Thank you, it simplifies the current logic. It seems that it should work as we expected.

#51 - 06/27/2017 02:47 PM - Sergey Ivanovskiy

No, it has an issue with the requirement to reconnect to the web client if the network is temporarily down and then is on again. We can't distinguish the case if it is error due to MSG_PAGE_LOADED fails or connectivity timer fails trying to restore the connection.

#52 - 06/27/2017 05:12 PM - Sergey Ivanovskiy

Hynek Cihlar wrote:

> Obviously the time condition in isRequiredToRedirect() is there to distinguish the case when the page loads right after F5 and the case when the reconnection timer fires.

It seems that it is here to distinguish reload use case from reopen use case, because it checks if the marker object "exitTheApplication" is present, I think that the watchdog timeout of the server can help to distinguish the reopen case from the reload case, because if the JS client will be disconnected from the web client within the time that exceeds the watchdog timeout, then the old web client will be closed. Committed revision 11165 fixed misspelled code that sets incorrectly "watchdogTimeout" value.

#53 - 06/28/2017 01:21 AM - Sergey Ivanovskiy

Committed revision 11167 prepared this branch 2683a for the review.

#54 - 06/28/2017 06:19 AM - Greg Shah

Code Review Task Branch 2683a Revision 11167

I have checked in some changes to fix code formatting issues. Overall, I like the result.

1. In ClientCore, please cache the WebClientRegistrar instance obtained on line 263 so that the freeWebClientResources() on line 385 is called on an existing instance instead of obtaining a new one.

2. In WebClientsManager.allocateClient(), is it safe to always hard code the webClientHost to localhost?

3. Please add javadoc for ProcessClientSpawner.releaseClient(), SpawnerImpl.releaseClient(), WebClientSpawner.webClientsManager, WebClientSpawner.requestParameters, WebClientSpawner.TemporaryClientTask.proxyServerParameters.

#55 - 06/28/2017 07:28 AM - Constantin Asofiei

Sergey, the issues I have are these:

- have you tested if the FWD client terminates abruptly? In this case, the server-side needs to cleanup the resources on its own. This code in ClientCore is correct:

```
            if (!running)
            {
               if (referrer != null)
               {
                  WebClientRegistrar registrar =
                           (WebClientRegistrar) RemoteObject.obtainNetworkInstance(
                                   WebClientRegistrar.class, session);
                  registrar.freeWebClientResources(uuid);
               }
            }
            session.terminate();
```

  as the session is terminated only after the web client resources are freed, but WebClientsManager.terminate should do this, too.

#56 - 06/28/2017 09:03 AM - Sergey Ivanovskiy

If the web client is failed, then its session is closed, and it seems that all listeners should be notified and terminate() invoked, should not they?

#57 - 06/28/2017 09:07 AM - Constantin Asofiei

Sergey Ivanovskiy wrote:

> If the web client is failed, then its session is closed, and it seems that all listeners should be notified and terminate() invoked, should not they?

Yes, if the FWD session is terminated, then WebClientsManager.terminate is executed. I missed the part where isStillAlive calls freeClient (which releases the resources), but this is done after the timeout (60 seconds), correct? It seems a little to late... why not do this immediately, as the FWD session can't be re-established (as the FWD client is already dead/disconnected/etc and in no way can recover the same context)? Otherwise, the port will be kept as used for the duration of the timeout.

#58 - 06/28/2017 09:17 AM - Sergey Ivanovskiy

Constantin Asofiei wrote:

> Sergey Ivanovskiy wrote:
>
> > If the web client is failed, then its session is closed, and it seems that all listeners should be notified and terminate() invoked, should not they?
>
> Yes, if the FWD session is terminated, then WebClientsManager.terminate is executed. I missed the part where isStillAlive calls freeClient (which releases the resources), but this is done after the timeout (60 seconds), correct? It seems a little to late... why not do this immediately, as the FWD session can't be re-established (as the FWD client is already dead/disconnected/etc and in no way can recover the same context)? Otherwise, the port will be kept as used for the duration of the timeout.

I did this because we can't free a given port until its system resources are released. But we didn't know if this session is terminated and the client is not alive or is about to reconnect with new session. For users it means that if the resource pool is exhausted, then a user can get message all resources are used, but after refresh it can get a free connection. It seems that it is a correct behaviour.

#59 - 06/28/2017 09:31 AM - Constantin Asofiei

Sergey Ivanovskiy wrote:

> I did this because we can't free a given port until its system resources are released.

You are referring the client releasing its associated port, correct?

> But we didn't know if this session is terminated and the client is not alive or is about to reconnect with new session.

When SessionListener.terminate is called, the FWD client is no longer available/seen by the FWD server - the client can be dead, network connection to the FWD server dropped, etc. For the FWD server, this FWD client can no longer be reconnected, and all its resources need to be cleaned up. But the server side doesn't know if the FWD client process is still alive or not... it sees only that its session was terminated.

> For users it means that if the resource pool is exhausted, then a user can get message all resources are used, but after refresh it can get a free connection. It seems that it is a correct behaviour.

What I'm trying to confirm is that the FWD server always frees the resources, for a FWD client which has disconnected from the server. When isStillAlive is called from terminate, at least it should continue to attempt a request until a failure occurs: currently, if the client is still alive (but not connected to the FWD server), and isStillAlive is called from terminate, there is a chance for the request to succeed and the resources will not be released.

So, I'm thinking that isStillAlive should have a boolean parameter which 'forces' the resource cleanup, when called from terminate: it will execute in a loop (with some Thread.sleep code) until the HTTP request fails and resources are cleaned up.

#60 - 06/28/2017 09:33 AM - Hynek Cihlar

Code Review Task Branch 2683a.

In addition to Greg's and Constantin's finds.

- In WebClientSpawner
  - if (!"/".equals(webClientConfig.getWebRoot())) what if webClientConfig.getWebRoot() returns an empty/blank string or null? should the config value be overriden in this case? also it would be safer to trim the compared string values or to use URI class for the comparison.
  - it would be safer to do spawner.releaseClient(uuid); in a finally block. this would cover the expected case of uri == null as well as any unexpected error conditions
  - using URI(String scheme, String host, String path, String fragment) instead of URI.create(forwardedProto + "://" + forwardedHost + forwardedPath + "/") would improve robustness as the leading slash in forwardedPath or the ending slash in forwardedHost would not have to be expected.
- portsResticted => portsRestricted
- WebClientsManager: // TODO Auto-generated catch block
- Below the port should be given out when isStillAlive() returns false. Also this may be a performance bottleneck once the number of clients increases. I think it would help if the network check would be performed only when no free port is found in usedLocalPorts.

```
      while (freePort <= to)
+      {
+         WebClientConfig config = usedLocalPorts.get(freePort);
+         if (config == null)
+         {
+            return freePort;
+         }
+         else
+         {
+            isStillAlive(config);
+         }
+
+         freePort++;
+      }
```

- sendForbiddenIfVirtualDesktopDisabled is missing @return javadoc
- a hardcoded https: String forwardedProto = "https"; //base.getHeader(HttpHeader.X_FORWARDED_PROTO.asString());
- DojoToolkitHandler should be changed to take into account a non-emoty webRoot (and probably all the other resource handlers (js, mp3, wav, etc) too?)

#61 - 06/28/2017 09:45 AM - Sergey Ivanovskiy

Constantin Asofiei wrote:

> What I'm trying to confirm is that the FWD server always frees the resources, for a FWD client which has disconnected from the server. When isStillAlive is called from terminate, at least it should continue to attempt a request until a failure occurs: currently, if the client is still alive (but not connected to the FWD server), and isStillAlive is called from terminate, there is a chance for the request to succeed and the resources will not be released.

All resources will be released on new requests to log in via the web client.

> So, I'm thinking that isStillAlive should have a boolean parameter which 'forces' the resource cleanup, when called from terminate: it will execute in a loop (with some Thread.sleep code) until the HTTP request fails and resources are cleaned up.

I missed this point because I think that according to ClientCore the session can be terminated, but the web client as a process can be alive and then new session can be initiated with the same parameters for the embedded web server. What is the purpose of running flag? If the web client as a instance of ClientCore can't create new session, then we can implement it. Please explain more thoroughly why the web client can't reestablish new session with the server.

#62 - 06/28/2017 10:05 AM - Sergey Ivanovskiy

Hynek Cihlar wrote:

> Code Review Task Branch 2683a.
>
> In addition to Greg's and Constantin's finds.
>
> - In WebClientSpawner
> - if (!"/".equals(webClientConfig.getWebRoot())) what if webClientConfig.getWebRoot() returns an empty/blank string or null? should the config value be overriden in this case? also it would be safer to trim the compared string values or to use URI class for the comparison.
> - it would be safer to do spawner.releaseClient(uuid); in a finally block. this would cover the expected case of uri == null as well as any unexpected error conditions
> - using URI(String scheme, String host, String path, String fragment) instead of URI.create(forwardedProto + "://" + forwardedHost + forwardedPath + "/") would improve robustness as the leading slash in forwardedPath or the ending slash in forwardedHost would not have to be expected.
> - portsResticted => portsRestricted
> - WebClientsManager: // TODO Auto-generated catch block
> - Below the port should be given out when isStillAlive() returns false. Also this may be a performance bottleneck once the number of clients increases. I think it would help if the network check would be performed only when no free port is found in usedLocalPorts.
> [...]
>
> - sendForbiddenIfVirtualDesktopDisabled is missing @return javadoc
> - a hardcoded https: String forwardedProto = "https"; //base.getHeader(HttpHeader.X_FORWARDED_PROTO.asString());

Planning to fix them.

- DojoToolkitHandler should be changed to take into account a non-emoty webRoot (and probably all the other resource handlers (js, mp3, wav, etc) too?)

It seems that DojoToolkitHandler should not be changed, because it is not a web virtual directory for the embedded server, but a user sees it like a virtual directory. webRoot is a mapping for the reverse proxy.

#63 - 06/28/2017 10:09 AM - Sergey Ivanovskiy

It seems that DojoToolkitHandler should not be changed, because it is not a web virtual directory for the embedded server, but a user sees it like a virtual directory. webRoot is a mapping for the reverse proxy. For users in the same network it is a root "/", but for others it is a none empty string like /server/client1/

#64 - 06/28/2017 10:30 AM - Hynek Cihlar

Sergey Ivanovskiy wrote:

> It seems that DojoToolkitHandler should not be changed, because it is not a web virtual directory for the embedded server, but a user sees it like a virtual directory. webRoot is a mapping for the reverse proxy. For users in the same network it is a root "/", but for others it is a none empty string like /server/client1/

Good, you would find this one right away anyway.

#65 - 06/28/2017 11:11 AM - Sergey Ivanovskiy

Hynek Cihlar wrote:

> Code Review Task Branch 2683a.
>
> - In WebClientSpawner
> - if (!"/".equals(webClientConfig.getWebRoot())) what if webClientConfig.getWebRoot() returns an empty/blank string or null? should the config value be overriden in this case? also it would be safer to trim the compared string values or to use URI class for the comparison.
> - it would be safer to do spawner.releaseClient(uuid); in a finally block. this would cover the expected case of uri == null as well as any unexpected error conditions
> - using URI(String scheme, String host, String path, String fragment) instead of URI.create(forwardedProto + "://" + forwardedHost + forwardedPath + "/") would improve robustness as the leading slash in forwardedPath or the ending slash in forwardedHost would not have to be expected.

At the moment I don't have another design construction, now webClientConfig.getWebRoot() is always not null and can be "/" or the special mapping.

#66 - 06/28/2017 11:14 AM - Sergey Ivanovskiy

Hynek Cihlar wrote:

> Code Review Task Branch 2683a.
>
> - In WebClientSpawner
> - Below the port should be given out when isStillAlive() returns false. Also this may be a performance bottleneck once the number of clients increases. I think it would help if the network check would be performed only when no free port is found in usedLocalPorts.

```
while (freePort <= to)
+       {
+          WebClientConfig config = usedLocalPorts.get(freePort);
+          if (config == null)
+          {
+             return freePort;
+          }
+          else
+          {
+             isStillAlive(config);
+          }
+
+          freePort++;
+       }
```

Please explain what is the bottleneck?

#67 - 06/28/2017 11:58 AM - Hynek Cihlar

Sergey Ivanovskiy wrote:

> Hynek Cihlar wrote:
>
> > Code Review Task Branch 2683a.
> >
> > - In WebClientSpawner
> > - Below the port should be given out when isStillAlive() returns false. Also this may be a performance bottleneck once the number of clients increases. I think it would help if the network check would be performed only when no free port is found in usedLocalPorts.

> [...]
>
> Please explain what is the bottleneck?

The bottleneck is the isStillAlive() which does a network IO to find out whether a client is still alive.

#68 - 06/28/2017 12:04 PM - Sergey Ivanovskiy

I agree if you mean that https involves heavy calculations. Committed rev. 11169 fixed all comments that I could do.

#69 - 06/28/2017 12:10 PM - Hynek Cihlar

Sergey Ivanovskiy wrote:

> I agree if you mean that https involves heavy calculations.

Even without https, one check may cost you several milliseconds (and depending on the network conditions up to tens of milliseconds).

#70 - 06/28/2017 12:13 PM - Greg Shah

If I understand correctly, the check only occurs if we can't allocate a port. In the common case (there is a free port available), this doesn't gt executed.

Perhaps it is OK in this scenario?

#71 - 06/28/2017 12:32 PM - Hynek Cihlar

Greg Shah wrote:

> If I understand correctly, the check only occurs if we can't allocate a port.

Unless I am reading the code wrong, the method goes through all active/nonactive clients and performs the net IO, until it hits a free port. Also the method is called on client session start and on client session termination.

#72 - 06/28/2017 12:53 PM - Constantin Asofiei

Sergey Ivanovskiy wrote:

> So, I'm thinking that isStillAlive should have a boolean parameter which 'forces' the resource cleanup, when called from terminate: it will execute in a loop (with some Thread.sleep code) until the HTTP request fails and resources are cleaned up.

> I missed this point because I think that according to ClientCore the session can be terminated, but the web client as a process can be alive and then new session can be initiated with the same parameters for the embedded web server. What is the purpose of running flag?

When false, the running flag determines if the FWD client is terminating.

> If the web client as a instance of ClientCore can't create new session, then we can implement it. Please explain more thoroughly why the web client can't reestablish new session with the server.

When true, running flag indicates the client must restart, but the FWD session will not be re-used - a new one will be created. Also, in this case, registrar.freeWebClientResources(uuid); will not be called, but a new uuid will be created... see line 266, which executes registrar.registerWebClientSession(uuid, session.getNodeAddress()); on each iteration of this running loop. I assume that the web driver will still re-use the assigned port (during spawn), but it will not be able to clean it, as the UUID will no longer match.

Beside the above, what I have in mind is a possible race condition; consider these execution points, concurrently:

1. FWD client is executing just before ClientCore.start:366:

```
if (!running) // client is here
{
   driver.shutdown();
}
```

2. assume for some reason the client-server FWD connection is broken; now, FWD server will receive a terminate event and call isStillAlive, but it can see that connectTest is still responsive, thus it doesn't clean.
3. the FWD client continues executing, but it can't call the server side to cleanup.

The two cases above are both possible scenarios where the port can be 'leaked'. Eventually, it will be determined as 'not in use' by getFreeLocalPort, so at least we have this.

#73 - 06/28/2017 12:55 PM - Constantin Asofiei

Constantin Asofiei wrote:

> When true, running flag indicates the client must restart, but the FWD session will not be re-used - a new one will be created. Also, in this case, registrar.freeWebClientResources(uuid); will not be called, but a new uuid will be created... see line 266, which executes registrar.registerWebClientSession(uuid, session.getNodeAddress()); on each iteration of this running loop. I assume that the web driver will still re-use the assigned port (during spawn), but it will not be able to clean it, as the UUID will no longer match.

Sorry, the UUID is not recreated, so this case looks OK.

#74 - 06/28/2017 01:46 PM - Greg Shah

Hynek Cihlar wrote:

> Greg Shah wrote:
>
> > If I understand correctly, the check only occurs if we can't allocate a port.
>
> Unless I am reading the code wrong, the method goes through all active/nonactive clients and performs the net IO, until it hits a free port. Also the method is called on client session start and on client session termination.

Hmm. You're right.

Sergey: it would be better to find a free port without doing the isStillAlive() check and only use this if all ports are unavailable.

#75 - 06/28/2017 02:24 PM - Sergey Ivanovskiy

Constantin Asofiei wrote:

> Beside the above, what I have in mind is a possible race condition; consider these execution points, concurrently:
>
> 1. FWD client is executing just before ClientCore.start:366:
> [...]
> 2. assume for some reason the client-server FWD connection is broken; now, FWD server will receive a terminate event and call isStillAlive, but it can see that connectTest is still responsive, thus it doesn't clean.
> 3. the FWD client continues executing, but it can't call the server side to cleanup.

> The two cases above are both possible scenarios where the port can be 'leaked'. Eventually, it will be determined as 'not in use' by getFreeLocalPort, so at least we have this.

Constantin, does it mean that this problem of leaking resources described above exists in the trunc version? The web client is running, using the port, but the server and the client communication loses forever.

#76 - 06/28/2017 02:27 PM - Sergey Ivanovskiy

Greg Shah wrote:

> Hynek Cihlar wrote:
>
> > Greg Shah wrote:
> >
> > > If I understand correctly, the check only occurs if we can't allocate a port.
> >
> > Unless I am reading the code wrong, the method goes through all active/nonactive clients and performs the net IO, until it hits a free port. Also the method is called on client session start and on client session termination.
>
> Hmm. You're right.
>
> Sergey: it would be better to find a free port without doing the isStillAlive() check and only use this if all ports are unavailable.

Ok, this method isStillAlive() works asynchronously, it only loads the server for a while.

#77 - 06/28/2017 02:37 PM - Hynek Cihlar

Sergey Ivanovskiy wrote:

> Greg Shah wrote:

Hynek Cihlar wrote:

Greg Shah wrote:

If I understand correctly, the check only occurs if we can't allocate a port.

Unless I am reading the code wrong, the method goes through all active/nonactive clients and performs the net IO, until it hits a free port. Also the method is called on client session start and on client session termination.

Hmm. You're right.

Sergey: it would be better to find a free port without doing the isStillAlive() check and only use this if all ports are unavailable.

Ok, this method isStillAlive() works asynchronously, it only loads the server for a while.

Still, the extra load can be easily avoided by scanning the usedLocalPorts map and only do the IO when no free port is found.

Since the method is async you need a join point that would be executed when the net IO finishes. An example: the port range is 2, the usedLocalPorts is full, first entry points to a dead client, second to the live. In this case the algorithm should return the port of the dead client, but it doesn't.

#78 - 06/28/2017 02:45 PM - Sergey Ivanovskiy

Hynek Cihlar wrote:

Still, the extra load can be easily avoided by scanning the usedLocalPorts map and only do the IO when no free port is found.

Since the method is async you need a join point that would be executed when the net IO finishes. An example: the port range is 2, the usedLocalPorts is full, first entry points to a dead client, second to the live. In this case the algorithm should return the port of the dead client, but it doesn't.

You proposed another algorithm, but you reviewed the different method, it works as expected. Please approve that I should implement your proposals.

#79 - 06/28/2017 02:53 PM - Hynek Cihlar

Sergey Ivanovskiy wrote:

> Hynek Cihlar wrote:
>
> > Still, the extra load can be easily avoided by scanning the usedLocalPorts map and only do the IO when no free port is found.
> >
> > Since the method is async you need a join point that would be executed when the net IO finishes. An example: the port range is 2, the usedLocalPorts is full, first entry points to a dead client, second to the live. In this case the algorithm should return the port of the dead client, but it doesn't.
>
> You proposed another algorithm, but you reviewed the different method, it works as expected. Please approve that I should implement your proposals.

What do you mean by different method? Below is the diff I see between the branch and trunk. Assuming the diff shows the most recent version the method will return -1 when the configured port range is 2 and usedLocalPorts is filled with one dead client and one alive client, which is clearly not correct.

```
+   private int getFreeLocalPort()
+   {
+      if (!portsRestricted)
+      {
+         return 0;
+      }
+
+      int freePort = from;
+
+      while (freePort <= to)
+      {
+         WebClientConfig config = usedLocalPorts.get(freePort);
+         if (config == null)
+         {
+            return freePort;
+         }
+         else
+         {
+            isStillAlive(config);
+         }
+
+         freePort++;
+      }
+
+      return -1;
+   }
```

#80 - 06/28/2017 02:57 PM - Sergey Ivanovskiy

Yes, it is expected behaviour, but it is only for the case if the almost all resources are used (#range -1), correct? Please consider the case when all resources are exhausted, then new requests can also load the system and users will wait for negative responses.

#81 - 06/28/2017 02:57 PM - Constantin Asofiei

Sergey Ivanovskiy wrote:

> Constantin, does it mean that this problem of leaking resources described above exists in the trunc version? The web client is running, using the port, but the server and the client communication loses forever.

In trunk, we don't have explicit web port management: is either set static (and only one client available at a time) or is set dynamic (and the OS takes care of assigning the port). So, the limit would be reached if no free socket is available at the OS level.

In our case, where we want to manage the ports explicitly (and the range I assume will be limited), we need to make sure we clean up properly...

One other thought about the case described in notes 72 (and 73): when the client is re-initiating, the driver will remain configured with the same web port - this is OK. But, considering the previous FWD session will be terminated (and another one will be created in ClientCore), the server-side can't know that the port is still in use or the client is terminating. What about this:

1. WebClientConfig has an additional counter (synchronized access), which gets incremented each time registerWebClientSession is called.
2. in WebClientsManager.terminate, pass the value of this counter to isStillAlive.
3. in isStillAlive, when called from terminate, there is a loop which:

- if the HTTP request returns as failed, then we know the port is free and just free the resources
- in the for loop, if the counter for the WebClientConfig instance is not the same as the counter received as argument (thus gets modified), then we know the config is being re-used so we can exit the loop safely
- otherwise, if the counter doesn't modify in a certain number of iterations and in all these iterations the HTTP request is passed, log something that the client is 'runaway'. We can't actually free the port, as is in use at the OS level...
- also, as the newRequest call is async, you will need some synchronization to make it sync - use some latch and block on it until the request is processed, and the result is saved.

#82 - 06/28/2017 03:04 PM - Hynek Cihlar

Sergey Ivanovskiy wrote:

> Yes, it is expected behaviour, but it is only for the case if the almost all resources are used (#range -1), correct?

Are you saying that it is an expected bahavior to deny a user access when a port is available to be used but blocked by a dead client entry? If so then

please ignore my comments.

> Please consider the case when all resources are exhausted, then new requests can also load the system and users will wait for negative responses.

Which is the case with the current implementation, too - when all ports are taken and alive, the method will check all the active clients for every new client access request.

#83 - 06/28/2017 03:08 PM - Sergey Ivanovskiy

Constantin Asofiei wrote:

> Sergey Ivanovskiy wrote:
>
> > Constantin, does it mean that this problem of leaking resources described above exists in the trunc version? The web client is running, using the port, but the server and the client communication loses forever.
>
> In trunk, we don't have explicit web port management: is either set static (and only one client available at a time) or is set dynamic (and the OS takes care of assigning the port). So, the limit would be reached if no free socket is available at the OS level.

You considered the case of leaking resources in the general case, the port management that is under development and review doesn't invluence this case. If the system is already has possibility of leaking resources, then how to find ways to manage the web clients?

> In our case, where we want to manage the ports explicitly (and the range I assume will be limited), we need to make sure we clean up properly...

I understand what is required.

#84 - 06/28/2017 03:18 PM - Sergey Ivanovskiy

Hynek Cihlar wrote:

> Sergey Ivanovskiy wrote:
>
> > Yes, it is expected behaviour, but it is only for the case if the almost all resources are used (#range -1), correct?
>
> Are you saying that it is an expected bahavior to deny a user access when a port is available to be used but blocked by a dead client entry? If so then please ignore my comments.
>
> > Please consider the case when all resources are exhausted, then new requests can also load the system and users will wait for negative responses.
>
> Which is the case with the current implementation, too - when all ports are taken and alive, the method will check all the active clients for every new client access request.

I will try to fix it.

#85 - 06/28/2017 03:23 PM - Hynek Cihlar

Sergey Ivanovskiy wrote:

> Which is the case with the current implementation, too - when all ports are taken and alive, the method will check all the active clients for every new client access request.
>
> I will try to fix it.

Sergey, this should be an easy fix. Just scan the usedLocalPorts map first for free ports, when you find one, return it. When you find no free port, issue those asynchronous requests and wait for the first that indicates a dead client. But I would suggest to first work in those inputs from Constantin. Hope this helps.

#86 - 06/28/2017 03:30 PM - Sergey Ivanovskiy

Ok, thank you, we can try blocking API for this case.

#87 - 06/28/2017 03:31 PM - Sergey Ivanovskiy

Constantin, Hynek as I understand correctly there is a case when the web client lives forever and can't reconnect to the server again, and the server loses its connection with its web client forever. For this case this web client port can't be used and we have a leak. It doesn't depend on how we manage web clients, correct?

#88 - 06/28/2017 03:32 PM - Constantin Asofiei

Sergey Ivanovskiy wrote:

> Constantin, Hynek as I understand correctly there is a case when the web client lives forever and can't reconnect to the server again, and the server loses its connection with its web client forever. For this case this web client port can't be used and we have a leak. It doesn't depend on how we manage web clients, correct?

Yes, in this case we can't free it as is still in use at the OS level. Something else went wrong and manual intervention (i.e. kill the process) is required.

#89 - 06/28/2017 03:57 PM - Constantin Asofiei

Sergey Ivanovskiy wrote:

> You considered the case of leaking resources in the general case, the port management that is under development and review doesn't invluence this case. If the system is already has possibility of leaking resources, then how to find ways to manage the web clients?

With your changes for virtual desktop mode, a web port is 'in use' only if spawn (and OS-level authentication) passes, correct? So we can't have a leak, just (as you mentioned) a denial of service possibility (and you mentioned this already in a previous comment).

With embedded mode, (for example, how embedded Hotel GUI is built), the FWD iframe will be 'launched' only when authentication data is submitted (i.e. user clicks 'Login'); and if authentication fails, then the FWD client is terminated (and port freed). In this case too, only a DoS is possible.

At this time, and with your changes, I don't think a leak is possible where the FWD web client gets started and web port/connection established without passing the authentication phase.

#90 - 06/28/2017 06:07 PM - Sergey Ivanovskiy

Committed revision 11170 fixed getFreeLocalPort, now exploring "session counter" from the note 81.

#91 - 06/28/2017 08:38 PM - Sergey Ivanovskiy

Please review committed revision 11171 that added "session counter" from the note 81.

#92 - 06/29/2017 04:55 AM - Greg Shah

Code Review Task Branch 2683a Revision 11171

1. The WebClientManager.try{Acquire|Release}Port() methods have a common sleeping loop that probably can be abstracted into a separate method.

2. I think that #3236-85 is not yet addressed. What is your plan there?

3. What notes still need to be addressed?

4. I have checked in some changes to fix code formatting problems. Please be more careful with the code.

5. WebClientConfig.sessionsCounter and its getter/setter methods need javadoc.

6. In WebClientManager please fix the imports to use *.

#93 - 06/29/2017 04:57 AM - Greg Shah

Constantin: do the changes suitably address your concerns?

Also: perhaps we need to add some comments to make sure that your "leaking resources" concerns are clear to future readers of the code. I don't want people changing the code later and not realizing that such changes could cause this condition.

#94 - 06/29/2017 05:47 AM - Hynek Cihlar

Greg Shah wrote:

> 2. I think that #3236-85 is not yet addressed. What is your plan there?

I think it is addressed in the current top branch revision. The code now checks the map of used client ports and only performs net IO when no free one is found.

Sergey, just one thing about the implementation. You should not ignore the InterruptedException in tryAcquirePort(), instead do something like this:

```
        try
        {
           synchronizer.await(timeout, TimeUnit.MILLISECONDS);
        }
        catch (InterruptedException e)
        {
           Thread.currentThread().interrupt(); // turn the flag back on so that the other call stack frames h
ave a chance to handle the case
           throw new RuntimeException(e);
        }

        return availablePort.get();
```

And you no longer need the outer while loop.

I noticed you use this pattern on other place(s), too. Be sure to fix those, too.

#95 - 06/29/2017 06:01 AM - Constantin Asofiei

Greg Shah wrote:

> Constantin: do the changes suitably address your concerns?

Yes.

> Also: perhaps we need to add some comments to make sure that your "leaking resources" concerns are clear to future readers of the code. I don't want people changing the code later and not realizing that such changes could cause this condition.

Sergey, please do the following:

- in the javadoc for tryReleasePort add comments stating that this will try to free the port for a terminated session, but it might not free it if is still in use (client was restarted) or if the client is 'runaway' and the port is open at the OS level, without re-using it.
- in tryReleasePort, if the loop ends with the session counter same as expectedCount, log a message stating that the port couldn't be freed because is still open at OS level and the client is not re-using it.

#96 - 06/29/2017 08:10 AM - Sergey Ivanovskiy

Committed revision 11173 fixed all found issues, but now I am investigating the case when all two ports are used and the third user gets OutOfResources, and then one of the running client is terminated, so the resource should be free, but the code doesn't work as expected. Planning to fix it today.

#97 - 06/29/2017 08:11 AM - Greg Shah

Sergey: Please finish item 6 in #3236-92 and handle #3236-95. These should be trivial.

I do need you utility program for creating the reverse proxy configuration. But that tool can be added safely after testing, assuming it doesn't get used at conversion/runtime.

As soon as you have the recently found bug (#3236-96) fixed, is everything else done which is needed for testing?

#98 - 06/29/2017 08:22 AM - Sergey Ivanovskiy

Greg Shah wrote:

> Sergey: Please finish item 6 in #3236-92 and handle #3236-95. These should be trivial.

Committed revision 11173 fixed these trivial issues.

> I do need you utility program for creating the reverse proxy configuration. But that tool can be added safely after testing, assuming it doesn't get used at conversion/runtime.

It is not ready yet but it is not difficult, as an example I am using SSLCertGenUtil

> As soon as you have the recently found bug (#3236-96) fixed, is everything else done which is needed for testing?

Apache server should be installed and these two modules: mod_proxy and mod_proxy_wstunnel should be enabled. Then it is required to setup Apache configuration to use https and to use the reverse proxy configuration from https://proj.goldencode.com/issues/2683#note-54.

#99 - 06/29/2017 08:29 AM - Sergey Ivanovskiy

I am using the default /etc/apache2/sites-available/default-ssl.conf.

```
<IfModule mod_ssl.c>
<VirtualHost _default_:443>
....................
# the reverse proxy settings
#place here configuration of the reverse proxy
</VirtualHost>
</IfModule>
```

Please add configuration from https://proj.goldencode.com/issues/2683#note-54 at the end of the VirtualHost directive and enable mod_proxy and mod_proxy_wstunnel modules:

```
sudo a2enmod proxy
sudo a2enmod proxy_http
sudo a2enmod proxy_ajp
sudo a2enmod rewrite
sudo a2enmod deflate
sudo a2enmod headers
sudo a2enmod proxy_balancer
sudo a2enmod proxy_connect
sudo a2enmod proxy_html
sudo a2enmod proxy_html
sudo a2enmod proxy_wstunnel
```

#100 - 06/29/2017 08:34 AM - Greg Shah

Committed revision 11173 fixed these trivial issues.

I don't see any new javadoc for Constantin's request in note 95:

> in the javadoc for tryReleasePort add comments stating that this will try to free the port for a terminated session, but it might not free it if is still in use (client was restarted) or if the client is 'runaway' and the port is open at the OS level, without re-using it.
> in tryReleasePort, if the loop ends with the session counter same as expectedCount, log a message stating that the port couldn't be freed because is still open at OS level and the client is not re-using it.

is everything else done which is needed for testing?

I do need the full details on setup/config for the reverse proxy. But in this question, I was asking if you can get the 2683a into regression testing today (ChUI stuff on devsrv01 and also manual GUI testing). I need this to be in the trunk ASAP.

#101 - 06/29/2017 08:38 AM - Sergey Ivanovskiy

Greg Shah wrote:

> Committed revision 11173 fixed these trivial issues.
>
> I don't see any new javadoc for Constantin's request in note 95:
>
> > in the javadoc for tryReleasePort add comments stating that this will try to free the port for a terminated session, but it might not free it if is still in use (client was restarted) or if the client is 'runaway' and the port is open at the OS level, without re-using it.
> > in tryReleasePort, if the loop ends with the session counter same as expectedCount, log a message stating that the port couldn't be freed because is still open at OS level and the client is not re-using it.
>
> is everything else done which is needed for testing?
>
> I do need the full details on setup/config for the reverse proxy. But in this question, I was asking if you can get the 2683a into regression testing today (ChUI stuff on devsrv01 and also manual GUI testing). I need this to be in the trunk ASAP.

They are present. Please look at these changes

```
    /**
-    * Try to release port acquired by the target web client.
+    * Try to release port acquired by the target web client on triggered terminated session events,
+    * but it might not free it if is still in use (client was restarted) or if the client is
+    * 'runaway' and the port is open at the OS level, without re-using it.
     *
     * @param    config
```

and

```
+               if (synchronizer.await(TIMEOUT, TimeUnit.MILLISECONDS))
+               {
+                  return;
+               }
+            }
+         }
+         catch(InterruptedException e)
+         {
+            Thread.currentThread().interrupt();
+            throw new RuntimeException(e);
+         }
+         if (config.getSessionsCounter().compareAndSet(expectedCount, expectedCount))
+         {
+            System.err.println("The port '" + config.getPort() + "' couldn't be freed because" +
+                     "is still open at OS level and the client uuid='" + config.getUuid() +
+                     "' is not re-using it.");
          }
       });
```

#102 - 06/29/2017 08:40 AM - Greg Shah

Sorry, I didn't look carefully enough.

What about the readiness for regression/GUI testing?

#103 - 06/29/2017 08:45 AM - Constantin Asofiei

Sergey Ivanovskiy wrote:

> They are present. Please look at these changes
> [...]

Please use a logger and not System.err.

#104 - 06/29/2017 08:59 AM - Sergey Ivanovskiy

The manual GUI and regression testing are in process now, but the current code has some unknown issue and it should be fixed. The proposed schema doesn't work as expected and now I am investigating the reason.

The reverse proxy settings are in note 99. The Apache was setup 4 months ago and I can miss something related to setup but the configuration is clean.

#105 - 06/29/2017 10:27 AM - Sergey Ivanovskiy

The considered issue is that if all ports are used by the clients and new one user is trying to get a client, then it acquires WebClientsManager and releases it only after TIMEOUT period.

#106 - 06/29/2017 10:39 AM - Hynek Cihlar

Sergey Ivanovskiy wrote:

> The considered issue is that if all ports are used by the clients and new one user is trying to get a client, then it acquires WebClientsManager and releases it only after TIMEOUT period.

Do you count down or interrupt the synchronizer in tryAcquirePort() after all ports are checked? If I remember correctly the synchronizer is interrupted only when the client check fails.

#107 - 06/29/2017 10:48 AM - Sergey Ivanovskiy

Yes, but it is required to wait until requests timeout is passed. Committed revision 11175 fixed this strict synchronization on WebClientsManager and added logger. This version is ready ffor regression testing. Now planning to do manual GUI tests.

#108 - 06/29/2017 10:57 AM - Hynek Cihlar

Sergey Ivanovskiy wrote:

> Yes, but it is required to wait until requests timeout is passed. Committed revision 11175 fixed this strict synchronization on WebClientsManager and added logger. This version is ready ffor regression testing. Now planning to do manual GUI tests.

In tryAcquirePort() you call synchronizer.countDown() only when client check fails. Thus when all the clients are alive and all ports are taken, the method will sit and wait for the duration of TIMEOUT milliseconds before it returns -1. Which seems like the behavior you mention in note 105. So again, you should do synchronizer.countDown() when all the clients have been tested!

#109 - 06/29/2017 10:58 AM - Sergey Ivanovskiy

Hynek Cihlar wrote:

> Sergey Ivanovskiy wrote:
>
> > Yes, but it is required to wait until requests timeout is passed. Committed revision 11175 fixed this strict synchronization on WebClientsManager and added logger. This version is ready ffor regression testing. Now planning to do manual GUI tests.
>
> In tryAcquirePort() you call synchronizer.countDown() only when client check fails. Thus when all the clients are alive and all ports are taken, the method will sit and wait for the duration of TIMEOUT milliseconds before it returns -1. Which seems like the behavior you mention in note 105. So again, you should do synchronizer.countDown() when all the clients have been tested!

You are correct, but it is not a issue. I will do it later.

#110 - 06/29/2017 11:01 AM - Hynek Cihlar

Sergey Ivanovskiy wrote:

> Hynek Cihlar wrote:
>
> > Sergey Ivanovskiy wrote:
> >
> > > Yes, but it is required to wait until requests timeout is passed. Committed revision 11175 fixed this strict synchronization on WebClientsManager and added logger. This version is ready ffor regression testing. Now planning to do manual GUI tests.

> > In tryAcquirePort() you call synchronizer.countDown() only when client check fails. Thus when all the clients are alive and all ports are taken, the method will sit and wait for the duration of TIMEOUT milliseconds before it returns -1. Which seems like the behavior you mention in note 105. So again, you should do synchronizer.countDown() when all the clients have been tested!
>
> You are correct, but it is not a issue.

I thought it was an issue according to your note 105: "The considered issue is that if all ports are used by the clients and new one user is trying to get a client, then it acquires WebClientsManager and releases it only after TIMEOUT period."

#111 - 06/29/2017 11:02 AM - Sergey Ivanovskiy

You are right, it is an issue.

#112 - 06/29/2017 02:29 PM - Sergey Ivanovskiy

I have problems with hotel_gui, and investigating them now.

#113 - 06/29/2017 03:53 PM - Sergey Ivanovskiy

Sergey Ivanovskiy wrote:

> I have problems with hotel_gui, and investigating them now.

It seems that the spawned process is failed silently for hotel_gui and a user is redirected to the web client and all resources are loaded, but the web socket is failed.
But with the P2J trunc it starts the web client successfully. so I missed something that should be easy. How to log the spawned process? The part with returned uri works properly, but then the JS client can't connect its web client and logging processes displays that it was failed silently.

#114 - 06/29/2017 03:57 PM - Hynek Cihlar

Sergey Ivanovskiy wrote:

> Sergey Ivanovskiy wrote:
>
> > I have problems with hotel_gui, and investigating them now.

> It seems that the spawned process is failed silently for hotel_gui and a user is redirected to the web client and all resources are loaded, but the web socket is failed.
> But with the P2J trunc it starts the web client successfully. so I missed something that should be easy. How to log the spawned process? The part with returned uri works properly, but then the JS client can't connect its web client and logging processes displays that it was failed silently.

Check the java client log in deploy/client.

#115 - 06/29/2017 04:00 PM - Sergey Ivanovskiy

Thank you, I see here only logs for P2J trunc's web client, so the embedded server was not started at all and the spawned process is failed before that moment.

#116 - 06/29/2017 04:05 PM - Hynek Cihlar

Sergey Ivanovskiy wrote:

> Thank you, I see here only logs for P2J trunc's web client, so the embedded server was not started at all and the spawned process is failed before that moment.

Did you do gradlew all -Dpost.build=yes -Dspawn.install.folder=/opt/spawner on your branch?

#117 - 06/29/2017 04:16 PM - Sergey Ivanovskiy

Thank you :), I rerun with 2683a and found that client logs are rewritten.

```
java.lang.InterruptedException: sleep interrupted
    at java.lang.Thread.sleep(Native Method)
    at com.goldencode.p2j.ui.client.driver.web.WatchdogTimer.run(WatchdogTimer.java:128)
java.lang.RuntimeException: Unresolvable remote export public abstract void com.goldencode.p2j.main.WebClientR
egistrar.registerWebClientSession(java.lang.String,int).
    at com.goldencode.p2j.net.RemoteObject$RemoteAccess.obtainRoutingKey(RemoteObject.java:1580)
    at com.goldencode.p2j.net.RemoteObject$RemoteAccess.invokeCore(RemoteObject.java:1464)
    at com.goldencode.p2j.net.InvocationStub.invoke(InvocationStub.java:145)
    at com.sun.proxy.$Proxy8.registerWebClientSession(Unknown Source)
    at com.goldencode.p2j.main.ClientCore.start(ClientCore.java:266)
    at com.goldencode.p2j.main.ClientCore.start(ClientCore.java:158)
    at com.goldencode.p2j.main.ClientDriver.start(ClientDriver.java:250)
    at com.goldencode.p2j.main.CommonDriver.process(CommonDriver.java:444)
    at com.goldencode.p2j.main.ClientDriver.process(ClientDriver.java:144)
    at com.goldencode.p2j.main.ClientDriver.main(ClientDriver.java:313)
Caused by: java.lang.RuntimeException: No export or no access to com.goldencode.p2j.main.WebClientRegistrar:pu
blic abstract void com.goldencode.p2j.main.WebClientRegistrar.registerWebClientSession(java.lang.String,int)
    at com.goldencode.p2j.net.HighLevelObject.getKey(HighLevelObject.java:169)
    at com.goldencode.p2j.net.RemoteObject$RemoteAccess.obtainRoutingKey(RemoteObject.java:1552)
    ... 9 more
java.lang.InterruptedException
```

```
    at java.lang.Object.wait(Native Method)
    at java.lang.Object.wait(Object.java:502)
    at com.goldencode.p2j.ui.client.driver.web.PushMessagesWorker.waitForMessages(PushMessagesWorker.java:187)
    at com.goldencode.p2j.ui.client.driver.web.PushMessagesWorker.run(PushMessagesWorker.java:144)
java.lang.InterruptedException
    at java.lang.Object.wait(Native Method)
    at java.lang.Object.wait(Object.java:502)
    at com.goldencode.p2j.ui.client.driver.web.WebTaskWorker.run(WebTaskWorker.java:110)
```

#118 - 06/29/2017 04:30 PM - Greg Shah

This is an ACL issue. In /security/acl/net/ you need to add a section for WebClientRegistrar to allow access to the exported API. You can pattern it after the section on WebClientLauncher.

Make sure you check in your minimum needed changes to the directory template (for both Hotel ChUI and Hotel GUI) so that we can all use this new feature.

#119 - 06/29/2017 04:51 PM - Sergey Ivanovskiy

It works now, thank you, I just copied WebClientLauncher section and replaced it with WebClientRegistrar. Are these settings correctly added?

```
        <node class="container" name="002275">
          <node class="resource" name="resource-instance">
            <node-attribute name="reftype" value="TRUE"/>
            <node-attribute name="reference" value="com.goldencode.p2j.main.WebClientLauncher"/>
          </node>
          <node class="netRights" name="rights">
            <node-attribute name="permissions" value="'0101'B"/>
          </node>
          <node class="strings" name="subjects">
            <node-attribute name="values" value="all_others"/>
          </node>
        </node>
        <node class="container" name="002276">
          <node class="resource" name="resource-instance">
            <node-attribute name="reftype" value="TRUE"/>
            <node-attribute name="reference" value="com.goldencode.p2j.main.WebClientRegistrar"/>
          </node>
          <node class="netRights" name="rights">
            <node-attribute name="permissions" value="'0101'B"/>
          </node>
          <node class="strings" name="subjects">
            <node-attribute name="values" value="all_others"/>
          </node>
        </node>
```

#120 - 06/29/2017 05:15 PM - Greg Shah

Yes.

#121 - 06/30/2017 01:21 AM - Sergey Ivanovskiy

I have the following issue with the embedded client, probably due to certificate issue.
If the port settings in the standard hotel_gui directory.xml for the web client has a port=7449 for which the web client certificate was accepted by the browser, then
the embedded hotel works properly, but if the port number is zero, then the embedded client is not loaded by iframe. The web client is started successfully, there are no errors in its log, but the server log has this error

```
Jun 30, 2017 8:09:30 AM Protocol.Reader.run()
WARNING: {Reader} failure in reading loop
javax.net.ssl.SSLProtocolException: Data received in non-data state: 6
    at sun.security.ssl.SSLSocketImpl.readRecord(SSLSocketImpl.java:1109)
    at sun.security.ssl.SSLSocketImpl.readDataRecord(SSLSocketImpl.java:930)
    at sun.security.ssl.AppInputStream.read(AppInputStream.java:105)
    at sun.security.ssl.AppInputStream.read(AppInputStream.java:71)
    at java.io.ObjectInputStream$PeekInputStream.peek(ObjectInputStream.java:2598)
    at java.io.ObjectInputStream$BlockDataInputStream.peek(ObjectInputStream.java:2905)
    at java.io.ObjectInputStream$BlockDataInputStream.peekByte(ObjectInputStream.java:2915)
    at java.io.ObjectInputStream.readObject0(ObjectInputStream.java:1502)
    at java.io.ObjectInputStream.readObject(ObjectInputStream.java:422)
    at com.goldencode.p2j.net.Protocol$Reader.run(Protocol.java:416)
    at java.lang.Thread.run(Thread.java:748)
```

So this standard configuration

```
    <node class="container" name="server">
      <node class="container" name="default">
        <node class="container" name="webClient">
          <node class="boolean" name="embedded">
            <node-attribute name="value" value="FALSE"/>
          </node>
          <node class="boolean" name="enabled">
            <node-attribute name="value" value="TRUE"/>
          </node>
          <node class="string" name="host">
            <node-attribute name="value" value="localhost"/>
          </node>
          <node class="integer" name="port">
            <node-attribute name="value" value="0"/>
          </node>
          <node class="integer" name="maxBinaryMessage">
            <node-attribute name="value" value="32894"/>
          </node>
          <node class="integer" name="maxIdleTime">
            <node-attribute name="value" value="90000"/>
          </node>
          <node class="integer" name="watchdogTimeout">
            <node-attribute name="value" value="120000"/>
          </node>
```

doesnt work but this one

```
    <node class="container" name="server">
      <node class="container" name="default">
        <node class="container" name="webClient">
          <node class="boolean" name="embedded">
            <node-attribute name="value" value="FALSE"/>
          </node>
          <node class="boolean" name="enabled">
            <node-attribute name="value" value="TRUE"/>
          </node>
          <node class="string" name="host">
            <node-attribute name="value" value="localhost"/>
          </node>
          <node class="integer" name="port">
            <node-attribute name="value" value="7449"/>
```

```
        </node>
        <node class="integer" name="maxBinaryMessage">
          <node-attribute name="value" value="32894"/>
        </node>
        <node class="integer" name="maxIdleTime">
          <node-attribute name="value" value="90000"/>
        </node>
        <node class="integer" name="watchdogTimeout">
          <node-attribute name="value" value="120000"/>
        </node>
```

works with the embedded client. Constantin, can you recall how this problem was solved if it is not a new one?

#122 - 06/30/2017 01:30 AM - Sergey Ivanovskiy

I removed

```
        <node class="boolean" name="virtualDesktopEnabled">
          <node-attribute name="value" value="TRUE"/>
        </node>
```

from these tested configurations in order to use the standard hotel_gui directory settings, but the case is the same port=7449 works for the embedded client, but port=0 doesn't .

#123 - 06/30/2017 04:07 AM - Constantin Asofiei

Sergey Ivanovskiy wrote:

> I removed
> [...]
> from these tested configurations in order to use the standard hotel_gui directory settings, but the case is the same port=7449 works for the embedded client, but port=0 doesn't .

Have you imported the root CA into the browser? Is the https connection green in the left-side part of the URL?

#124 - 06/30/2017 06:00 AM - Sergey Ivanovskiy

I did not import CA into the browser, it seems that in the Firefox the icon looks like an orange triangle with an exclamation sign, but the Chrome logs insecure connection response from ehotel.js:529 GET https://localhost:41337/index.html?token=021c63c8592056a8866f4bf7ebd925d6 net::ERR_INSECURE_RESPONSE. Thank you, it helps to detect this problem with the certificate that appears again.

#125 - 06/30/2017 06:35 AM - Sergey Ivanovskiy

Looking into these

```
webSocketTimeout     integer     -1     no     Web socket timeout (in milliseconds). This is a timeout after w
hich the web socket is closed .
watchdogTimeout     integer     -1     no     Watchdog timeout (in milliseconds). After the web socket is clos
ed, the watchdog will wait at least that certain amount of time and, if the web socket was connected again, it
 will decide to terminate the FWD client, as the remote user is no longer connected to it. A value of -1 will
mean the remote FWD client will be terminated as soon as the web socket is closed.
```

I found that if watchdogTimeout == -1 is not set, then it means that the watchdog timer can be interrupted immediately, since once started it fails to sleep for 60000 milliseconds.
But if webSocketTimeout is not set and -1, then I think that the underlined web socket API has (unknown to me) the default session timeout that is taken into account if there are no messages during this period. Who have had a knowledge about this default setting?

#126 - 06/30/2017 06:51 AM - Sergey Ivanovskiy

Web sockets use keep-alive HTTP connections, but does it mean that they leave forever or there exists a default timeout that is set by a particular web server?

#127 - 06/30/2017 07:45 AM - Greg Shah

I don't know the answer to this. How critical is it to our delivery today?

Can you provide a status of things?

If Hynek or Constantin need to help they will. We have to get this fixed, tested and into the trunk as early today as possible.

#128 - 06/30/2017 07:55 AM - Sergey Ivanovskiy

Greg Shah wrote:

> I don't know the answer to this. How critical is it to our delivery today?
>
> Can you provide a status of things?

I tested with hotel_gui it seems that there are no regressions in this code and there are no new issues and now I am doing ClientsToPortsGenerator utility to create map.clients-to-backends file that is referenced by the Apache configuration https://proj.goldencode.com/issues/2683#note-54.

> If Hynek or Constantin need to help they will. We have to get this fixed, tested and into the trunk as early today as possible.

I didn't run ChUI regression tests.

#129 - 06/30/2017 07:55 AM - Constantin Asofiei

Sergey Ivanovskiy wrote:

> I didn't run ChUI regression tests.

I'll start these now.

#130 - 06/30/2017 07:56 AM - Greg Shah

Please run the ChUI regression testing immediately. The CTRL-C tests will be quite important.

#131 - 06/30/2017 07:57 AM - Greg Shah

Constantin: Thanks!

#132 - 06/30/2017 08:00 AM - Sergey Ivanovskiy

Thanks, I need this help to run 2683a at least once manually with hotel_gui using the embedded and web clients taking into account that it is required to change directory settings according 118, 119 and 120.

#133 - 06/30/2017 08:02 AM - Hynek Cihlar

Sergey Ivanovskiy wrote:

> Thanks, I need this help to run 2683a at least once manually with hotel_gui using the embedded and web clients.

I can help with this one. Is this for regression test? In other words is the reverse proxy setup needed?

#134 - 06/30/2017 08:06 AM - Sergey Ivanovskiy

Hynek Cihlar wrote:

> Sergey Ivanovskiy wrote:
>
> > Thanks, I need this help to run 2683a at least once manually with hotel_gui using the embedded and web clients.
>
> I can help with this one. Is this for regression test? In other words is the reverse proxy setup needed?

Thank you, for regression only, please take into account that it is required to add this new settings to the directory for hotel GUI (notes 118, 119, 120)

```
        <node class="container" name="002275">
          <node class="resource" name="resource-instance">
            <node-attribute name="reftype" value="TRUE"/>
            <node-attribute name="reference" value="com.goldencode.p2j.main.WebClientLauncher"/>
          </node>
          <node class="netRights" name="rights">
            <node-attribute name="permissions" value="'0101'B"/>
          </node>
          <node class="strings" name="subjects">
            <node-attribute name="values" value="all_others"/>
          </node>
        </node>
        <node class="container" name="002276">
          <node class="resource" name="resource-instance">
            <node-attribute name="reftype" value="TRUE"/>
            <node-attribute name="reference" value="com.goldencode.p2j.main.WebClientRegistrar"/>
          </node>
          <node class="netRights" name="rights">
            <node-attribute name="permissions" value="'0101'B"/>
          </node>
          <node class="strings" name="subjects">
            <node-attribute name="values" value="all_others"/>
          </node>
        </node>
```

#135 - 06/30/2017 08:09 AM - Greg Shah

Code Review Task branch 2683a Revision 11177

I like the changes.

I have only one question: is it possible to hit any of the WebClientsManager APIs during authentication processing? We have found this problem in other web-sockets code:

```
 * IMPORTANT: Do not use logging anywhere inside this class. When an authentication plugin
 * is running during session establishment with the P2J server using logging no messages are
 * written into the log and the application becomes deadlocked.
```

You can search on this text to find the files affected. Please make consider if this can affect WebClientsManager (since it now uses logging).

Otherwise, if testing shows everything working properly, then we are good to go for a merge to trunk.

#136 - 06/30/2017 09:58 AM - Hynek Cihlar

Sergey Ivanovskiy wrote:

> Hynek Cihlar wrote:
>
> > Sergey Ivanovskiy wrote:
> >
> > > Thanks, I need this help to run 2683a at least once manually with hotel_gui using the embedded and web clients.
> >
> > I can help with this one. Is this for regression test? In other words is the reverse proxy setup needed?
>
> Thank you, for regression only, please take into account that it is required to add this new settings to the directory for hotel GUI (notes 118, 119, 120)

The hotel_gui app runs OK in both the virtualdesktop and embedded modes. Are there any particular tests you would like to run?

#137 - 06/30/2017 10:04 AM - Sergey Ivanovskiy

Thank you, I don't know that it seems that if the ports range is restricted, then it can be a weak place. I tested when ports are exhausted and clients are refreshed, closed or logged out, but it seems working.

#138 - 06/30/2017 10:05 AM - Constantin Asofiei

Sergey Ivanovskiy wrote:

> Thank you, I don't know that it seems that if the ports range is restricted, then it can be a weak place. I tested when ports are exhausted and clients are refreshed, closed or logged out, but it seems working.

Does this mean with restricted ports and the reverse proxy, Hotel GUI works OK?

#139 - 06/30/2017 10:06 AM - Greg Shah

Make sure that we are testing in 2 modes:

1. Through a reverse proxy.
2. Direct access.

#140 - 06/30/2017 10:12 AM - Sergey Ivanovskiy

Constantin Asofiei wrote:

> Sergey Ivanovskiy wrote:
>
> > Thank you, I don't know that it seems that if the ports range is restricted, then it can be a weak place. I tested when ports are exhausted and clients are refreshed, closed or logged out, but it seems working.
>
> Does this mean with restricted ports and the reverse proxy, Hotel GUI works OK?

It seems ok, tested them via Apache proxy server and run clients that are under this reverse proxy and clients that have direct access simultaneously.

#141 - 06/30/2017 10:15 AM - Sergey Ivanovskiy

But it will be great if the Apache environment manual tests will be approved and tested on the different host machine.

#142 - 06/30/2017 10:21 AM - Sergey Ivanovskiy

Constantin, could you help with this. I will try to add node to the directory using directory API, is it a correct way?

```
ds.addNode("/server/default/webClient/portsRange", "container", null);
```

Exception in thread "main" java.lang.RuntimeException: addNode() failed for the /server/default/webClient/portsRange node
at com.goldencode.p2j.main.ClientsToPortsGenerator.addNode(ClientsToPortsGenerator.java:346)
at com.goldencode.p2j.main.ClientsToPortsGenerator.generate(ClientsToPortsGenerator.java:163)
at com.goldencode.p2j.main.ClientsToPortsGenerator.main(ClientsToPortsGenerator.java:468)

#143 - 06/30/2017 10:26 AM - Constantin Asofiei

Sergey Ivanovskiy wrote:

> Constantin, could you help with this. I will try to add node to the directory using directory API, is it a correct way?
> [...]

What do you need this for? You can debug and step into DirectoryService.addNode and see where it fails. Do you have this ds.addNode call in a ds.openBatch(nodeBasePath); and ds.closeBatch(true) bracket? (nodeBasePath is /server/default/webClient).

#144 - 06/30/2017 10:28 AM - Sergey Ivanovskiy

Thank you, I read docs and thought that openBatch brackets are used only for security nodes.

#145 - 06/30/2017 10:49 AM - Sergey Ivanovskiy

The clients to ports mapping tool works properly, only it creates case insensitive node name like 'portsrange' instead of 'portsRange'. It is due to IdUtils.normalize(nodeId)

#146 - 06/30/2017 10:51 AM - Sergey Ivanovskiy

Can I commit com.goldencode.p2j.main.ClientsToPortsGenerator into 2683a?

#147 - 06/30/2017 10:58 AM - Greg Shah

Sergey Ivanovskiy wrote:

> Can I commit com.goldencode.p2j.main.ClientsToPortsGenerator into 2683a?

Yes.

#148 - 06/30/2017 11:24 AM - Sergey Ivanovskiy

Committed revision 11178, sorry for delay, during tests found that nodes are not updated if it is necessarily to satisfy (to >= from) and fixed it by placing them in the batch brackets to lock a node for updates.

#149 - 06/30/2017 11:33 AM - Sergey Ivanovskiy

Committed revision 11179 fixed java docs.

#150 - 06/30/2017 12:00 PM - Sergey Ivanovskiy

- File map.clients-to-backends added

- File default-ssl.conf added

Added my default ssl configuration with a mapping file.

```
sbi@am:~/goldencode/usrbin$ ll /etc/apache2
total 124
drwxr-xr-x   9 root root  4096 Jun 29 00:14 ./
drwxr-xr-x 171 root root 12288 Jun 30 08:17 ../
-rw-r--r--   1 root root  7052 Jan 16 00:51 apache2.conf
drwxr-xr-x   2 root root  4096 Jun 27 07:32 conf-available/
drwxr-xr-x   2 root root  4096 Jan 12 17:21 conf-enabled/
-rw-r--r--   1 root root  1782 Mar 19  2016 envvars
-rw-r--r--   1 root root   379 Jan 16 09:18 gcd-site-map.txt
-rw-r--r--   1 root root   193 Jan 16 16:46 jta-subsite-map.txt
-rw-r--r--   1 root root   181 Jan 16 09:18 jvm-subsite-map.txt
-rw-r--r--   1 root root   208 Jan 16 16:48 kto-subsite-map.txt
-rw-r--r--   1 root root 31063 Mar 19  2016 magic
-rw-r--r--   1 root root    47 Jun 29 00:14 map.clients-to-backends
drwxr-xr-x   2 root root 12288 Jun 27 07:32 mods-available/
drwxr-xr-x   2 root root  4096 Mar  2 01:24 mods-enabled/
-rw-r--r--   1 root root   208 Jan 16 16:49 nto-subsite-map.txt
-rw-r--r--   1 root root   320 Mar 19  2016 ports.conf
drwxr-xr-x   2 root root  4096 Jun 30 18:55 sites-available/
drwxr-xr-x   2 root root  4096 Jun 21 00:26 sites-enabled/
drwxr-xr-x   4 root root  4096 Mar  1 23:39 ssl/
```

and

```
sbi@am:~/goldencode/usrbin$ ll /etc/apache2/sites-available/
total 36
drwxr-xr-x 2 root root 4096 Jun 30 18:55 ./
drwxr-xr-x 9 root root 4096 Jun 29 00:14 ../
-rw-r--r-- 1 root root 1332 Mar 19  2016 000-default.conf
-rw-r----- 1 root root 1640 Apr  5 12:15 am.conf
-rw-r----- 1 root root 1648 Jan 19 19:43 amd64.conf
-rw-r--r-- 1 root root 7054 Jun 30 18:55 default-ssl.conf
-rw-r----- 1 root root 1473 Apr  5 11:15 fwd.conf
-rw-r----- 1 root root 3974 Apr  5 12:37 gcd.conf
```

#151 - 06/30/2017 12:18 PM - Sergey Ivanovskiy

I just imported hotel server der certificate into Firefox and got embedded clients working properly.

#152 - 06/30/2017 12:22 PM - Sergey Ivanovskiy

Greg, if you need to proxy this url https://localhost:8443/, then it should be added separetely, like for this url https://localhost:7443

```
ProxyPass /gui  https://localhost:7443/gui
ProxyPassReverse /gui https://localhost:7443/gui

ProxyPass /embedded  https://localhost:8443
ProxyPassReverse /embedded https://localhost:8443
```

#153 - 06/30/2017 12:57 PM - Greg Shah

Yes, I will need to proxy both 8443 and 7443. 8443 is the really important one.

Is there anything else needed?

#154 - 06/30/2017 12:57 PM - Sergey Ivanovskiy

Sergey Ivanovskiy wrote:

> Greg, if you need to proxy this url https://localhost:8443/, then it should be added separetely, like for this url https://localhost:7443
> [...]

There is a issue with dojo loader if we will be proxy https://localhost:8443/, because it works by adding <script> nodes to the JS client DOM and it creates problems for correct mappings.

#155 - 06/30/2017 12:58 PM - Sergey Ivanovskiy

I guess that we need to fix dojo configuration settings.

```
var dojoConfig =
{
   baseUrl : "./",
   tlmSiblingOfDojo : false,
   packages : [
                { name: "dojo", location: "dojo-release-1.10.0/dojo" }
               ]
};
```

#156 - 06/30/2017 01:00 PM - Greg Shah

All access is through port 443 right? So that is the only port exposed via the firewall?

What are the starting URLs for embedded mode and for virtual desktop mode?

#157 - 06/30/2017 01:03 PM - Sergey Ivanovskiy

Greg Shah wrote:

> All access is through port 443 right? So that is the only port exposed via the firewall?
>
> What are the starting URLs for embedded mode and for virtual desktop mode?

For virtual desktop mode it is https://forwardedhost/gui
For embedded client it should be https://forwardedhost/embedded but I have not fixed the correct mapping for this 3 levels proxy<->web-server<->web-client

#158 - 06/30/2017 01:10 PM - Sergey Ivanovskiy

Sergey Ivanovskiy wrote:

> Greg Shah wrote:
>
> > All access is through port 443 right? So that is the only port exposed via the firewall?
> >
> > What are the starting URLs for embedded mode and for virtual desktop mode?
>
> For embedded client it should be https://forwardedhost/embedded but I have not fixed the correct mapping for this 3 levels proxy<->web-server<->web-client

At first it needs to change dojo config that was set by index.html to

```
    <script>
       var dojoConfig =
       {
          baseUrl : "./embedded",
          tlmSiblingOfDojo : false,
          packages : [
                      { name: "dojo", location: "dojo-release-1.10.0/dojo" }
                     ]
       };
    </script>
```

in order to load scripts and the second issue with this incorrect POST request

```
 /**
   * Log into the client driver in order to load the embedded, converted ABL content.
   *
```

```
    * @param   {string} u
    *          User name.
    * @param   {string} p
    *          Password.
    */
   me.login = function(u, p)
   {
      var ifr = document.getElementById("embeddedP2J");

      fwd_u = u;
      fwd_p = p;

      if (ifr.src.startsWith("https://"))
      {
         var msgId = getNextMessageId();

         var args =
         {
            "messageId": msgId.toString(),
            "username" : fwd_u,
            "password" : fwd_p
         };

         p2j.embedded.invokeProcedure(null, "doLogin", args, null);
      }
      else
      {
         var xmlHttp = new XMLHttpRequest();
         xmlHttp.onreadystatechange = function()
         {
            if (xmlHttp.readyState == 4 && xmlHttp.status == 200)
            {
               var fwd_url = xmlHttp.responseText;

               if (fwd_url.startsWith("null"))
               {
                  showError({ "error": "Could not connect to the FWD server!"} );
               }
               else
               {
                  ifr.src = fwd_url;

                  $("#p2jLoginContainer").hide();

                  // the main view is transparent at this point; however, we need to realize it here to
                  // work around a firefox bug (https://bugzilla.mozilla.org/show_bug.cgi?id=941146),
                  // which we would otherwise hit due to FWD's use of an unrealized canvas to calculate
                  // some font metrics
                  $("#mainView").show();

                  // mostly hide background image with a nearly opaque overlay
                  $("#bgFader").show().css("opacity", 0.87);

                  p2j.embedded.init("embeddedP2J", "embeddedP2JOverlay", "embedded-overlay");
                  p2j.embedded.setMessageCallback(ehotel.messageCallback);
                  p2j.embedded.setPageLoadedCallback(ehotel.pageLoaded);
                  p2j.embedded.setPageClosedCallback(ehotel.pageClosed);
                  p2j.embedded.setSocketDisconnectedCallback(ehotel.socketDisconnected);
                  p2j.embedded.setSocketReconnectedCallback(ehotel.socketReconnected);
               }
            }
         }

         xmlHttp.open("GET", "/launch", true);
         xmlHttp.send(null);
      }
   };
```

#159 - 06/30/2017 01:16 PM - Constantin Asofiei

Sergey Ivanovskiy wrote:

> At first it needs to change dojo config that was set by index.html to

Are these changes compatible with non-proxy (normal) mode for embedded Hotel GUI?

> in order to load scripts and the second issue with this incorrect POST request
> [...]

The /launch in xmlHttp.open("GET", "/launch", true); is used to start the FWD embedded in the iframe (see EmbeddedWebAppDriver$EmbeddedPageHandler.handle).

#160 - 06/30/2017 01:20 PM - Sergey Ivanovskiy

Yes, I understand, but there are no ways, we need to change index.html and ehotel.js. It should take into account window.location to access /launch. Otherwise we can't create working example even using Apache html rewrite module since it can't handle such coded url mappings.

#161 - 06/30/2017 01:26 PM - Constantin Asofiei

Sergey Ivanovskiy wrote:

> Yes, I understand, but there are no ways, we need to change index.html and ehotel.js. It should take into account window.location to access /launch. Otherwise we can't create working example even using Apache html rewrite module since it can't handle such coded url mappings.

OK, please fix the /launch, too.

#162 - 06/30/2017 01:39 PM - Greg Shah

Any code changes to embedded mode will need to be applied to the customer POC project too.

#163 - 06/30/2017 01:52 PM - Sergey Ivanovskiy

- File required_changes.txt added

Greg, Constantin these changes work with the Apache configuration provided by note 152, I did not use HTML rewrite module to fix urls inside html, because it is impossible for our case. Please review this fix, it works :)

#164 - 06/30/2017 02:51 PM - Sergey Ivanovskiy

Greg Shah wrote:

> Any code changes to embedded mode will need to be applied to the customer POC project too.

Greg, does it need to do this work today?

#165 - 06/30/2017 03:17 PM - Greg Shah

Sergey Ivanovskiy wrote:

> Greg Shah wrote:
>
> > Any code changes to embedded mode will need to be applied to the customer POC project too.
>
> Greg, does it need to do this work today?

Yes, it is that customer POC code that we are having to deliver today.

#166 - 06/30/2017 03:31 PM - Greg Shah

Is there any remaining known issue with 2683a?

I understand that the POC embedded code needs changes. Is there anything else?

Besides the ChUI regression testing, is there any other testing that is yet to be done?

Finally: do these scenarios fully work with a single FWD server, reverse proxy and embedded server configuration?

```
1. reverse proxy to GUI embedded mode

user <--https_port_443-- apache_reverse_proxy --https_port_8443--> embedded_mode_web_server <--https_port_7443
--> fwd_server
                                          |
      |
                                          --https_wss_port_XYZ--> embedded_mode_web_client <-----spawn--------
-------

2. reverse proxy to GUI virtual desktop mode

user <--https_port_443-- apache_reverse_proxy --https_port_7443--> fwd_server ---------------------------
                                          |                                                             |
                                          --https_wss_port_XYZ--> embedded_mode_web_client <-----spawn---
```

```
3. direct to GUI embedded mode

user <--https_port_8443--> embedded_mode_web_server <--https_port_7443--> fwd_server----------------------
  |                                                                                                      |
  -------------------------------------------https_wss_port_XYZ--> embedded_mode_web_client <-----spawn---

4. direct to GUI virtual desktop

user <--https_port_7443--> fwd_server ----------------------------------------------------------------------
  |                                                                                                      |
  -------------------------------------------https_wss_port_XYZ--> embedded_mode_web_client <-----spawn---
```

In cases 1 and 2, port XYZ can be any of these:

- dynamically allocated from a port range
- a single fixed port (single-user only)

In cases 3 and 4, port XYZ can be:

- dynamically allocated from any open port on the system (port == 0 in the directory)
- dynamically allocated from a port range
- a single fixed port (single-user only)

#167 - 06/30/2017 03:41 PM - Sergey Ivanovskiy

Greg Shah wrote:

> Is there any remaining known issue with 2683a?
>
> I understand that the POC embedded code needs changes. Is there anything else?
>
> Besides the ChUI regression testing, is there any other testing that is yet to be done?
>
> Finally: do these scenarios fully work with a single FWD server, reverse proxy and embedded server configuration?
>
> [...]
>
> In cases 1 and 2, port XYZ can be any of these:
>
> - dynamically allocated from a port range
> - a single fixed port (single-user only)

It seems that a single fixed port is the same as dynamically allocated from a degraded port range. I didn't provide configuration for a single fixed port and even it doesn't work because of the design. If the restricted port range is not given and the reverse proxy is present then WebClientsManager.allocateClient throws an exception. Please look at this code

```
      if (viaProxyServer && !portsRestricted)
      {
         throw new IllegalStateException("Correct range: "
                  + "'webClient/portsRange/from' < 'webClient/portsRange/to' must be provided");
      }
```

> In cases 3 and 4, port XYZ can be:
>
> - dynamically allocated from any open port on the system (port == 0 in the directory)
> - dynamically allocated from a port range
> - a single fixed port (single-user only)

All these cases I tested on my host but no one else had tested them.

#168 - 06/30/2017 03:46 PM - Greg Shah

> It seems that a single fixed port is the same as dynamically allocated from a degraded port range. I didn't provide configuration for a single fixed port and even it doesn't work because of the design. If the restricted port range is not given and the reverse proxy is present then WebClientsManager.allocateClient throws an exception. Please look at this code

OK, this is fine. As long as the port range works with both cases 1 and 2, then it is good.

My next step is to get the POC directory.xml configured properly.

If I understand correctly, I only need to do these two things:

- Run ClientsToPortsGenerator. This will create the mapping text file and will enable the port range in the directory.
- Add the ACL for WebClientsManager.

Is any other directory.xml change needed?

#169 - 06/30/2017 03:51 PM - Sergey Ivanovskiy

It seems that there no another changes. I updated this poc and started postgresql server, now running ant deploy.all it takes 7-8 minutes, then I will change the required embedded web server files, the same as for hotel_gui embedded and will test these changes.

#170 - 06/30/2017 03:56 PM - Sergey Ivanovskiy

I need help with poc setup, I run ant deploy.all and got

```
nConnection(LogicalConnectionImpl.java:297)
     [java]     ... 22 more
     [java] Caused by: com.mchange.v2.resourcepool.CannotAcquireResourceException: A ResourcePool could not ac
quire a resource from its primary factory or source.
     [java]     at com.mchange.v2.resourcepool.BasicResourcePool.awaitAvailable(BasicResourcePool.java:1469)
     [java]     at com.mchange.v2.resourcepool.BasicResourcePool.prelimCheckoutResource(BasicResourcePool.java
:644)
     [java]     at com.mchange.v2.resourcepool.BasicResourcePool.checkoutResource(BasicResourcePool.java:554)
     [java]     at com.mchange.v2.c3p0.impl.C3P0PooledConnectionPool.checkoutAndMarkConnectionInUse(C3P0Pooled
ConnectionPool.java:758)
     [java]     at com.mchange.v2.c3p0.impl.C3P0PooledConnectionPool.checkoutPooledConnection(C3P0PooledConnec
tionPool.java:685)
     [java]     ... 26 more
     [java] Caused by: org.postgresql.util.PSQLException: FATAL: password authentication failed for user "fwd_
user"
     [java]     at org.postgresql.core.v3.ConnectionFactoryImpl.doAuthentication(ConnectionFactoryImpl.java:44
6)
     [java]     at org.postgresql.core.v3.ConnectionFactoryImpl.openConnectionImpl(ConnectionFactoryImpl.java:
220)
     [java]     at org.postgresql.core.ConnectionFactory.openConnection(ConnectionFactory.java:55)
     [java]     at org.postgresql.jdbc.PgConnection.<init>(PgConnection.java:219)
     [java]     at org.postgresql.Driver.makeConnection(Driver.java:407)
     [java]     at org.postgresql.Driver.connect(Driver.java:275)
     [java]     at com.mchange.v2.c3p0.DriverManagerDataSource.getConnection(DriverManagerDataSource.java:175)
     [java]     at com.mchange.v2.c3p0.WrapperConnectionPoolDataSource.getPooledConnection(WrapperConnectionPo
olDataSource.java:220)
     [java]     at com.mchange.v2.c3p0.WrapperConnectionPoolDataSource.getPooledConnection(WrapperConnectionPo
olDataSource.java:206)
     [java]     at com.mchange.v2.c3p0.impl.C3P0PooledConnectionPool$1PooledConnectionResourcePoolManager.acqu
ireResource(C3P0PooledConnectionPool.java:203)
     [java]     at com.mchange.v2.resourcepool.BasicResourcePool.doAcquire(BasicResourcePool.java:1138)
     [java]     at com.mchange.v2.resourcepool.BasicResourcePool.doAcquireAndDecrementPendingAcquiresWithinLoc
kOnSuccess(BasicResourcePool.java:1125)
     [java]     at com.mchange.v2.resourcepool.BasicResourcePool.access$700(BasicResourcePool.java:44)
     [java]     at com.mchange.v2.resourcepool.BasicResourcePool$ScatteredAcquireTask.run(BasicResourcePool.ja
va:1870)
     [java]     at com.mchange.v2.async.ThreadPoolAsynchronousRunner$PoolThread.run(ThreadPoolAsynchronousRunn
er.java:696)
     [java] Jun 30, 2017 10:54:32 PM com.mchange.v2.log.slf4j.Slf4jMLog$Slf4jMLogger$WarnLogger log
     [java] WARNING: com.mchange.v2.resourcepool.BasicResourcePool$ScatteredAcquireTask@7dc2a0df -- Acquisitio
n Attempt Failed!!! Clearing pending acquires. While trying to acquire a needed new resource, we failed to suc
ceed more than the maximum number of allowed acquisition attempts (30). Last acquisition attempt exception:
     [java] org.postgresql.util.PSQLException: FATAL: password authentication failed for user "fwd_user"
     [java]     at org.postgresql.core.v3.ConnectionFactoryImpl.doAuthentication(ConnectionFactoryImpl.java:44
6)
     [java]     at org.postgresql.core.v3.ConnectionFactoryImpl.openConnectionImpl(ConnectionFactoryImpl.java:
220)
     [java]     at org.postgresql.core.ConnectionFactory.openConnection(ConnectionFactory.java:55)
     [java]     at org.postgresql.jdbc.PgConnection.<init>(PgConnection.java:219)
     [java]     at org.postgresql.Driver.makeConnection(Driver.java:407)
     [java]     at org.postgresql.Driver.connect(Driver.java:275)
     [java]     at com.mchange.v2.c3p0.DriverManagerDataSource.getConnection(DriverManagerDataSource.java:175)
     [java]     at com.mchange.v2.c3p0.WrapperConnectionPoolDataSource.getPooledConnection(WrapperConnectionPo
olDataSource.java:220)
     [java]     at com.mchange.v2.c3p0.WrapperConnectionPoolDataSource.getPooledConnection(WrapperConnectionPo
olDataSource.java:206)
     [java]     at com.mchange.v2.c3p0.impl.C3P0PooledConnectionPool$1PooledConnectionResourcePoolManager.acqu
ireResource(C3P0PooledConnectionPool.java:203)
     [java]     at com.mchange.v2.resourcepool.BasicResourcePool.doAcquire(BasicResourcePool.java:1138)
     [java]     at com.mchange.v2.resourcepool.BasicResourcePool.doAcquireAndDecrementPendingAcquiresWithinLoc
kOnSuccess(BasicResourcePool.java:1125)
     [java]     at com.mchange.v2.resourcepool.BasicResourcePool.access$700(BasicResourcePool.java:44)
     [java]     at com.mchange.v2.resourcepool.BasicResourcePool$ScatteredAcquireTask.run(BasicResourcePool.ja
va:1870)
```

```
     [java]     at com.mchange.v2.async.ThreadPoolAsynchronousRunner$PoolThread.run(ThreadPoolAsynchronousRunn
er.java:696)
     [java]
     [java] Jun 30, 2017 10:54:32 PM com.mchange.v2.log.slf4j.Slf4jMLog$Slf4jMLogger$WarnLogger log
     [java] WARNING: Having failed to acquire a resource, com.mchange.v2.resourcepool.BasicResourcePool@f0da94
5 is interrupting all Threads waiting on a resource to check out. Will try again in response to new client req
uests.
     [java] Jun 30, 2017 10:54:32 PM com.mchange.v2.log.slf4j.Slf4jMLog$Slf4jMLogger$WarnLogger log
     [java] WARNING: com.mchange.v2.resourcepool.BasicResourcePool$ScatteredAcquireTask@6bb42a93 -- Acquisitio
n Attempt Failed!!! Clearing pending acquires. While trying to acquire a needed new resource, we failed to suc
ceed more than the maximum number of allowed acquisition attempts (30). Last acquisition attempt exception:
     [java] org.postgresql.util.PSQLException: FATAL: password authentication failed for user "fwd_user"
     [java]     at org.postgresql.core.v3.ConnectionFactoryImpl.doAuthentication(ConnectionFactoryImpl.java:44
6)
     [java]     at org.postgresql.core.v3.ConnectionFactoryImpl.openConnectionImpl(ConnectionFactoryImpl.java:
220)
     [java]     at org.postgresql.core.ConnectionFactory.openConnection(ConnectionFactory.java:55)
     [java]     at org.postgresql.jdbc.PgConnection.<init>(PgConnection.java:219)
     [java]     at org.postgresql.Driver.makeConnection(Driver.java:407)
     [java]     at org.postgresql.Driver.connect(Driver.java:275)
     [java]     at com.mchange.v2.c3p0.DriverManagerDataSource.getConnection(DriverManagerDataSource.java:175)
     [java]     at com.mchange.v2.c3p0.WrapperConnectionPoolDataSource.getPooledConnection(WrapperConnectionPo
olDataSource.java:220)
     [java]     at com.mchange.v2.c3p0.WrapperConnectionPoolDataSource.getPooledConnection(WrapperConnectionPo
olDataSource.java:206)
     [java]     at com.mchange.v2.c3p0.impl.C3P0PooledConnectionPool$1PooledConnectionResourcePoolManager.acqu
ireResource(C3P0PooledConnectionPool.java:203)
     [java]     at com.mchange.v2.resourcepool.BasicResourcePool.doAcquire(BasicResourcePool.java:1138)
     [java]     at com.mchange.v2.resourcepool.BasicResourcePool.doAcquireAndDecrementPendingAcquiresWithinLoc
kOnSuccess(BasicResourcePool.java:1125)
     [java]     at com.mchange.v2.resourcepool.BasicResourcePool.access$700(BasicResourcePool.java:44)
     [java]     at com.mchange.v2.resourcepool.BasicResourcePool$ScatteredAcquireTask.run(BasicResourcePool.ja
va:1870)
     [java]     at com.mchange.v2.async.ThreadPoolAsynchronousRunner$PoolThread.run(ThreadPoolAsynchronousRunn
er.java:696)
     [java]
     [java] Jun 30, 2017 10:54:32 PM com.mchange.v2.log.slf4j.Slf4jMLog$Slf4jMLogger$WarnLogger log
     [java] WARNING: com.mchange.v2.resourcepool.BasicResourcePool$ScatteredAcquireTask@341915f0 -- Acquisitio
n Attempt Failed!!! Clearing pending acquires. While trying to acquire a needed new resource, we failed to suc
ceed more than the maximum number of allowed acquisition attempts (30). Last acquisition attempt exception:
     [java] org.postgresql.util.PSQLException: FATAL: password authentication failed for user "fwd_user"
     [java]     at org.postgresql.core.v3.ConnectionFactoryImpl.doAuthentication(ConnectionFactoryImpl.java:44
6)
     [java]     at org.postgresql.core.v3.ConnectionFactoryImpl.openConnectionImpl(ConnectionFactoryImpl.java:
220)
     [java]     at org.postgresql.core.ConnectionFactory.openConnection(ConnectionFactory.java:55)
```

#171 - 06/30/2017 04:00 PM - Greg Shah

Are you on revision 53 of the POC project? You are probably on rev 52 which was hard coded to PostgreSQL.

Try rev 53. Please note that there are some values in the directory that have to be hand-modified to make it work:

UNIX_USER_NAME
PROJECT_PARENT_PATH

#172 - 06/30/2017 04:02 PM - Sergey Ivanovskiy

Ok, but it worked on my host with Posgresql before that.

#173 - 06/30/2017 04:05 PM - Greg Shah

Sergey Ivanovskiy wrote:

> Ok, but it worked on my host with Posgresql before that.

Hotel was setup for postgresql. But the customer POC code was only ever working on H2 until a couple of days ago. We've backed that out (as the default) temporarily.

For now, just use the H2 path. The work in this task doesn't need postgresql.

#174 - 06/30/2017 04:29 PM - Sergey Ivanovskiy

Thank you, the poc is built now. I forgot that the embedded client needs a direct access too, but I changed its root consistently to /embedded, so the web server also need to be changed. I tested via proxy. It should not be a problem, I would like to change a direct access to https://localhost:8443/embedded

#175 - 06/30/2017 04:30 PM - Greg Shah

> It should be a problem, I would like to change a direct access to https://localhost:8443/embedded

Go ahead.

#176 - 06/30/2017 04:37 PM - Sergey Ivanovskiy

Greg Shah wrote:

> > It should be a problem, I would like to change a direct access to https://localhost:8443/embedded

> Go ahead.

It is stupid, I fooled myself, it doesn't need to do any changes, it is enough to change an access url to https://localhost for the embedded and to change the reverse proxy to

```
ProxyPass / https://localhost:8443/
ProxyPassReverse / https://localhost:8443/
```

#177 - 06/30/2017 04:48 PM - Sergey Ivanovskiy

Sergey Ivanovskiy wrote:

> Greg, Constantin these changes work with the Apache configuration provided by note 152, I did not use HTML rewrite module to fix urls inside html, because it is impossible for our case. Please review this fix, it works :)

Don't mind, these changes are not required at all. We can simply have another url and different proxy settings. For the embedded client it can be https://localhost/
and mapping can be

```
ProxyPass / https://localhost:8443/
ProxyPassReverse / https://localhost:8443/
```

Greg, if it is enough, I have done.

#178 - 06/30/2017 04:53 PM - Sergey Ivanovskiy

- File deleted (required_changes.txt)

#179 - 06/30/2017 04:54 PM - Greg Shah

Can you check in your changes to the POC embedded code?

#180 - 06/30/2017 05:00 PM - Sergey Ivanovskiy

Greg Shah wrote:

> Can you check in your changes to the POC embedded code?

It doesn't need to change the source code if you are agree to use https://localhost/ for https://localhost:8443/. Please recall the account/password for poc embedded client.

#181 - 06/30/2017 05:03 PM - Sergey Ivanovskiy

Don't mind, I found REDACTED/REDACTED

#182 - 06/30/2017 05:08 PM - Sergey Ivanovskiy

Do you need my help with Apache configuration?

#183 - 06/30/2017 05:31 PM - Greg Shah

Yes.

I'm about to work on that next.

Please make a list of the following:

1. Packages to install to get Apache and all modules in place.
2. Command line module enable/site enable statements.
3. Specific configuration files OR changes to configuration files that are needed.
4. Whatever else needs to be done that I'm not thinking of.

I need a kind of "cookbook" for getting this going. I will take that and script it in Ansible.

#184 - 06/30/2017 05:35 PM - Greg Shah

Also, please update the directory.xml to have the right ACL and the new port range support for 10 clients.

Include the client match list in the apache cfg.

#185 - 06/30/2017 06:18 PM - Sergey Ivanovskiy

- File default-ssl.conf added

- File map.clients-to-backends added

Please look at these resources:

1) https://www.digitalocean.com/community/tutorials/how-to-install-linux-apache-mysql-php-lamp-stack-on-ubuntu-16-04#step-1-install-apache-and-allow-in-firewall

```
sudo apt-get update
sudo apt-get install apache2
```

2) https://www.digitalocean.com/community/tutorials/how-to-use-apache-http-server-as-reverse-proxy-using-mod_proxy-extension

```
sudo a2enmod proxy
sudo a2enmod proxy_http
sudo a2enmod proxy_ajp
sudo a2enmod rewrite
sudo a2enmod deflate
sudo a2enmod headers
sudo a2enmod proxy_balancer
sudo a2enmod proxy_connect
sudo a2enmod proxy_html
sudo a2enmod proxy_wstunnel
```

and use this configuration for default-ssl.conf and place required ssl certificates according to this configuration

```
<IfModule mod_ssl.c>
    <VirtualHost _default_:443>
        ServerAdmin webmaster@localhost
                ServerName acme
                ServerAlias acme
        DocumentRoot /var/www/html

        # Available loglevels: trace8, ..., trace1, debug, info, notice, warn,
        # error, crit, alert, emerg.
        # It is also possible to configure the loglevel for particular
        # modules, e.g.
        #LogLevel info ssl:warn

        ErrorLog ${APACHE_LOG_DIR}/error.log
        CustomLog ${APACHE_LOG_DIR}/access.log combined

        # For most configuration files from conf-available/, which are
        # enabled or disabled at a global level, it is possible to
        # include a line for only one particular virtual host. For example the
        # following line enables the CGI configuration for this host only
        # after it has been globally disabled with "a2disconf".
        #Include conf-available/serve-cgi-bin.conf

        #   SSL Engine Switch:
        #   Enable/Disable SSL for this virtual host.
        SSLEngine on
        SSLProxyEngine On
        #   A self-signed (snakeoil) certificate can be created by installing
        #   the ssl-cert package. See
        #   /usr/share/doc/apache2/README.Debian.gz for more info.
        #   If both key and certificate are stored in the same file, only the
        #   SSLCertificateFile directive is needed.
        SSLCertificateFile    /etc/apache2/ssl/certs/apache.crt
        SSLCertificateKeyFile /etc/apache2/ssl/private/apache.key

        #   Server Certificate Chain:
        #   Point SSLCertificateChainFile at a file containing the
        #   concatenation of PEM encoded CA certificates which form the
        #   certificate chain for the server certificate. Alternatively
        #   the referenced file can be the same as SSLCertificateFile
        #   when the CA certificates are directly appended to the server
        #   certificate for convinience.
        #SSLCertificateChainFile /etc/apache2/ssl.crt/server-ca.crt

        #   Certificate Authority (CA):
        #   Set the CA certificate verification path where to find CA
```

```
#   certificates for client authentication or alternatively one
#   huge file containing all of them (file must be PEM encoded)
#   Note: Inside SSLCACertificatePath you need hash symlinks
#         to point to the certificate files. Use the provided
#         Makefile to update the hash symlinks after changes.
#SSLCACertificatePath /etc/ssl/certs/
#SSLCACertificateFile /etc/apache2/ssl.crt/ca-bundle.crt

#   Certificate Revocation Lists (CRL):
#   Set the CA revocation path where to find CA CRLs for client
#   authentication or alternatively one huge file containing all
#   of them (file must be PEM encoded)
#   Note: Inside SSLCARevocationPath you need hash symlinks
#         to point to the certificate files. Use the provided
#         Makefile to update the hash symlinks after changes.
#SSLCARevocationPath /etc/apache2/ssl.crl/
#SSLCARevocationFile /etc/apache2/ssl.crl/ca-bundle.crl

#   Client Authentication (Type):
#   Client certificate verification type and depth.  Types are
#   none, optional, require and optional_no_ca.  Depth is a
#   number which specifies how deeply to verify the certificate
#   issuer chain before deciding the certificate is not valid.
#SSLVerifyClient require
#SSLVerifyDepth  10

#   SSL Engine Options:
#   Set various options for the SSL engine.
#   o FakeBasicAuth:
#     Translate the client X.509 into a Basic Authorisation.  This means that
#     the standard Auth/DBMAuth methods can be used for access control.  The
#     user name is the `one line' version of the client's X.509 certificate.
#     Note that no password is obtained from the user. Every entry in the user
#     file needs this password: `xxj31ZMTZzkVA'.
#   o ExportCertData:
#     This exports two additional environment variables: SSL_CLIENT_CERT and
#     SSL_SERVER_CERT. These contain the PEM-encoded certificates of the
#     server (always existing) and the client (only existing when client
#     authentication is used). This can be used to import the certificates
#     into CGI scripts.
#   o StdEnvVars:
#     This exports the standard SSL/TLS related `SSL_*' environment variables.
#     Per default this exportation is switched off for performance reasons,
#     because the extraction step is an expensive operation and is usually
#     useless for serving static content. So one usually enables the
#     exportation for CGI and SSI requests only.
#   o OptRenegotiate:
#     This enables optimized SSL connection renegotiation handling when SSL
#     directives are used in per-directory context.
#SSLOptions +FakeBasicAuth +ExportCertData +StrictRequire
<FilesMatch "\.(cgi|shtml|phtml|php)$">
        SSLOptions +StdEnvVars
</FilesMatch>
<Directory /usr/lib/cgi-bin>
        SSLOptions +StdEnvVars
</Directory>

#   SSL Protocol Adjustments:
#   The safe and default but still SSL/TLS standard compliant shutdown
#   approach is that mod_ssl sends the close notify alert but doesn't wait for
#   the close notify alert from client. When you need a different shutdown
#   approach you can use one of the following variables:
#   o ssl-unclean-shutdown:
#     This forces an unclean shutdown when the connection is closed, i.e. no
#     SSL close notify alert is send or allowed to received.  This violates
#     the SSL/TLS standard but is needed for some brain-dead browsers. Use
#     this when you receive I/O errors because of the standard approach where
#     mod_ssl sends the close notify alert.
#   o ssl-accurate-shutdown:
#     This forces an accurate shutdown when the connection is closed, i.e. a
#     SSL close notify alert is send and mod_ssl waits for the close notify
#     alert of the client. This is 100% SSL/TLS standard compliant, but in
#     practice often causes hanging connections with brain-dead browsers. Use
#     this only for browsers where you know that their SSL implementation
#     works correctly.
```

```
        #   Notice: Most problems of broken clients are also related to the HTTP
        #   keep-alive facility, so you usually additionally want to disable
        #   keep-alive for those clients, too. Use variable "nokeepalive" for this.
        #   Similarly, one has to force some clients to use HTTP/1.0 to workaround
        #   their broken HTTP/1.1 implementation. Use variables "downgrade-1.0" and
        #   "force-response-1.0" for this.
        # BrowserMatch "MSIE [2-6]" \
        #        nokeepalive ssl-unclean-shutdown \
        #        downgrade-1.0 force-response-1.0

ProxyRequests Off
SSLProxyEngine On
ProxyReceiveBufferSize 4096
SSLProxyVerify none
SSLProxyCheckPeerCN off
SSLProxyCheckPeerName off
RewriteEngine On

RewriteMap  clients-to-backends   "txt:/etc/apache2/map.clients-to-backends"

RewriteCond %{HTTP:Connection} Upgrade [NC]
RewriteCond %{HTTP:Upgrade} websocket [NC]
RewriteRule /server/([^/]+)/(.*)  wss://${clients-to-backends:$1}/$2 [P,L]
RewriteRule /server/([^/]+)/(.*) https://${clients-to-backends:$1}/$2 [P,L]

ProxyPreserveHost On
ProxyAddHeaders On

ProxyPass /gui  https://localhost:7443/gui
ProxyPassReverse /gui https://localhost:7443/gui

ProxyPass /  https://localhost:8443/
ProxyPassReverse / https://localhost:8443/

    </VirtualHost>
</IfModule>
# vim: syntax=apache ts=4 sw=4 sts=4 sr noet
```

3)
https://www.digitalocean.com/community/tutorials/how-to-set-up-apache-virtual-hosts-on-ubuntu-14-04-lts
Enable this configuration by

```
sudo a2ensite default-ssl.conf
sudo service apache2 restart
```

For my host configuration please look at the note 150

#186 - 06/30/2017 06:19 PM - Sergey Ivanovskiy

Committed revision 55 modified poc directory.xml.

#187 - 06/30/2017 06:19 PM - Sergey Ivanovskiy

- File deleted (default-ssl.conf)

#188 - 06/30/2017 06:20 PM - Sergey Ivanovskiy

- File deleted (map.clients-to-backends)

#189 - 06/30/2017 06:27 PM - Sergey Ivanovskiy

Also after your configuration has been done, you can check the installed modules by running

```
sudo a2query -m
```

on my host it lists these modules

```
access_compat (enabled by maintainer script)
alias (enabled by maintainer script)
setenvif (enabled by maintainer script)
headers (enabled by site administrator)
proxy_connect (enabled by site administrator)
deflate (enabled by maintainer script)
filter (enabled by maintainer script)
authz_user (enabled by maintainer script)
proxy_wstunnel (enabled by site administrator)
dir (enabled by maintainer script)
authz_core (enabled by maintainer script)
proxy_html (enabled by site administrator)
auth_basic (enabled by maintainer script)
negotiation (enabled by maintainer script)
env (enabled by maintainer script)
authn_file (enabled by maintainer script)
proxy (enabled by site administrator)
status (enabled by maintainer script)
authn_core (enabled by maintainer script)
proxy_http (enabled by site administrator)
socache_shmcb (enabled by site administrator)
rewrite (enabled by site administrator)
mpm_prefork (enabled by site administrator)
autoindex (enabled by maintainer script)
authz_host (enabled by maintainer script)
mime (enabled by maintainer script)
ssl (enabled by site administrator)
```

#190 - 06/30/2017 06:30 PM - Sergey Ivanovskiy

Please notice modules that enabled by site administrator. All these modules must be installed if you would like to get the similar configuration on your host.

#191 - 06/30/2017 06:34 PM - Greg Shah

Do all of these get installed with the apache2 package?

#192 - 06/30/2017 06:43 PM - Sergey Ivanovskiy

Greg Shah wrote:

> Do all of these get installed with the apache2 package?

I am not sure, but it seems that they are included in this package.

#193 - 06/30/2017 06:45 PM - Sergey Ivanovskiy

openssl package must be installed.

#194 - 06/30/2017 06:49 PM - Sergey Ivanovskiy

At first it is required to create ssl virtual host and it seems that I followed this article
https://www.digitalocean.com/community/tutorials/how-to-create-a-ssl-certificate-on-apache-for-ubuntu-14-04

#195 - 06/30/2017 06:54 PM - Greg Shah

Can't we just pass through the SSL of the FWD server's? We don't want to generate yet another certificate that is only used for reverse proxy anyway (a pass through);

#196 - 06/30/2017 06:56 PM - Sergey Ivanovskiy

Yes, I think it helps to fix embedded client's certificate issue. It seems that you wrote about it when preparing an access to demo.

#197 - 06/30/2017 06:59 PM - Greg Shah

I setup certificates for the FWD server and the embedded web server for demo.goldencode.com. Those just replaced the certs in the poc project.

I did not setup an Apache proxy configuration with certificates. What I am hoping is that Apache can be setup to just pass through the SSL.

If that isn't possible, then we would want to re-use the certs that are already in the poc project.

#198 - 06/30/2017 07:07 PM - Sergey Ivanovskiy

I followed https://www.digitalocean.com/community/tutorials/how-to-use-apache-http-server-as-reverse-proxy-using-mod_proxy-extension.

#199 - 06/30/2017 07:16 PM - Sergey Ivanovskiy

Planning to log out sooner.

#200 - 06/30/2017 07:17 PM - Greg Shah

Constantin: Is the ChUI regression testing still running?

#201 - 06/30/2017 07:19 PM - Constantin Asofiei

Greg Shah wrote:

> Constantin: Is the ChUI regression testing still running?

CTRL-C I think is passed (cumulative, 3-way and 2-way sets pass in different runs, couldn't get a clean run...).

main part is still running.

#202 - 06/30/2017 09:07 PM - Constantin Asofiei

main part passed, too.

#203 - 06/30/2017 09:14 PM - Greg Shah

Are you OK with merging to trunk?

#204 - 06/30/2017 09:15 PM - Greg Shah

Also: I plan to enable virtual desktop in the poc directory. Is there any reason to disable it?

#205 - 06/30/2017 09:22 PM - Constantin Asofiei

Greg Shah wrote:

> Are you OK with merging to trunk?

Yes, I don't see any issues.

> Also: I plan to enable virtual desktop in the poc directory. Is there any reason to disable it?

I don't see one, but I'll do a quick test.

#206 - 06/30/2017 09:25 PM - Greg Shah

I'll test it now.

Please consider the cert re-gen issue.

#207 - 06/30/2017 09:30 PM - Greg Shah

Virtual desktop mode works in the POC. AND because of the port range support, I can get multiple simultaneous sessions going to the same server!

If we can provide the mechanism to regen the cert for an arbitrary hostname, then we can meet our deadline.

#208 - 06/30/2017 09:33 PM - Constantin Asofiei

Greg Shah wrote:

> If we can provide the mechanism to regen the cert for an arbitrary hostname, then we can meet our deadline.

This is a little tricky; we have two certificates in use via web:

- the "embedded" process - this is not linked to FWD, I've added it in the directory just to be able to use the FWD tool to generate the certificate
- the "server" process - in use by the FWD iframe. This is not standalone, so it can't be easily switched.

I can add some property to the process account (or in webClient/<process-id>/<host-name>) to be able to configure the common name, or otherwise change SSLCertGenUtil to ask for the common name to be used.

But regardless, after the certs are re-issued, the embedded_server_cfg_launch.sh (and .cmd) will need to be reconfigured with the proper passwords. And when these are set, some characters will require to be escaped (% in Windows and), $ and others in Linux).

Ideally, I think we would want the web certificates to not be linked to any FWD internals (i.e. not use the server certificate), and be just a standalone certificate, which can be managed externally and just configured in the directory.

So, the fast way at this time I think is to use the webClient/<process-id>/<host-name> approach.

#209 - 06/30/2017 09:38 PM - Greg Shah

I am already automatically updating the directory.xml placeholders using the Ansible scripting. We can add another placeholder and it is easy to handle.

Likewise, we can pass a name as part of the ansible command line (I am already passing many other parameters this way). And we can pass this name to SSLCertGenUtil from Ansible if that is helpful.

I can make similar edits to embedded_server_cfg_launch.sh BUT why not reuse the same passwords? This is the simplest solution and will work for the poc.

#210 - 06/30/2017 09:43 PM - Constantin Asofiei

Greg Shah wrote:

> I am already automatically updating the directory.xml placeholders using the Ansible scripting. We can add another placeholder and it is easy to handle.

Yes, I'll add a directory configuration under company/hostnames/<process>/<host> , with <host> needing to be a placeholder.

> Likewise, we can pass a name as part of the ansible command line (I am already passing many other parameters this way). And we can pass this name to SSLCertGenUtil from Ansible if that is helpful.

SSLCertGenUtil reads a lot of input from STDIN. Can this be managed in ansible? Otherwise I'll need to read the input from the arguments (or use a 'auto' mode).

> I can make similar edits to embedded_server_cfg_launch.sh BUT why not reuse the same passwords? This is the simplest solution and will work for the poc.

We can re-use the passwords used to encrypt the private keys in the directory. But it will be used to encrypt both the private key and the key store.

#211 - 06/30/2017 09:44 PM - Greg Shah

> Otherwise I'll need to read the input from the arguments

Yes, let's make these arguments. I can pass them in as needed.

> We can re-use the passwords used to encrypt the private keys in the directory. But it will be used to encrypt both the private key and the key store.

This is fine for tonight.

#212 - 06/30/2017 10:43 PM - Constantin Asofiei

Greg, I'm almost done with this: do you want to add 're-use root CA', too? Because otherwise this would need to be re-installed.

#213 - 06/30/2017 10:49 PM - Greg Shah

Yes.

#214 - 06/30/2017 10:50 PM - Constantin Asofiei

Greg Shah wrote:

> Yes.

OK, then I'll need to save its encryption password for its private key in the directory.

#215 - 06/30/2017 10:50 PM - Greg Shah

Or we can pass it in as a parameter.

#216 - 06/30/2017 11:40 PM - Sergey Ivanovskiy

There is this option webClient/portsRange/forwardedHost that can be important to set in Amazon environment. (https://proj.goldencode.com/issues/3236#note-47) It is a frontend proxy server host (external host that is visible outside the private network).

```
          <node class="container" name="portsrange">
            <node class="integer" name="from">
              <node-attribute name="value" value="7449"/>
            </node>
            <node class="integer" name="to">
              <node-attribute name="value" value="7459"/>
            </node>
            <node class="string" name="nameprefix">
              <node-attribute name="value" value="client"/>
            </node>
            <node class="string" name="forwardedHost">
              <node-attribute name="value" value="www.goldencode.com"/>
            </node>
<!-- it can be skipped since its value is always 'https' -->
            <node class="string" name="forwardedProto">
              <node-attribute name="value" value="https"/>
            </node>
          </node>
```

#217 - 07/01/2017 12:12 AM - Constantin Asofiei

Greg, 2683a rev 11180 contains the certificate-related changes (and the chrome 'subject-alternate-name' fix, too). Also, POC rev 60 contains the new certificates.

To be able to use SSLCertGenUtil and custom common names, these are needed:

1. in directory.xml, search for hostnames and replace localhost with your placeholder
2. execute SSLCertGenUtil with these arguments: directory.xml 2048 65337 yes yes fBMaU6k(r6<G^tUeZEg=BW48z3mM4M29 yes no yes no yes yes srv-certs.store yes root-ca-pk.store, from deploy/server.
3. execute this command in deploy/server to generate the .der:

   ```
   keytool -exportcert -alias hotel-root -keystore srv-certs.store -file hotel-root.der
   ```

   and use 169ghwa^G0f%O9p<`feODy1q7NTFWI8j as password (will be read from STDIN).
4. import the hotel-root.der into the browser(s).
5. re-install spawner.

After 2683a rev 11180, each time SSLCertGenUtil is executed with the above parameters, the root CA will be re-used and the encryption password will be used for the stores.

#218 - 07/01/2017 05:54 AM - Constantin Asofiei

Greg, I think I gave you the wrong password, and I need to re-generate them again - I get the same now (even from Eclipse), something messed up when I committed the directory.

#219 - 07/03/2017 05:54 AM - Greg Shah

- Status changed from New to Closed

- Assignee set to Sergey Ivanovskiy

- % Done changed from 0 to 100

#220 - 07/03/2017 05:54 AM - Greg Shah

2683a was merged to trunk as revision 11154.

Files

default-ssl.conf	6.97 KB	06/30/2017	Sergey Ivanovskiy
map.clients-to-backends	288 Bytes	06/30/2017	Sergey Ivanovskiy