

**git\_comments:**

1. If timezone is provided by tunnel parameter, and not overridden by connection parameter, set it.
2. \* \* Return the timezone as reported by the client. \* \* @returns \* A string value of the timezone reported by the client.
3. \* \* The timezone report by the client.
4. \* \* Set the string value of the timezone. \* \* @param timezone \* The string value of the timezone reported by the client.
5. \* \* Returns the value of the timezone parameter declared within the \* tunnel request. \* \* @return \* The string value of the timezone parameter as reported by \* the client.
6. \* \* The name of the parameter specifying the timezone of the client.
7. Get the timezone value

**git\_commits:**

1. **summary:** GUACAMOLE-422: Add timezone to tunnel connections.  
**message:** GUACAMOLE-422: Add timezone to tunnel connections.

**github\_issues:****github\_issues\_comments:****github\_pulls:****github\_pulls\_comments:****github\_pulls\_reviews:****jira\_issues:**

1. **summary:** Timezone Redirection

**description:** It would be nice to support full timezone redirection from the browser client through to a RDP server. There are a few challenges to making this work: - Detect timezone at browser and pass that through to the servlet side, somewhere. - Handle timezone on servlet side and pass through to guacd - Allow guacd to process the timezone and add it to the RDP connection. For browser side, there is a JavaScript library, jstz, that can do the detection and provide the timezone. Handling it on the servlet side, however, requires some way to pass it in during the connection. This could be done as a parameter at login time, or a separate api call. Once it is into the servlet, there has to be a way to process it as a parameter to a connection. I'm thinking maybe a token, but not sure that's the right way to go. Once it's on the connection it can pass through to guacd. The biggest issue, I think, is then getting guacd to be able to add it to a FreeRDP connection. From my looking at the current FreeRDP source code (stable-1.1 branch), FreeRDP currently detects the timezone on the system where it's running, first using the TZ variable, then looking at /etc/localtime and/or /etc/TZ on the system. This means that it's always going to redirect the timezone of the system where guacd is running, and not the timezone of the browser. Unfortunately there don't seem to be any available functions for sending FreeRDP a string representation of a timezone and having it process it that way, and all of the timezone information is located in the FreeRDP .c file, making it difficult to write a method for handling this without duplicating a lot of code and all of the timezone tables. So, not entirely sure the best route on that angle at this point, but we'll see. Any input on whether this is a reasonable thing to address, or a corner-case not worth handling?

2. **summary:** Timezone Redirection

**description:** It would be nice to support full timezone redirection from the browser client through to a RDP server. There are a few challenges to making this work: - Detect timezone at browser and pass that through to the servlet side, somewhere. - Handle timezone on servlet side and pass through to guacd - Allow guacd to process the timezone and add it to the RDP connection. For browser side, there is a JavaScript library, jstz, that can do the detection and provide the timezone. Handling it on the servlet side, however, requires some way to pass it in during the connection. This could be done as a parameter at login time, or a separate api call. Once it is into the servlet, there has to be a way to process it as a parameter to a connection. I'm thinking maybe a token, but not sure that's the right way to go. Once it's on the connection it can pass through to guacd. The biggest issue, I think, is then getting guacd to be able to add it to a FreeRDP connection. From my looking at the current FreeRDP source code (stable-1.1 branch), FreeRDP currently detects the timezone on the system where it's running, first using the TZ variable, then looking at /etc/localtime and/or /etc/TZ on the system. This means that it's always going to redirect the timezone of the system where guacd is running, and not the timezone of the browser. Unfortunately there don't seem to be any available functions for sending FreeRDP a string representation of a timezone and having it process it that way, and all of the timezone information is located in the FreeRDP .c file, making it difficult to write a method for handling this without duplicating a lot of code and all of the timezone tables. So, not entirely sure the best route on that angle at this point, but we'll see. Any input on whether this is a reasonable thing to address, or a corner-case not worth handling?

3. **summary:** Timezone Redirection

**description:** It would be nice to support full timezone redirection from the browser client through to a RDP server. There are a few challenges to making this work: - Detect timezone at browser and pass that through to the servlet side, somewhere. - Handle timezone on servlet side and pass through to guacd - Allow guacd to process the timezone and add it to the RDP connection. For browser side, there is a JavaScript library, jstz, that can do the detection and provide the timezone. Handling it on the servlet side, however, requires some way to pass it in during the connection. This could be done as a parameter at login time, or a separate api call. Once it is into the servlet, there has to be a way to process it as a parameter to a connection. I'm thinking maybe a token, but not sure that's the right way to go. Once it's on the connection it can pass through to guacd. The biggest issue, I think, is then getting guacd to be able to add it to a FreeRDP connection. From my looking at the current

FreeRDP source code (stable-1.1 branch), FreeRDP currently detects the timezone on the system where it's running, first using the TZ variable, then looking at /etc/localtime and/or /etc/TZ on the system. This means that it's always going to redirect the timezone of the system where guacd is running, and not the timezone of the browser. Unfortunately there don't seem to be any available functions for sending FreeRDP a string representation of a timezone and having it process it that way, and all of the timezone information is located in the FreeRDP .c file, making it difficult to write a method for handling this without duplicating a lot of code and all of the timezone tables. So, not entirely sure the best route on that angle at this point, but we'll see. Any input on whether this is a reasonable thing to address, or a corner-case not worth handling?

4. **summary:** Timezone Redirection

**description:** It would be nice to support full timezone redirection from the browser client through to a RDP server. There are a few challenges to making this work: - Detect timezone at browser and pass that through to the servlet side, somewhere. - Handle timezone on servlet side and pass through to guacd - Allow guacd to process the timezone and add it to the RDP connection. For browser side, there is a JavaScript library, jstz, that can do the detection and provide the timezone. Handling it on the servlet side, however, requires some way to pass it in during the connection. This could be done as a parameter at login time, or a separate api call. Once it is into the servlet, there has to be a way to process it as a parameter to a connection. I'm thinking maybe a token, but not sure that's the right way to go. Once it's on the connection it can pass through to guacd. The biggest issue, I think, is then getting guacd to be able to add it to a FreeRDP connection. From my looking at the current FreeRDP source code (stable-1.1 branch), FreeRDP currently detects the timezone on the system where it's running, first using the TZ variable, then looking at /etc/localtime and/or /etc/TZ on the system. This means that it's always going to redirect the timezone of the system where guacd is running, and not the timezone of the browser. Unfortunately there don't seem to be any available functions for sending FreeRDP a string representation of a timezone and having it process it that way, and all of the timezone information is located in the FreeRDP .c file, making it difficult to write a method for handling this without duplicating a lot of code and all of the timezone tables. So, not entirely sure the best route on that angle at this point, but we'll see. Any input on whether this is a reasonable thing to address, or a corner-case not worth handling?

**label:** code-design

5. **summary:** Timezone Redirection

**description:** It would be nice to support full timezone redirection from the browser client through to a RDP server. There are a few challenges to making this work: - Detect timezone at browser and pass that through to the servlet side, somewhere. - Handle timezone on servlet side and pass through to guacd - Allow guacd to process the timezone and add it to the RDP connection. For browser side, there is a JavaScript library, jstz, that can do the detection and provide the timezone. Handling it on the servlet side, however, requires some way to pass it in during the connection. This could be done as a parameter at login time, or a separate api call. Once it is into the servlet, there has to be a way to process it as a parameter to a connection. I'm thinking maybe a token, but not sure that's the right way to go. Once it's on the connection it can pass through to guacd. The biggest issue, I think, is then getting guacd to be able to add it to a FreeRDP connection. From my looking at the current FreeRDP source code (stable-1.1 branch), FreeRDP currently detects the timezone on the system where it's running, first using the TZ variable, then looking at /etc/localtime and/or /etc/TZ on the system. This means that it's always going to redirect the timezone of the system where guacd is running, and not the timezone of the browser. Unfortunately there don't seem to be any available functions for sending FreeRDP a string representation of a timezone and having it process it that way, and all of the timezone information is located in the FreeRDP .c file, making it difficult to write a method for handling this without duplicating a lot of code and all of the timezone tables. So, not entirely sure the best route on that angle at this point, but we'll see. Any input on whether this is a reasonable thing to address, or a corner-case not worth handling?

**label:** code-design

6. **summary:** Timezone Redirection

**description:** It would be nice to support full timezone redirection from the browser client through to a RDP server. There are a few challenges to making this work: - Detect timezone at browser and pass that through to the servlet side, somewhere. - Handle timezone on servlet side and pass through to guacd - Allow guacd to process the timezone and add it to the RDP connection. For browser side, there is a JavaScript library, jstz, that can do the detection and provide the timezone. Handling it on the servlet side, however, requires some way to pass it in during the connection. This could be done as a parameter at login time, or a separate api call. Once it is into the servlet, there has to be a way to process it as a parameter to a connection. I'm thinking maybe a token, but not sure that's the right way to go. Once it's on the connection it can pass through to guacd. The biggest issue, I think, is then getting guacd to be able to add it to a FreeRDP connection. From my looking at the current FreeRDP source code (stable-1.1 branch), FreeRDP currently detects the timezone on the system where it's running, first using the TZ variable, then looking at /etc/localtime and/or /etc/TZ on the system. This means that it's always going to redirect the timezone of the system where guacd is running, and not the timezone of the browser. Unfortunately there don't seem to be any available functions for sending FreeRDP a string representation of a timezone and having it process it that way, and all of the timezone information is located in the FreeRDP .c file, making it difficult to write a method for handling this without duplicating a lot of code and all of the timezone tables. So, not entirely sure the best route on that angle at this point, but we'll see. Any input on whether this is a reasonable thing to address, or a corner-case not worth handling?

7. **summary:** Timezone Redirection

**description:** It would be nice to support full timezone redirection from the browser client through to a RDP server. There are a few challenges to making this work: - Detect timezone at browser and pass that through to the servlet side, somewhere. - Handle timezone on servlet side and pass through to guacd - Allow guacd to process the timezone and add it to the RDP connection. For browser side, there is a JavaScript library, jstz, that can do the detection and provide the timezone. Handling it on the servlet side, however, requires some way to pass it in during the connection. This could be done as a parameter at login time, or a separate api call. Once it is into the servlet, there has to be a way to process it as a parameter to a connection. I'm thinking maybe a token, but not sure that's the right way to go. Once it's on the connection it can pass through to guacd. The biggest issue, I think, is then getting guacd to be able to add it to a FreeRDP connection. From my looking at the current FreeRDP source code (stable-1.1 branch), FreeRDP currently detects the timezone on the system where it's running, first using the TZ variable, then looking at /etc/localtime and/or /etc/TZ on the system. This means that it's always going to redirect the timezone of the system where guacd is running, and not the timezone of the browser. Unfortunately there don't seem to be any available functions for sending FreeRDP a string representation of a timezone and having it process it that way, and all of the

timezone information is located in the FreeRDP .c file, making it difficult to write a method for handling this without duplicating a lot of code and all of the timezone tables. So, not entirely sure the best route on that angle at this point, but we'll see. Any input on whether this is a reasonable thing to address, or a corner-case not worth handling?

**label:** code-design

8. **summary:** Timezone Redirection

**description:** It would be nice to support full timezone redirection from the browser client through to a RDP server. There are a few challenges to making this work: - Detect timezone at browser and pass that through to the servlet side, somewhere. - Handle timezone on servlet side and pass through to guacd - Allow guacd to process the timezone and add it to the RDP connection. For browser side, there is a JavaScript library, jstz, that can do the detection and provide the timezone. Handling it on the servlet side, however, requires some way to pass it in during the connection. This could be done as a parameter at login time, or a separate api call. Once it is into the servlet, there has to be a way to process it as a parameter to a connection. I'm thinking maybe a token, but not sure that's the right way to go. Once it's on the connection it can pass through to guacd. The biggest issue, I think, is then getting guacd to be able to add it to a FreeRDP connection. From my looking at the current FreeRDP source code (stable-1.1 branch), FreeRDP currently detects the timezone on the system where it's running, first using the TZ variable, then looking at /etc/localtime and/or /etc/TZ on the system. This means that it's always going to redirect the timezone of the system where guacd is running, and not the timezone of the browser. Unfortunately there don't seem to be any available functions for sending FreeRDP a string representation of a timezone and having it process it that way, and all of the timezone information is located in the FreeRDP .c file, making it difficult to write a method for handling this without duplicating a lot of code and all of the timezone tables. So, not entirely sure the best route on that angle at this point, but we'll see. Any input on whether this is a reasonable thing to address, or a corner-case not worth handling?

9. **summary:** Timezone Redirection

**description:** It would be nice to support full timezone redirection from the browser client through to a RDP server. There are a few challenges to making this work: - Detect timezone at browser and pass that through to the servlet side, somewhere. - Handle timezone on servlet side and pass through to guacd - Allow guacd to process the timezone and add it to the RDP connection. For browser side, there is a JavaScript library, jstz, that can do the detection and provide the timezone. Handling it on the servlet side, however, requires some way to pass it in during the connection. This could be done as a parameter at login time, or a separate api call. Once it is into the servlet, there has to be a way to process it as a parameter to a connection. I'm thinking maybe a token, but not sure that's the right way to go. Once it's on the connection it can pass through to guacd. The biggest issue, I think, is then getting guacd to be able to add it to a FreeRDP connection. From my looking at the current FreeRDP source code (stable-1.1 branch), FreeRDP currently detects the timezone on the system where it's running, first using the TZ variable, then looking at /etc/localtime and/or /etc/TZ on the system. This means that it's always going to redirect the timezone of the system where guacd is running, and not the timezone of the browser. Unfortunately there don't seem to be any available functions for sending FreeRDP a string representation of a timezone and having it process it that way, and all of the timezone information is located in the FreeRDP .c file, making it difficult to write a method for handling this without duplicating a lot of code and all of the timezone tables. So, not entirely sure the best route on that angle at this point, but we'll see. Any input on whether this is a reasonable thing to address, or a corner-case not worth handling?

10. **summary:** Timezone Redirection

**description:** It would be nice to support full timezone redirection from the browser client through to a RDP server. There are a few challenges to making this work: - Detect timezone at browser and pass that through to the servlet side, somewhere. - Handle timezone on servlet side and pass through to guacd - Allow guacd to process the timezone and add it to the RDP connection. For browser side, there is a JavaScript library, jstz, that can do the detection and provide the timezone. Handling it on the servlet side, however, requires some way to pass it in during the connection. This could be done as a parameter at login time, or a separate api call. Once it is into the servlet, there has to be a way to process it as a parameter to a connection. I'm thinking maybe a token, but not sure that's the right way to go. Once it's on the connection it can pass through to guacd. The biggest issue, I think, is then getting guacd to be able to add it to a FreeRDP connection. From my looking at the current FreeRDP source code (stable-1.1 branch), FreeRDP currently detects the timezone on the system where it's running, first using the TZ variable, then looking at /etc/localtime and/or /etc/TZ on the system. This means that it's always going to redirect the timezone of the system where guacd is running, and not the timezone of the browser. Unfortunately there don't seem to be any available functions for sending FreeRDP a string representation of a timezone and having it process it that way, and all of the timezone information is located in the FreeRDP .c file, making it difficult to write a method for handling this without duplicating a lot of code and all of the timezone tables. So, not entirely sure the best route on that angle at this point, but we'll see. Any input on whether this is a reasonable thing to address, or a corner-case not worth handling?

11. **summary:** Timezone Redirection

**description:** It would be nice to support full timezone redirection from the browser client through to a RDP server. There are a few challenges to making this work: - Detect timezone at browser and pass that through to the servlet side, somewhere. - Handle timezone on servlet side and pass through to guacd - Allow guacd to process the timezone and add it to the RDP connection. For browser side, there is a JavaScript library, jstz, that can do the detection and provide the timezone. Handling it on the servlet side, however, requires some way to pass it in during the connection. This could be done as a parameter at login time, or a separate api call. Once it is into the servlet, there has to be a way to process it as a parameter to a connection. I'm thinking maybe a token, but not sure that's the right way to go. Once it's on the connection it can pass through to guacd. The biggest issue, I think, is then getting guacd to be able to add it to a FreeRDP connection. From my looking at the current FreeRDP source code (stable-1.1 branch), FreeRDP currently detects the timezone on the system where it's running, first using the TZ variable, then looking at /etc/localtime and/or /etc/TZ on the system. This means that it's always going to redirect the timezone of the system where guacd is running, and not the timezone of the browser. Unfortunately there don't seem to be any available functions for sending FreeRDP a string representation of a timezone and having it process it that way, and all of the timezone information is located in the FreeRDP .c file, making it difficult to write a method for handling this without duplicating a lot of code and all of the timezone tables. So, not entirely sure the best route on that angle at this point, but we'll see. Any input on whether this is a reasonable thing to address, or a corner-case not worth handling?

12. **summary:** Timezone Redirection

**description:** It would be nice to support full timezone redirection from the browser client through to a RDP server. There are a few challenges to making this work: - Detect timezone at browser and pass that through to the servlet side, somewhere. - Handle timezone on servlet side and pass through to guacd - Allow guacd to process the timezone and add it to the RDP connection. For browser side, there is a JavaScript library, jstz, that can do the detection and provide the timezone. Handling it on the servlet side, however, requires some way to pass it in during the connection. This could be done as a parameter at login time, or a separate api call. Once it is into the servlet, there has to be a way to process it as a parameter to a connection. I'm thinking maybe a token, but not sure that's the right way to go. Once it's on the connection it can pass through to guacd. The biggest issue, I think, is then getting guacd to be able to add it to a FreeRDP connection. From my looking at the current FreeRDP source code (stable-1.1 branch), FreeRDP currently detects the timezone on the system where it's running, first using the TZ variable, then looking at /etc/localtime and/or /etc/TZ on the system. This means that it's always going to redirect the timezone of the system where guacd is running, and not the timezone of the browser. Unfortunately there don't seem to be any available functions for sending FreeRDP a string representation of a timezone and having it process it that way, and all of the timezone information is located in the FreeRDP .c file, making it difficult to write a method for handling this without duplicating a lot of code and all of the timezone tables. So, not entirely sure the best route on that angle at this point, but we'll see. Any input on whether this is a reasonable thing to address, or a corner-case not worth handling?

13. **summary:** Timezone Redirection

**description:** It would be nice to support full timezone redirection from the browser client through to a RDP server. There are a few challenges to making this work: - Detect timezone at browser and pass that through to the servlet side, somewhere. - Handle timezone on servlet side and pass through to guacd - Allow guacd to process the timezone and add it to the RDP connection. For browser side, there is a JavaScript library, jstz, that can do the detection and provide the timezone. Handling it on the servlet side, however, requires some way to pass it in during the connection. This could be done as a parameter at login time, or a separate api call. Once it is into the servlet, there has to be a way to process it as a parameter to a connection. I'm thinking maybe a token, but not sure that's the right way to go. Once it's on the connection it can pass through to guacd. The biggest issue, I think, is then getting guacd to be able to add it to a FreeRDP connection. From my looking at the current FreeRDP source code (stable-1.1 branch), FreeRDP currently detects the timezone on the system where it's running, first using the TZ variable, then looking at /etc/localtime and/or /etc/TZ on the system. This means that it's always going to redirect the timezone of the system where guacd is running, and not the timezone of the browser. Unfortunately there don't seem to be any available functions for sending FreeRDP a string representation of a timezone and having it process it that way, and all of the timezone information is located in the FreeRDP .c file, making it difficult to write a method for handling this without duplicating a lot of code and all of the timezone tables. So, not entirely sure the best route on that angle at this point, but we'll see. Any input on whether this is a reasonable thing to address, or a corner-case not worth handling?

**jira\_issues\_comments:**

1. This sounds like a great idea, and likely something that should be part of the Guacamole protocol in general (part of the handshake). We'd need to find a standard representation for those timezones.
2. It seems like FreeRDP can handle the standard Region/Locale method that Guacamole uses. At least, there's a large translation table for it:  
<https://github.com/FreeRDP/FreeRDP/blob/03ab68318966c3a22935a02838daaea7b7f96c/libfreerdp/locale/timezone.c#L1113>  
For the other protocols, I'm not sure if there's any value at all in passing it through. I suppose for SSH you could try to inject the TZ environment variable into the connection at/after login, and, assuming the SSH server allows it, that would allow for "localization" of a SSH connection, as well.
3. {quote} (part of the handshake) {quote} So, are you thinking like a parameter, similar to the width, height, etc., that gets passed to the tunnel?connect endpoint?
4. **body:** Well, FreeRDP is being very challenging in this regard. It does not look like there's any method built-in to the API to feed the timezone to FreeRDP and have it use the already-provided translation tables for that timezone. It looks like the FreeRDP folks have assumed that the only place you'd \*ever\* want to get the timezone info from for a connection through FreeRDP is in one of the platform-specific ways they have determined. Also, their entire translation table for UNIX-to-Windows timezones is in the .c file, not the .h file, making it difficult to even use the translation table as part of an external program. There is a "winpr\_match\_unix\_timezone\_identifier\_with\_list" function that takes a UNIX timezone ID in as a string and matches it to a list, but you have to feed it the list, as well, so, again, the value is limited when the list isn't part of the header. So, am I just missing something here, or is supporting this in RDP going to be incredibly ugly?  
**label:** code-design
5. **body:** Yeah, it might be ugly. If there is no way to programmatically inform FreeRDP of the desired timezone, but it checks a "TZ" environment variable, we may be able to still achieve this by setting that variable when the initial connection is established. Each connection is isolated within its own process, and thus has its own environment.  
**label:** code-design
6. What about:  
<https://github.com/FreeRDP/FreeRDP/blob/851f0979d5b0b45a7595cab7ab1dd2d41d4a73d1/include/freerdp/settings.h#L875-L878> ?
7. **body:** {quote} So, are you thinking like a parameter, similar to the width, height, etc., that gets passed to the tunnel?connect endpoint? {quote} That would likely be how it ends up being implemented at the webapp level, yes. At an API and protocol level, this would mean changes to {{GuacamoleClientInformation}}, to the parts of libguac which handle the handshake, to the parts of {{ConfiguredGuacamoleSocket}} which handle the handshake, to the docs, etc. such that the client timezone can be forwarded in a standardized way, just as other client information is already forwarded. This isn't to say that we can't / shouldn't offer some parameter which allows the timezone to be explicitly overridden at the connection level (similar to how we provide width/height parameters for RDP), but I definitely don't think we should hard-code magic which relies on connection parameters having particular names. It would need to be made part of the Guacamole protocol, such that this particular aspect is standardized (rather than arbitrary like connection parameters).  
**label:** code-design

8. {quote} Yeah, it might be ugly. If there is no way to programmatically inform FreeRDP of the desired timezone, but it checks a "TZ" environment variable, we may be able to still achieve this by setting that variable when the initial connection is established. Each connection is isolated within its own process, and thus has its own environment. {quote} Yeah, I just submitted a PR, and that's the route I ended up going. It's how the SSH connections will work, too, so it isn't all bad, even if it does feel a little crude. {quote} What about:  
<https://github.com/FreeRDP/FreeRDP/blob/851f0979d5b0b45a7595cab7ab1dd2d41d4a73d1/include/freerdp/settings.h#L875-L878> {quote} Yes, you would think "aha, here's how I can do it." But, alas, no. The ClientTimeZone field is in some sort of special format, and there's a good bit of code that does the translation from UNIX time to that format:  
<https://github.com/FreeRDP/FreeRDP/blob/stable-1.1/libfreerdp/locale/timezone.c> Basically, that file has a translation table that it uses to look up the timezone from the various standard time formats (e.g America/Los\_Angeles) to the time zone format expected by RDP. And, unfortunately, the FreeRDP API does not provide any standard way to translate it - most of the functions used for it are not provided in header files that can be leveraged by anything external, particularly when it comes to the translation table, so using the functions there would, as far as I can tell, require completely rewriting the translation tables. You're welcome to take a look and tell me if I'm overlooking something, but I've traced around those function calls and can't seem to find a way to call any of them in such a way that would just let me feed it a standard timezone and have it spit out the RDP identifier on the other end.
9. {quote} That would likely be how it ends up being implemented at the webapp level, yes. ... This isn't to say that we can't / shouldn't offer some parameter which allows the timezone to be explicitly overridden at the connection level (similar to how we provide width/height parameters for RDP), but I definitely don't think we should hard-code magic which relies on connection parameters having particular names. {quote} Sounds good. To do my testing I just added a parameter to the RDP and SSH connections that allows for setting it, but I think ideally we want to be able to use a JavaScript library (JSTZ) to detect the timezone and have it sent over automatically. There are a few caveats - like, on Windows Server it only works when you've installed the Remote Desktop role, it doesn't work with the default RDP-based administration. On SSH, you have to enable the TZ parameter in the sshd\_config. So, it isn't something that's ever go to work out-of-the-box, but also isn't that hard to configure.
10. [~richard.lancaster]: I've opened a pull request that implements the empty {{nop}} instruction handler, which will log those at a {{TRACE}} level, so you should not see them show up in the log file unless you bump logging up to {{TRACE}}. However, as Mike points out the empty instructions should get filtered out by the tunnel implementation, so either you're using an old version of the tunnel (0.9.14 or earlier), or a different implementation?
11. {quote} So you should not see them show up in the log file unless you bump logging up to TRACE. {quote} Brilliant! Thanks. {quote} However, as Mike points out the empty instructions should get filtered out by the tunnel implementation, so either you're using an old version of the tunnel (0.9.14 or earlier), or a different implementation? {quote} You're right, we're not using the java server and have integrated guacamole with our device management platform.