
eG
Enabling Service Excellence

# Table of contents

# Table of Figures

1

# Introduction

Citrix Access Gateway™ products are universal SSL VPN appliances providing a secure, always-on, single point-of-access to an organization's applications and data. A comprehensive range of appliances and editions allow Access Gateway to meet the needs of any size organization, from small businesses to the most demanding global enterprises.

The Access Gateway appliance is deployed in an organization's demilitarized zone, and creates a virtual TCP connection with the client computer. Client computers launch the Citrix Secure Access Agent by simply accessing a secure Web URL or using the desktop icon. The Access Gateway then authenticates these credentials with a corporate authentication server and, if the credentials are correct, finishes the handshake with the client PC. Once authenticated, the Secure Access Agent is launched in the client computer, at which all network traffic destined for certain private networks is captured and redirected over the secure tunnel to the Access Gateway.

The error-free functioning of such an appliance is of tremendous significance in environments that span geographies and which support mission-critical applications handling highly sensitive information (like in the case of mobile/VoIP communication). Such environments often have to deal with concurrent access requests from remote users at disparate locations. With a defective Access Gateway, remote traffic could go unscanned and therefore unsecured, exposing the applications and resources to unauthorized usage, or worse, malicious virus attacks.

eG Enterprise offers out-of-the-box two specialized models for monitoring the Citrix Access Gateway – the *Citrix Access Gateway – Windows* model that focuses on the health of the Citrix Access Gateway operating on a Windows platform, and the *Citrix Access Gateway – Linux* model, which is a dedicated model for monitoring the Citrix Access Gateway component operating on Linux.

Using these models, administrators can constantly keep an eye on the operations of the Access Gateway and be proactively alerted of even minor non-conformances, so that the problem is resolved before non-genuine users gain access to critical applications and data.

2

# Administering the eG Manager to Work with the Citrix Access Gateway - Windows

To do this, do the following

1. Log into the eG administrative interface.
2. If a Citrix Acess Gateway - Windows is already discovered, then directly proceed towards managing it using the **COMPONENTS - MANAGE/UNMANAGE** page (Infrastructure -> Components -> Manage/Unmanage). However, if it is yet to be discovered, then run discovery (Infrastructure -> Components -> Discovery) to get it discovered or add the server manually using the **COMPONENTS** page (Infrastructure -> Components -> Add/Modify). Remember that components manually added are managed automatically. Discovered components, however, are managed using the **COMPONENTS - MANAGE/UNMANAGE** page. clearly illustrates the process of adding a Citrix Access Gateway - Windows.

   **Reference:**

   For more details on managing components, refer to *Configuring and Monitoring Web servers* document.

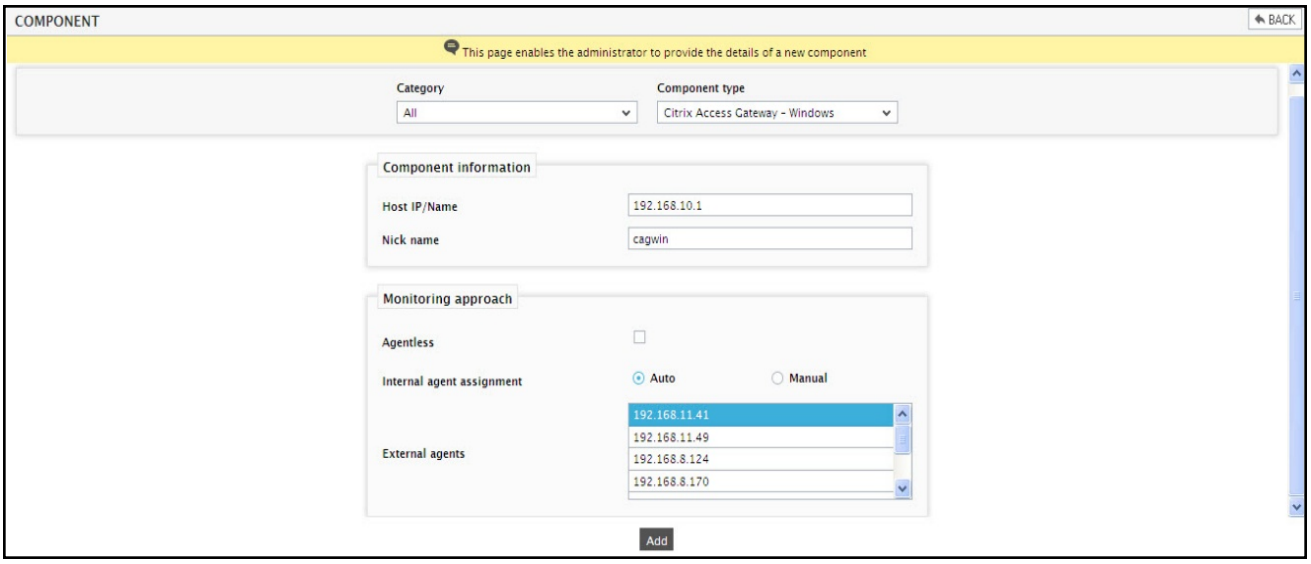


Figure 2.1: Adding a Citrix Access Gateway - Windows

3. Next, try to sign out of the eG administrative interface.
4. A list of unconfigured tests appears, listing the tests requiring configuration.

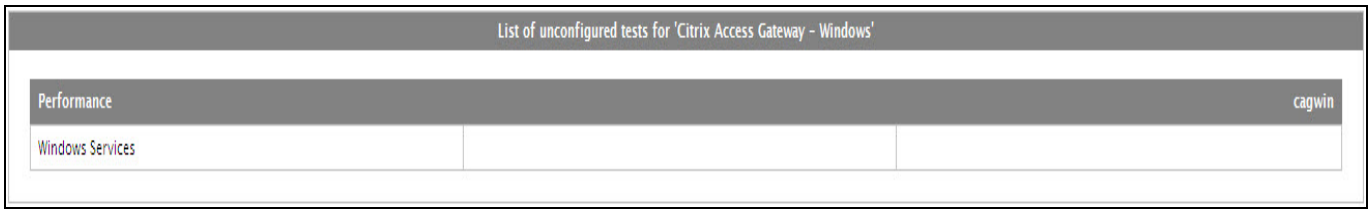


Figure 2.2: The list of unconfigured tests for the Citrix Access Gateway - Windows

5. Click on the **Windows Services** test in Figure 2.2 to configure it. You can configure critical Citrix Access Gateway-related services for monitoring by clicking on this test. Refer to the *Monitoring Unix and Windows Servers document* to learn how to configure the **Windows Services** test.

3

# Monitoring the Citrix Access Gateway on Windows

Figure 3.1 depicts the *Citrix Access Gateway – Windows* model.


Figure 3.1: Layer model of the Citrix Access Gateway

Every layer in the layer model of Figure 3.1 is attached to a wide variety of tests that explore one/more performance aspects of the Access Gateway. With the help of the results reported by these tests, the following performance queries can be easily answered; in the light of these answers, probable issues with the Access Gateway can be instantly detected.

- Is there a processing bottleneck on the Access Gateway?
- What are the type of requests that are being processed, and how quickly is the Access Gateway able to respond to them? Which requests are taking too long?
- Are the context pools adequately sized, or are too many requests waiting for contexts?
- Is the Access Gateway able to create/load sessions quickly upon request, or is there a bottleneck there that requires investigation?
- Is the session cache hit ratio optimal, or do more sessions need to be allocated to the cache?

The sections below discuss the top 3 layers of the layer model only, as the other layers have all been discussed thoroughly in the *Monitoring Unix and Windows Servers* document.

## 3.1 The .Net Layer

The **.Net** layer tracks the health of the ASP .Net framework on which the Access Gateway operates. Figure 3.2 reveals the tests mapped to this layer.

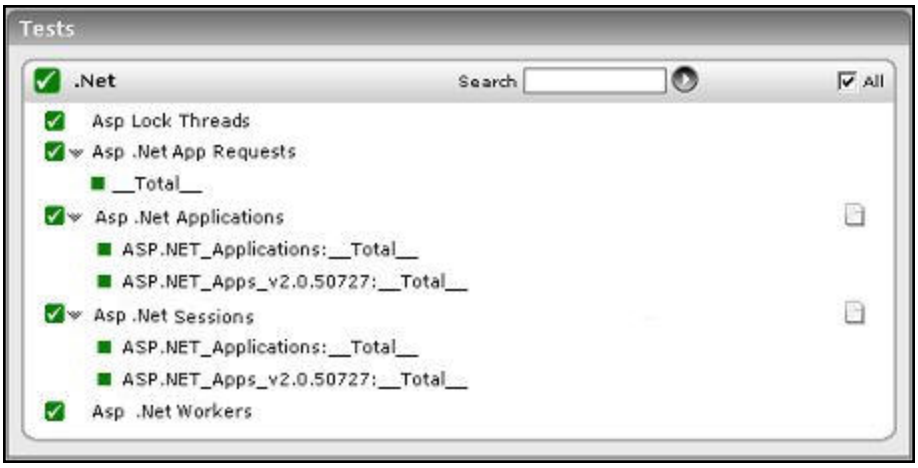


Figure 3.2: The tests mapped to the .Net layer

## 3.1.1 ASP Lock Threads Test

This test provides information about managed locks and threads that an application uses.

**Target of the test :** A Citrix Access Gateway

**Agent deploying the test :** An internal agent

**Outputs of the test :** One set of results for the Citrix Access Gateway being monitored

**Configurable parameters for the test**

1. **TEST PERIOD** - How often should the test be executed
2. **HOST** - The host for which the test is to be configured
3. **PORT** - The port at which the specified **HOST** listens

**Measurements made by the test**

| Measurement | Description | Measurement Unit | Interpretation |
|---|---|---|---|
| **Current logical threads:** | The number of current managed thread objects in the application. This measure maintains the count of both running and stopped threads. | Number | |
| **Current physical threads:** | The number of native operating system threads created and owned by the | Number | |

| Measurement | Description | Measurement Unit | Interpretation |
|---|---|---|---|
| | common language runtime to act as underlying threads for managed thread objects. This measure does not include the threads used by the runtime in its internal operations. | | |
| **Current recognized threads:** | The number of threads that are currently recognized by the runtime. These threads are associated with a corresponding managed thread object. | Number | |
| **Contention rate:** | The rate at which threads in the runtime attempt to acquire a managed lock unsuccessfully. | Rate/Sec | |
| **Current queue length:** | The total number of threads that are currently waiting to acquire a managed lock in the application. | Number | |

## 3.1.2 ASP .Net App Requests Test

This test monitors how well the application domain handles requests.

**Target of the test :** A Citrix Access Gateway

**Agent deploying the test :** An internal agent

**Outputs of the test :** One set of results for every application domain on the ASP .NET framework

**Test parameters**

1. **TEST PERIOD** - How often should the test be executed
2. **HOST** - The host for which the test is to be configured
3. **PORT** - The port at which the specified **HOST** listens

**Measurements made by the test**

| Measurement | Description | Measurement Unit | Interpretation |
|---|---|---|---|
| **Requests executing:** | The number of requests currently executing. | Number | This measure is incremented when the *HttpRuntime* begins to process the request and is decremented after the *HttpRuntime* finishes the request. |
| **Requests app queue:** | The number of requests currently in the application request queue. | Number | |
| **Requests not found:** | The number of requests that did not find the required resource. | Number | |
| **Requests not authorized:** | The number of request failed due to unauthorized access. | Number | Values greater than 0 indicate that proper authorization has not been provided, or invalid authors are trying to access a particular resource. |
| **Requests timed out:** | The number of requests timed out. | Number | |
| **Requests succeeded:** | The rate at which requests succeeded. | Requests/Sec | |

## 3.1.3 ASP .Net Applications Test

This test reports key statistics pertaining to applications deployed on the ASP .NET objects in the Citrix Access Gateway.

**Target of the test :** A Citrix Access Gateway

**Agent deploying the test :** An internal agent

**Outputs of the test :** One set of results for every ASP .NET object discovered in the Citrix Access Gateway

**Test parameters**

1. **TEST PERIOD** - How often should the test be executed
2. **HOST** - The host for which the test is to be configured
3. **PORT** - The port at which the specified **HOST** listens

**Measurements made by the test**

| Measurement | Description | Measurement Unit | Interpretation |
|---|---|---|---|
| **Request rate:** | Indicates the number of requests executed per second. | Number | This represents the current throughput of the application. |
| **Pipeline instances:** | Indicates the number of active pipeline instances for the ASP.NET application. | Number | Since only one execution thread can run within a pipeline instance, this number gives the maximum number of concurrent requests that are being processed for a given application. Ideally, the value of this measure should be low. |
| **Number of errors:** | Indicates the total sum of all errors that occur during the execution of HTTP requests. | Number | This measure should be kept at 0 or a very low value. |

## 3.1.4 ASP .Net Workers Test

This test reports statistics pertaining to the performance of the worker process of the ASP .NET framework of the Citrix Access Gateway.

**Target of the test :** A Citrix Access Gateway

**Agent deploying the test :** An internal agent

**Outputs of the test :** One set of results for Citrix Access Gateway monitored

**Configurable parameters for the test**

1. **TEST PERIOD** - How often should the test be executed
2. **HOST** - The host for which the test is to be configured
3. **PORT** - The port at which the specified **HOST** listens

**Measurements made by the test**

| Measurement | Description | Measurement Unit | Interpretation |
|---|---|---|---|
| **Application restarts:** | The number of application restarts. | Number | In a perfect world, the application domain will and should survive for the life of the |

| Measurement | Description | Measurement Unit | Interpretation |
|---|---|---|---|
| | | | process. Even if a single restart occurs, it is a cause for concern because proactive and reactive restarts cause automatic recycling of the worker process. Moreover, restarts warrant recreation of the application domain and recompilation of the pages, both of which consume a lot of time. To investigate the reasons for a restart, check the values set in the processModel configuration. |
| **Applications running:** | The number of applications currently running. | Number | |
| **Requests current:** | The number of requests currently handled by the ASP.NET ISAPI. This includes those that are queued , executing, or waiting to be written to the client. | Number | |
| **Request execution time:** | The number of seconds taken to execute the last request. | Number | In version 1.0 of the framework, the execution time begins when the worker process receives the request, and stop when the ASP.NET ISAPI sends HSE_REQ_DONE_WITH_SESSION to IIS. In version 1.1 of the framework, execution begins when the HttpContext for the request is created, and stop before the response is sent to IIS. The value of this measure should be stable. Any sudden change from the previous recorded values should be notified. |
| **Requests queued:** | The number of requests currently queued. | Number | When running on IIS 5.0, there is a queue between inetinfo and aspnet_wp, and there is one queue for each virtual |

| Measurement | Description | Measurement Unit | Interpretation |
|---|---|---|---|
| | | | directory. When running on IIS 6.0, there is a queue where requests are posted to the managed ThreadPool from native code, and a queue for each virtual directory. This counter includes requests in all queues. The queue between inetinfo and aspnet_wp is a named pipe through which the request is sent from one process to the other. The number of requests in this queue increases if there is a shortage of available I/O threads in the aspnet_wp process. On IIS 6.0 it increases when there are incoming requests and a shortage of worker threads. |
| **Requests rejected:** | The number of rejected requests | Number | Requests are rejected when one of the queue limits is exceeded. An excessive value of this measure hence indicates that the worker process is unable to process the requests due to overwhelming load or low memory in the processor. |
| **Requests wait time:** | The number of seconds that the most recent request spent waiting in the queue, or named pipe that exists between inetinfo and aspnet_wp. This does not include any time spent waiting in the application queues. | Secs | |
| **Worker processes running:** | The current number of aspnet_ wp worker processes | Number | Every application executing on the .NET server corresponds to a worker process. Sometimes, during active or proactive recycling, a new worker process and the worker process that is being replaced |

| Measurement | Description | Measurement Unit | Interpretation |
|---|---|---|---|
| | | | may coexist. Under such circumstances, a single application might have multiple worker processes executing for it. Therefore, if the value of this measure is not the same as that of Applications_ running, then it calls for closer examination of the reasons behind the occurrence. |
| **Worker process restarts:** | The number of aspnet_wp process restarts in the machine | Number | Process restarts are expensive and undesirable. The values of this metric are dependent upon the process model configuration settings, as well as unforeseen access violations, memory leaks, and deadlocks. |

## 3.1.5 ASP .Net Sessions Test

This test monitors the application sessions to the ASP .NET framework of the Citrix Access Gateway.

**Target of the test :** A Citrix Access Gateway

**Agent deploying the test :** An internal agent

**Outputs of the test :** One set of results for every application session to the ASP .NET framework

**Configurable parameters for the test**

1. **TEST PERIOD** - How often should the test be executed
2. **HOST** - The host for which the test is to be configured
3. **PORT** - The port at which the specified **HOST** listens

**Measurements made by the test**

| Measurement | Description | Measurement Unit | Interpretation |
|---|---|---|---|
| **SQL connections:** | Indicates the number of connections to the SQL Server used by session state. | Number | An unusually high value may indicate a sudden increase in sessions to the SQL Server. |

| Measurement | Description | Measurement Unit | Interpretation |
|---|---|---|---|
| **State server connections:** | Indicates the number of connections to the StateServer used by session state. | Number | An unusually high value may indicate a sudden increase in sessions to the StateServer. |
| **Abandoned ASPNet application sessions:** | Indicates the number of sessions that have been explicitly abandoned during the last measurement period. | Number | |
| **Active ASPNet application sessions:** | Indicates the currently active sessions. | Number | |
| **Timedout ASPNet application sessions:** | Indicates the number of sessions that timed out during the last measurement period. | Number | |
| **ASPNet application sessions:** | Indicates the total number of sessions during the last measurement period. | Number | |

## 3.2 The Web Server Layer

To track the availability, responsiveness, and overall health of the web server component of the Citrix Access Gateway, use the tests associated with this layer.


Figure 3.3: The tests associated with the Web Server layer

Since these tests have already been discussed in the *Monitoring Web Servers* document, let us straight away proceed to the **CAG Service** layer.

## 3.3 The CAG Service Layer

This layer continuously monitors the requests to the CAG, so as to proactively detect processing bottlenecks (if any), and keeps a check on any unsusual session behavior or session cache usage.


Figure 3.4: The tests associated with the CAG Service layer

### 3.3.1 CAG Sessions Test

This test monitors the sessions to the Citrix Access Gateway, exposes delays or other abnormalities in session creation/validation/loading, and stark inefficiencies (if any) in session cache utilization.

**Target of the test :** A Citrix Access Gateway

**Agent deploying the test :** An internal agent

**Outputs of the test :** One set of results for every Citrix Access Gateway being monitored

**Configurable parameters for the test**

1. **TEST PERIOD** – How often should the test be executed
2. **HOST** – The host for which the test is to be configured
3. **PORT** – Refers to the port used by the Citrix Access Gateway

**Measurements made by the test**

| Measurement | Description | Measurement Unit | Interpretation |
|---|---|---|---|
| **CAG sessions started:** | Indicates the rate at which sessions were created on | Creates/Sec | |

| Measurement | Description | Measurement Unit | Interpretation |
|---|---|---|---|
| | the Citrix Access Gateway. | | |
| **CAG sessions updated:** | Indicates the rate at which the sessions were updated during the last measurement period. | Updates/Sec | |
| **CAG sessions validated:** | Indicates the rate at which sessions were validated during the last measurement period. | Validates/Sec | |
| **CAG sessions loaded:** | Indicates the rate at which sessions were loaded during the last measurement period. | Updates/Sec | |
| **CAG sessions saved:** | Indicates the rate at which sessions were saved during the last measurement period. | Saves/Sec | |
| | Indicates the rate at which sessions were deleted during the last measurement period. | Deletes/Sec | |
| **CAG session cache hits:** | Indicates the rate at which session requests were serviced by the session-cache during the last measurement period. | Hits/Sec | Ideally, this value should be high. A low value indicates that session requests are often fulfilled by direct disk accesses, thus increasing the processing overheads. You might want to increase the session cache size, if the situation persists. |
| **CAG session cache misses:** | Indicates the rate at which the session-cache could not service session requests during the last measurement period. | Misses/Sec | Ideally, this value should be low. A high value indicates that session requests are often fulfilled by direct disk accesses, thus increasing the processing overheads. You might want |

| Measurement | Description | Measurement Unit | Interpretation |
|---|---|---|---|
| | | | to increase the session cache size, if the situation persists. |

4

# Administering the eG Manager to monitor a Citrix Access Gateway - Linux server

1. Log into the eG administrative interface.
2. eG Enterprise cannot automatically discover Citrix Access Gateway - Linux server. You need to manually add the server using the **COMPONENTS** page (see ) that appears when the *Infrastructure -> Components -> Add/Modify* menu sequence is followed. Remember that components manually added are managed automatically.

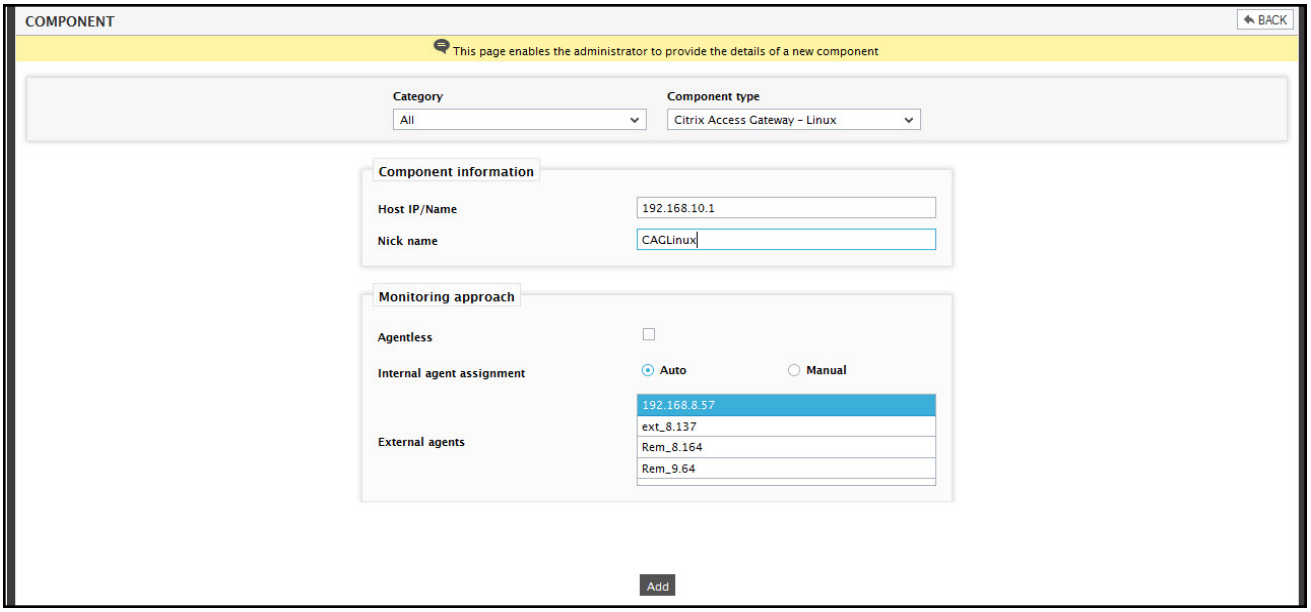


Figure 4.1: Adding the Citrix Access Gateway – Linux server

3. When you attempt to sign out, a list of unconfigured tests appears.

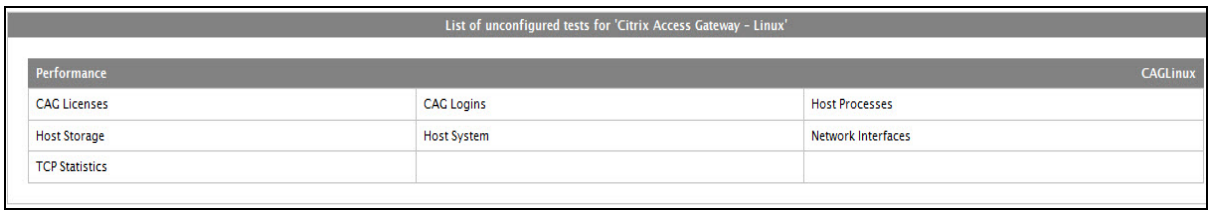


Figure 4.2: List of tests to be configured for Citrix Access Gateway – Linux server

4. Click on the **CAG Licenses** test to configure it. To know how to configure the test, Click here.

5

# Monitoring the Citrix Access Gateway on Linux

Figure 5.1 depicts the *Citrix Access Gateway - Linux* monitoring model that eG Enterprise offers.

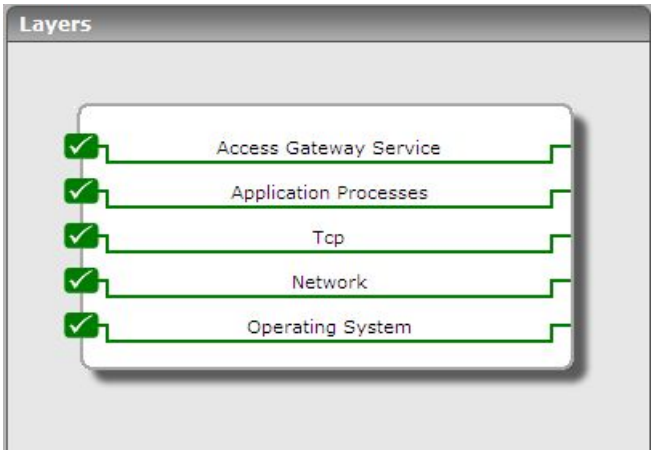


Figure 5.1: The layer model of the Citrix Access Gateway on Linux

Each layer is mapped to tests that periodically poll the SNMP MIB of the Citrix Access Gateway to retrieve useful performance statistics. These statistics reveal the following:

a. Have any login attempts to the CAG failed?

b. Have any administrative login attempts failed?

c. Has the connection pool been utilized optimally or have too many connections been used already?

## 5.1 The Operating System Layer

Using the tests mapped to this layer, administrators can track the usage of every storage area of the CAG and instantly identify the areas that are running out of storage space. In addition, the layer also monitors the number of processes running on the CAG and the number of users currently connected to it.

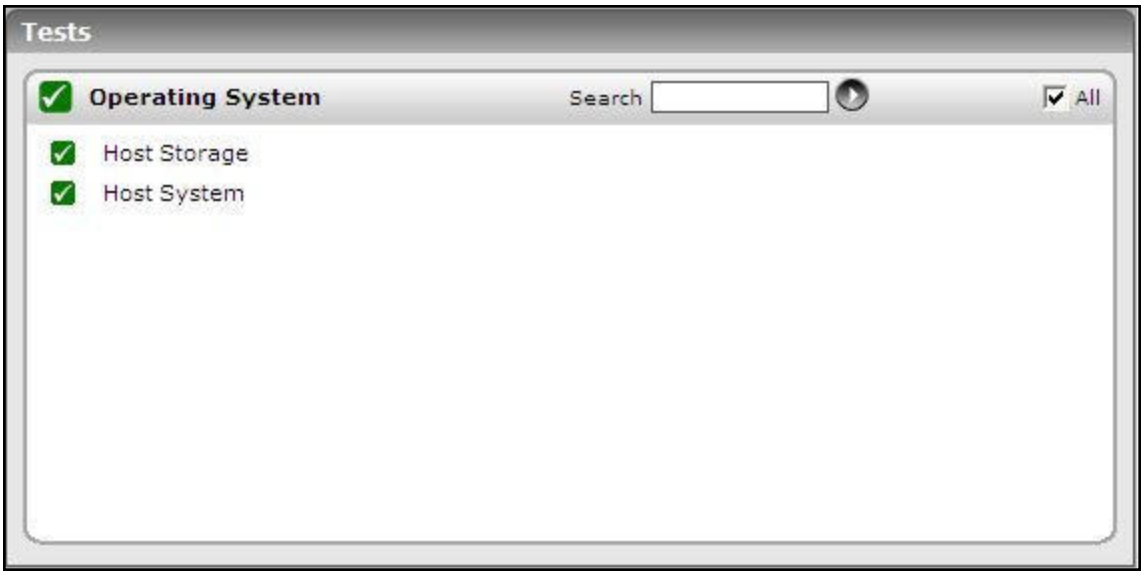


Figure 5.2: The tests mapped to the Operating System layer

## 5.1.1 Host Storage Test

This test auto-discovers all the storage areas of the CAG and tracks the usage of each of these areas.

**Target of the test :** CAG on Linux

**Agent deploying the test :** A remote agent

**Outputs of the test :** One set of results for every storage area on the server being monitored

**Configurable parameters for the test**

1. **TEST PERIOD** - How often should the test be executed
2. **HOST** - The host for which the test is to be configured
3. **SNMPPORT** - The port used to poll for SNMP statistics (default 161)
4. **SNMPVERSION** – By default, the eG agent supports SNMP version 1. Accordingly, the default selection in the **SNMPVERSION** list is **v1**. However, if a different SNMP framework is in use in your environment, say SNMP **v2** or **v3**, then select the corresponding option from this list.
5. **SNMPCOMMUNITY** – The SNMP community name that the test uses to communicate with the target host. This parameter is specific to SNMP **v1** and **v2** only. Therefore, if the **SNMPVERSION** chosen is **v3**, then this parameter will not appear.
6. **USERNAME** – This parameter appears only when **v3** is selected as the **SNMPVERSION**. SNMP version 3 (SNMPv3) is an extensible SNMP Framework which supplements the SNMPv2 Framework, by additionally supporting message security, access control, and remote SNMP configuration capabilities. To extract performance statistics from the MIB using the highly secure SNMP v3 protocol, the eG agent has to be configured with the required access privileges – in other words, the eG agent should connect to the MIB using the credentials of a user with access permissions to be MIB. Therefore,

specify the name of such a user against the **USERNAME** parameter.

7. **AUTHPASS** – Specify the password that corresponds to the above-mentioned **USERNAME**. This parameter once again appears only if the **SNMPVERSION** selected is **v3**.
8. **CONFIRM PASSWORD** – Confirm the **AUTHPASS** by retyping it here.
9. **AUTHTYPE** – This parameter too appears only if **v3** is selected as the **SNMPVERSION**. From the **AUTHTYPE** list box, choose the authentication algorithm using which SNMP v3 converts the specified **USERNAME** and **PASSWORD** into a 32-bit format to ensure security of SNMP transactions. You can choose between the following options:
   - **MD5** – Message Digest Algorithm
   - **SHA** – Secure Hash Algorithm
10. **ENCRYPTFLAG** – This flag appears only when **v3** is selected as the **SNMPVERSION**. By default, the eG agent does not encrypt SNMP requests. Accordingly, the **ENCRYPTFLAG** is set to **NO** by default. To ensure that SNMP requests sent by the eG agent are encrypted, select the **YES** option.
11. **ENCRYPTTYPE** – If the **ENCRYPTFLAG** is set to **YES**, then you will have to mention the encryption type by selecting an option from the **ENCRYPTTYPE** list. SNMP v3 supports the following encryption types:
   - **DES** – Data Encryption Standard
   - **AES** – Advanced Encryption Standard
12. **ENCRYPTPASSWORD** – Specify the encryption password here.
13. **CONFIRM PASSWORD** – Confirm the encryption password by retyping it here.
14. **TIMEOUT** – Specify the duration (in seconds) beyond which the SNMP query executed by this test should time out. The default is 10 seconds.

**Measurements made by the test**

| Measurement | Description | Measurement Unit | Interpretation |
|---|---|---|---|
| **Storage size:** | Represents the total size of a storage area associated with a server. | GB | |
| **Usage of storage area:** | This metric denotes the percentage capacity of a storage area that is currently allocated. | Percent | A value close to 100% denotes a storage area that is highly used. |
| **Free space on storage area:** | This metric denotes the amount of storage of a storage area that is currently available for use. | GB | |

| Measurement | Description | Measurement Unit | Interpretation |
|---|---|---|---|
| **Allocation failures on storage area:** | The number of requests for storage represented by this entity that could not be honored in the last measurement period because there was not enough storage available to service application requests | Number | Ideally, there should be no allocation failures. |

## 5.1.2 Host System Test

This test monitors the number of users accessing the CAG device and the processes executing on the device.

**Target of the test :** CAG on Linux

**Agent deploying the test :** A remote agent

**Outputs of the test :** One set of results for each server being monitored

**Configurable parameters for the test**

1. **TEST PERIOD** - How often should the test be executed
2. **HOST** - The host for which the test is to be configured
3. **SNMPPORT** - The port used to poll for SNMP statistics (default 161)
4. **SNMPVERSION** – By default, the eG agent supports SNMP version 1. Accordingly, the default selection in the **SNMPVERSION** list is **v1**. However, if a different SNMP framework is in use in your environment, say SNMP **v2** or **v3**, then select the corresponding option from this list.
5. **SNMPCOMMUNITY** – The SNMP community name that the test uses to communicate with the target host. This parameter is specific to SNMP **v1** and **v2** only. Therefore, if the **SNMPVERSION** chosen is **v3**, then this parameter will not appear.
6. **USERNAME** – This parameter appears only when **v3** is selected as the **SNMPVERSION**. SNMP version 3 (SNMPv3) is an extensible SNMP Framework which supplements the SNMPv2 Framework, by additionally supporting message security, access control, and remote SNMP configuration capabilities. To extract performance statistics from the MIB using the highly secure SNMP v3 protocol, the eG agent has to be configured with the required access privileges – in other words, the eG agent should connect to the MIB using the credentials of a user with access permissions to be MIB. Therefore, specify the name of such a user against the **USERNAME** parameter.
7. **AUTHPASS** – Specify the password that corresponds to the above-mentioned **USERNAME**. This

parameter once again appears only if the **SNMPVERSION** selected is **v3**.

8. **CONFIRM PASSWORD** – Confirm the **AUTHPASS** by retyping it here.
9. **AUTHTYPE** – This parameter too appears only if **v3** is selected as the **SNMPVERSION**. From the **AUTHTYPE** list box, choose the authentication algorithm using which SNMP v3 converts the specified **USERNAME** and **PASSWORD** into a 32-bit format to ensure security of SNMP transactions. You can choose between the following options:
   - **MD5** – Message Digest Algorithm
   - **SHA** – Secure Hash Algorithm
10. **ENCRYPTFLAG** – This flag appears only when **v3** is selected as the **SNMPVERSION**. By default, the eG agent does not encrypt SNMP requests. Accordingly, the **ENCRYPTFLAG** is set to **NO** by default. To ensure that SNMP requests sent by the eG agent are encrypted, select the **YES** option.
11. **ENCRYPTTYPE** – If the **ENCRYPTFLAG** is set to **YES**, then you will have to mention the encryption type by selecting an option from the **ENCRYPTTYPE** list. SNMP v3 supports the following encryption types:
   - **DES** – Data Encryption Standard
   - **AES** – Advanced Encryption Standard
12. **ENCRYPTPASSWORD** – Specify the encryption password here.
13. **CONFIRM PASSWORD** – Confirm the encryption password by retyping it here.
14. **TIMEOUT** – Specify the duration (in seconds) beyond which the SNMP query executed by this test should time out. The default is 10 seconds.

**Measurements made by the test**

| Measurement | Description | Measurement Unit | Interpretation |
|---|---|---|---|
| **Current users:** | The current number of users logged in to the server being monitored. | Number | |
| **Current processes:** | The current number of processes executing on the server being monitored. | Number | |

## 5.1.3 The Network Layer

Monitor the availability and responsiveness of the CAG over the network, and also measure the bandwidth usage of each network interface supported by the CAG, with the help of the tests mapped to this layer.

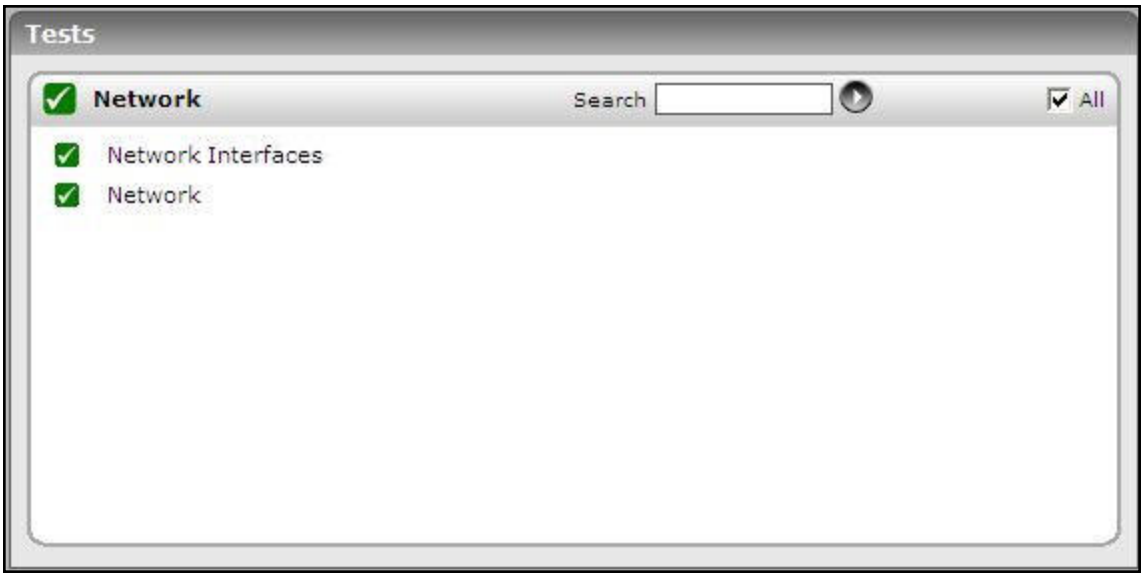


Figure 5.3: The tests mapped to the Network layer

Since these tests have already been discussed in the *Monitoring Unix and Windows Servers* document, let us proceed to the next layer.

## 5.2 The Tcp Layer

This layer measures the health of TCP connections to and from the CAG and also tracks TCP retransmissions.


Figure 5.4: The test mapped to the Tcp layer

As this test has been discussed elaborately in the *Monitoring Network elements* document, let us move to the next layer.

## 5.3 The Application Processes Layer

You can track the availability and resource usage of critical processes executing on the CAG using the test mapped to this layer.


Figure 5.5: The test mapped to the Application Processes layer

### 5.3.0.1 Host Processes Test

The **Host Processes** test monitors the specific processes executing on CAG and reports the resource usage of the processes.

**Target of the test :** CAG on Linux

**Agent deploying the test :** A remote agent

**Outputs of the test :** One set of results for every configured process pattern

**Configurable parameters for the test**

1. **TEST PERIOD** - How often should the test be executed
2. **HOST** - The host for which the test is to be configured
3. **SNMPPORT** - The port used to poll for SNMP statistics (default 161)
4. **SNMPVERSION** – By default, the eG agent supports SNMP version 1. Accordingly, the default selection in the **SNMPVERSION** list is **v1**. However, if a different SNMP framework is in use in your environment, say SNMP **v2** or **v3**, then select the corresponding option from this list.
5. **SNMPCOMMUNITY** – The SNMP community name that the test uses to communicate with the target server. This parameter is specific to SNMP **v1** and **v2** only. Therefore, if the **SNMPVERSION** chosen is **v3**, then this parameter will not appear.
6. **USERNAME** – This parameter appears only when **v3** is selected as the **SNMPVERSION**. SNMP version

3 (SNMPv3) is an extensible SNMP Framework which supplements the SNMPv2 Framework, by additionally supporting message security, access control, and remote SNMP configuration capabilities. To extract performance statistics from the MIB using the highly secure SNMP v3 protocol, the eG agent has to be configured with the required access privileges – in other words, the eG agent should connect to the MIB using the credentials of a user with access permissions to be MIB. Therefore, specify the name of such a user against the **USERNAME** parameter.

7. **AUTHPASS** – Specify the password that corresponds to the above-mentioned **USERNAME**. This parameter once again appears only if the **SNMPVERSION** selected is **v3**.
8. **CONFIRM PASSWORD** – Confirm the **AUTHPASS** by retyping it here.
9. **AUTHTYPE** – This parameter too appears only if **v3** is selected as the **SNMPVERSION**. From the **AUTHTYPE** list box, choose the authentication algorithm using which SNMP v3 converts the specified **USERNAME** and **PASSWORD** into a 32-bit format to ensure security of SNMP transactions. You can choose between the following options:
   - **MD5** – Message Digest Algorithm
   - **SHA** – Secure Hash Algorithm
10. **ENCRYPTFLAG** – This flag appears only when **v3** is selected as the **SNMPVERSION**. By default, the eG agent does not encrypt SNMP requests. Accordingly, the **ENCRYPTFLAG** is set to **NO** by default. To ensure that SNMP requests sent by the eG agent are encrypted, select the **YES** option.
11. **ENCRYPTTYPE** – If the **ENCRYPTFLAG** is set to **YES**, then you will have to mention the encryption type by selecting an option from the **ENCRYPTTYPE** list. SNMP v3 supports the following encryption types:
    - **DES** – Data Encryption Standard
    - **AES** – Advanced Encryption Standard
12. **ENCRYPTPASSWORD** – Specify the encryption password here.
13. **CONFIRM PASSWORD** – Confirm the encryption password by retyping it here.
14. **PROCESS** - Should contain the specific processes to be monitored. Each process to be monitored is specified in the format "name:pattern". The regular expression pattern denotes patterns that will be used to match processes on the server. For instance, to monitor all the Java processes on a server, specify the argument "java_processes:*java*".
15. **USEPROCESSPATH** - In some operating systems (example, OpenVMS), the process name in the HOST RESOURCES MIB will be an empty string, and the process path will include the process name. In such cases therefore, the test should be explicitly instructed to search the process path strings for the configured process names/patterns. To ensure this, set the **USEPROCESSPATH** parameter to **true**. By default, this parameter is set to **false**. Operating systems where process name (in the HOST RESOURCES MIB) is not an empty string can go with this default setting.
16. **TIMEOUT** – Specify the duration (in seconds) beyond which the SNMP query executed by this test should time out. The default is 10 seconds.

**Measurements made by the test**

| Measurement | Description | Measurement Unit | Interpretation |
|---|---|---|---|
| **Processes running:** | The number of processes currently executing on the server that match the pattern specified as parameter. | Number | This value indicates if too many or too few processes corresponding to an application are executing on the host. |
| **Memory utilization:** | The total memory usage of all processes executing on the server that match the pattern specified as parameter. The memory usage is specified as a percentage of the total memory available on the server. | Percent | A very high value could indicate that processes corresponding to the specified pattern are consuming excessive memory resources. |
| **Memory size:** | The total memory usage (in MB) of all processes executing on the server that match the pattern specified as parameter. | MB | A sudden increase in memory utilization for a process (es) may be indicative of memory leaks in the application. |
| **CPU utilization:** | The total CPU utilization of all processes executing on the server that match the configured process pattern. | Percent | A high value could signify a CPU bottleneck. The CPU utilization may be high because a few processes are consuming a lot of CPU, or because there are too many processes contending for a limited resource. Check the currently running processes to see the exact cause of the problem. |

## 5.4 The Access Gateway Service Layer

The tests mapped to this layer monitors the efficiency with which the CAG performs its core functions, which include:

- Login authentication
- Managing client connections


Figure 5.6: The tests mapped to the Access Gateway Service layer

## 5.4.1 CAG Licenses Test

This test monitors how well the CAG manages connections to the Citrix server.

**Target of the test :** CAG on Linux

**Agent deploying the test :** An internal/remote agent

**Outputs of the test :** One set of results for the CAG monitored

**Configurable parameters for the test**

1. **TEST PERIOD** - How often should the test be executed
2. **HOST** - The host for which the test is to be configured
3. **SNMPPORT** - The port used to poll for SNMP statistics (default 161)
4. **SNMPVERSION** – By default, the eG agent supports SNMP version 1. Accordingly, the default selection in the **SNMPVERSION** list is **v1**. However, if a different SNMP framework is in use in your environment, say SNMP **v2** or **v3**, then select the corresponding option from this list.
5. **SNMPCOMMUNITY** – The SNMP community name that the test uses to communicate with the target server. This parameter is specific to SNMP **v1** and **v2** only. Therefore, if the **SNMPVERSION** chosen is **v3**, then this parameter will not appear.
6. **USERNAME** – This parameter appears only when **v3** is selected as the **SNMPVERSION**. SNMP version 3 (SNMPv3) is an extensible SNMP Framework which supplements the SNMPv2 Framework, by additionally supporting message security, access control, and remote SNMP configuration capabilities. To extract performance statistics from the MIB using the highly secure SNMP v3 protocol, the eG agent has to be configured with the required access privileges – in other words, the eG agent should connect to the MIB using the credentials of a user with access permissions to be MIB. Therefore, specify the name of such a user against the **USERNAME** parameter.
7. **AUTHPASS** – Specify the password that corresponds to the above-mentioned **USERNAME**. This parameter once again appears only if the **SNMPVERSION** selected is **v3**.

8. **CONFIRM PASSWORD** – Confirm the **AUTHPASS** by retyping it here.
9. **AUTHTYPE** – This parameter too appears only if v3 is selected as the **SNMPVERSION**. From the **AUTHTYPE** list box, choose the authentication algorithm using which SNMP v3 converts the specified **USERNAME** and **PASSWORD** into a 32-bit format to ensure security of SNMP transactions. You can choose between the following options:
    - **MD5** – Message Digest Algorithm
    - **SHA** – Secure Hash Algorithm
10. **ENCRYPTFLAG** – This flag appears only when v3 is selected as the **SNMPVERSION**. By default, the eG agent does not encrypt SNMP requests. Accordingly, the **ENCRYPTFLAG** is set to **NO** by default. To ensure that SNMP requests sent by the eG agent are encrypted, select the **YES** option.
11. **ENCRYPTTYPE** – If the **ENCRYPTFLAG** is set to **YES**, then you will have to mention the encryption type by selecting an option from the **ENCRYPTTYPE** list. SNMP v3 supports the following encryption types:
    - **DES** – Data Encryption Standard
    - **AES** – Advanced Encryption Standard
12. **ENCRYPTPASSWORD** – Specify the encryption password here.
13. **CONFIRM PASSWORD** – Confirm the encryption password by retyping it here.
14. **TIMEOUT** – Specify the duration (in seconds) beyond which the SNMP query executed by this test should time out. The default is 10 seconds.

**Measurements made by the test**

| Measurement | Description | Measurement Unit | Interpretation |
|---|---|---|---|
| **Total licenses installed on the Access Gateway:** | Indicates the maximum number of client connections. | Number | |
| **Licenses in use:** | Indicates the number of connections currently used. | Number | |
| **Disabled licenses:** | Indicates the number of connections currently disabled. | Number | |
| **Licenses available for use:** | Indicates the number of connections currently unused. | Number | |
| **Available licenses percent:** | Indicates the percentage of | Percent | Ideally, this value should be high. A low |

| Measurement | Description | Measurement Unit | Interpretation |
|---|---|---|---|
| | unused connections. | | value indicates that too many connections are currently in use, and that the pool might not have enough connections to support subsequent connection requests. This can severely affect the user experience with the CAG. |

## 5.4.2 CAG LoginsTest

This test tracks the user logins to CAG, and captures failed login attempts.

**Target of the test :** CAG on Linux

**Agent deploying the test :** An internal/remote agent

**Outputs of the test :** One set of results for the CAG monitored

**Configurable parameters for the test**

1. **TEST PERIOD** - How often should the test be executed
2. **HOST** - The host for which the test is to be configured
3. **SNMPPORT** - The port used to poll for SNMP statistics (default 161)
4. **SNMPVERSION** – By default, the eG agent supports SNMP version 1. Accordingly, the default selection in the **SNMPVERSION** list is **v1**. However, if a different SNMP framework is in use in your environment, say SNMP **v2** or **v3**, then select the corresponding option from this list.
5. **SNMPCOMMUNITY** – The SNMP community name that the test uses to communicate with the target server. This parameter is specific to SNMP **v1** and **v2** only. Therefore, if the **SNMPVERSION** chosen is **v3**, then this parameter will not appear.
6. **USERNAME** – This parameter appears only when **v3** is selected as the **SNMPVERSION**. SNMP version 3 (SNMPv3) is an extensible SNMP Framework which supplements the SNMPv2 Framework, by additionally supporting message security, access control, and remote SNMP configuration capabilities. To extract performance statistics from the MIB using the highly secure SNMP v3 protocol, the eG agent has to be configured with the required access privileges – in other words, the eG agent should connect to the MIB using the credentials of a user with access permissions to be MIB. Therefore, specify the name of such a user against the **USERNAME** parameter.
7. **AUTHPASS** – Specify the password that corresponds to the above-mentioned **USERNAME**. This parameter once again appears only if the **SNMPVERSION** selected is **v3**.
8. **CONFIRM PASSWORD** – Confirm the **AUTHPASS** by retyping it here.

9. **AUTHTYPE** – This parameter too appears only if **v3** is selected as the **SNMPVERSION**. From the **AUTHTYPE** list box, choose the authentication algorithm using which SNMP v3 converts the specified **USERNAME** and **PASSWORD** into a 32-bit format to ensure security of SNMP transactions. You can choose between the following options:
   - **MD5** – Message Digest Algorithm
   - **SHA** – Secure Hash Algorithm
10. **ENCRYPTFLAG** – This flag appears only when **v3** is selected as the **SNMPVERSION**. By default, the eG agent does not encrypt SNMP requests. Accordingly, the **ENCRYPTFLAG** is set to **NO** by default. To ensure that SNMP requests sent by the eG agent are encrypted, select the **YES** option.
11. **ENCRYPTTYPE** – If the **ENCRYPTFLAG** is set to **YES**, then you will have to mention the encryption type by selecting an option from the **ENCRYPTTYPE** list. SNMP v3 supports the following encryption types:
   - **DES** – Data Encryption Standard
   - **AES** – Advanced Encryption Standard
12. **ENCRYPTPASSWORD** – Specify the encryption password here.
13. **CONFIRM PASSWORD** – Confirm the encryption password by retyping it here.
14. **TIMEOUT** – Specify the duration (in seconds) beyond which the SNMP query executed by this test should time out. The default is 10 seconds.

**Measurements made by the test**

| Measurement | Description | Measurement Unit | Interpretation |
|---|---|---|---|
| **Total logins:** | Indicates the number of logins during the last measurement period. | Number | |
| **Client user logins:** | Indicates the number of successful client logins to the CAG during the last measurement period. | Number | |
| **Failed logins:** | Indicates the number of client logins that failed during the last measurement period. | Number | Ideally, this value should be 0. |
| **Admin user logins:** | Indicates the number of successful admin user logins during the last measurement period. | Number | |

| Measurement | Description | Measurement Unit | Interpretation |
|---|---|---|---|
| **Failed admin user logins:** | Indicates the number of failed admin user logins during the last measurement period. | Percent | Ideally, this value should be 0. |

6

# Conclusion

This document has described in detail the monitoring paradigm used and the measurement capabilities of the eG Enterprise suite of products with respect to **Citrix Access Gateway**. For details of how to administer and use the eG Enterprise suite of products, refer to the user manuals.

We will be adding new measurement capabilities into the future versions of the eG Enterprise suite. If you can identify new capabilities that you would like us to incorporate in the eG Enterprise suite of products, please contact support@eginnovations.com. We look forward to your support and cooperation. Any feedback regarding this manual or any other aspects of the eG Enterprise suite can be forwarded to feedback@eginnovations.com.