

Network Working Group
Request for Comments: 3624
Category: Informational

B. Foster
D. Auerbach
F. Andreasen
Cisco Systems
November 2003

The Media Gateway Control Protocol (MGCP) Bulk Audit Package

Status of this Memo

This memo provides information for the Internet community. It does not specify an Internet standard of any kind. Distribution of this memo is unlimited.

Copyright Notice

Copyright (C) The Internet Society (2003). All Rights Reserved.

IESG Note

This document is being published for the information of the community. It describes a non-IETF protocol that is currently being deployed in a number of products. Implementers should be aware of RFC 3015, which was developed in the IETF Megaco Working Group and the ITU-T SG16, and which is considered by the IETF and the ITU-T to be the standards-based (including reviewed security considerations) way to meet the needs that MGCP was designed to address.

Abstract

The base Media Gateway Control Protocol (MGCP) includes audit commands that only allow a Call Agent to audit endpoint and/or connection state one endpoint at a time. This document describes a new MGCP package for bulk auditing of a group of gateway endpoints. It allows a Call Agent to determine the endpoint naming convention, the list of instantiated endpoints as well connection and endpoint state for the group of endpoints.

Table of Contents

- 1. Introduction. 2
- 2. Bulk Audit Package. 2
 - 2.1. Package Definition. 2
 - 2.1.1. Package Parameters 3
 - 2.1.2. Bulk Auditing of Non-persistent Virtual Endpoints. 11
 - 2.1.3. Package Specific Return Codes. 12
 - 2.2. Examples of Package Use 13
 - 2.2.1. Endpoint List. 13
 - 2.2.2. Connection Count List. 13
 - 2.2.3. Connection Mode List 15
 - 2.2.4. Endpoint State 15
- 3. IANA Considerations 17
- 4. Security Considerations 17
- 5. References. 18
- 6. Authors' Addresses. 18
- 7. Full Copyright Statement. 19

1. Introduction

The reader is assumed to be familiar with the base MGCP protocol [3].

The base Media Gateway Control Protocol (MGCP) [3] includes audit commands that only allow a Call Agent to audit an endpoint and/or a connection state, one endpoint at a time. This document describes a new MGCP package for bulk auditing of a group of gateway endpoints. It allows a Call Agent to determine the endpoint naming convention, to determine the list of instantiated endpoints, and to determine the connection and endpoint state for the group of endpoints. This is particularly important in fail-over situations in which there are gateways that have large numbers of endpoints.

Conventions Used in this Document

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [2].

2. Bulk Audit Package

2.1. Package Definition

Package Name: BA

Package Version: 0

Package Description: This package provides the Call Agent the ability to audit and obtain high-level view of endpoint and connection state for a group of endpoints in a gateway.

2.1.1.1. Package Parameters

A new BulkRequestedInfo parameter is defined for use in the AuditEndpoint command. The parameter can be used to request a compact list of EndpointIds or to request a high level view of endpoint or connection state for a group of endpoints as defined below:

```

ReturnCode,
[EndPointNameList,]
[InstantiatedEndpointList,]
[ConnectionCountList,]
[ConnectionModeList,]
[EndpointStateList,]
[NextEndpointName,]
[ReportedEndpointList]
<-- AuditEndPoint(EndpointId,
                  [StartEndpointName,]
                  [MaxNumEndpoints,]
                  [BulkRequestedInfo])

```

Unlike the normal RequestedInfo parameter in the base MGCP specification, the BulkRequestedInfo parameter associated with the Bulk Audits package can be used with "all-of" wildcards for auditing a collection of endpoints. However, it is not an error to specify an EndpointId without wildcards.

The following sub-sections describe the parameters associated with the Bulk Audit Command in detail. Sections 2.1.1.1 and 2.1.1.2 describe the parameters that can be included with a request and sections 2.1.1.3 to 2.1.1.8 describe return parameters.

2.1.1.1.1. StartEndpointName and MaxNumEndpoints Parameters

Because wild-carding may not be sufficient to qualify the endpoints of interest, further qualification can be provided by including a StartEndpointName (the first endpoint of interest) and MaxNumEndPoints (the maximum number of endpoints of interest). These parameters are described according to the following Augmented BNF (ABNF) Syntax (refer to RFC 2234 for ABNF syntax definitions [1]):

```

"BA/SE" ":" 0*WSP LocalEndpointName
"BA/NU" ":" 0*WSP MaxNumEndpoints

```

where MaxNumEndpoints is the decimal number of endpoints with a value in the range 1 to 65535. The MaxNumEndpoints parameter SHOULD only be included when requesting an audit for an EndpointStateList and/or ConnectionCountList. If included in a request for the EndPointNameList or InstantiatedEndpointList, it MAY be ignored.

Note that only the LocalEndpointName (see ABNF grammar in [3]) is provided in request and response parameter lines for this package rather than the full EndpointName. This is done for the sake of compactness, i.e., the domain name portion is left out since it is already available in the command line portion of a given request.

If the list of endpoints defined by the StartEndpointName and MaxNumEndPoints is outside the range designated by the wild-carding, a report will only be returned for endpoints up to those specified within the wild-card range.

2.1.1.2. BulkRequestedInfo Parameter

The BulkRequestedInfo parameter line is described according to the following ABNF syntax definitions:

```
BulkRequestedInfo = "BA/F:" 0*WSP
                    *( EndpointOrInstantList *(", " EndpointOrInstantList))
                    / *( EndpointOrConnState *(", " EndpointOrConnState))

EndpointOrConnState = "BA/C" / "BA/M" / EndpointStateParam

EndpointOrInstantList = "BA/Z" / "BA/X"

EndpointStateParam = "BA/S" "(" StateType
                    0*(", " 0*(WSP) StateType)"

StateType = "I" / "D" / "N" / "S" / "H"
```

where the BulkRequestedInfo parameters have the following meaning:

- * "BA/Z" is a request to return EndPointNameList
- * "BA/X" is a request to return InstantiatedEndpointList
- * "BA/C" is a request to return the ConnectionCountList
- * "BA/M" is a request to return the ConnectionModeList
- * "BA/S" is a request to return the EndpointStateList

Each of the parameters can be provided at most once in the BulkRequestedInfo.

EndpointStateParam Parameter:

As indicated in the above ABNF, the EndpointStateParam parameter is itself parameterized with one or more StateType parameters that define the conditions to be evaluated for the endpoint:

- * "I" - the endpoint is in-service,
- * "D" - the endpoint is disconnected (see sections 4.3 and 4.4.7 of [3] for a discussion on disconnected endpoints),
- * "N" - the endpoint is in the notification state,
- * "L" - the endpoint is in lockstep state (i.e., waiting for an RQNT after a response to a NTFY has occurred while in lockstep mode)
- * "S" - there is an active on-off (OO) or timeout (TO) signal on the endpoint,
- * "H" - the endpoint is in some state other than "idle". The meaning of this last parameter depends on the type of endpoint:
 - * The parameter has no meaning for endpoints that only provide bearer services (with no state that the endpoint is aware of). In this case, the condition is always evaluated to false (corresponding to "idle").
 - * For endpoints that have a state machine associated with them (such as a CAS endpoint), the endpoint MUST be in some state other than the "idle" state in order for the condition to be evaluated as true.
 - * In the case where the endpoint has hook-state associated with it, the hook-state MUST be off-hook. In the case of digital channel associated signaling (CAS) connections, hook-state may be provided in either direction. If the hook-state in either direction is off-hook, the endpoint is considered non-idle, i.e., the condition is satisfied.

The list of StateTypes may be extended in the future. If an unknown StateType is encountered, the command MUST be rejected with error code 803 (i.e., "unsupported StateType").

The report, provided as a result of this request, yields an indication of either "True", "False", or "Out of Service" for each endpoint. If the endpoint is in-service and any one of the criteria holds true, then the report for the endpoint will evaluate to "True". A "False" indication will only be reported if the endpoint is in-service and all criteria evaluate to false. The report thus provides the logical "OR" function over the conditions audited for endpoints in-service. Irrespective of the state being audited, an "Out of Service" indication will always be reported if the endpoint is considered out-of-service.

Note that the criteria "D", "N", "L", "S" and "H" can only be true if the endpoint is in-service, so that requesting "I" at the same time (although allowed) would be unnecessary (i.e., redundant).

Example: If the request for `EndpointStateList` for one or more endpoints includes the parameter line:

```
BA/F: BA/S(D,N)
```

indicating a request for a report on whether endpoints are disconnected or in the notification state. If a given endpoint is in either a "disconnected" or "notification" state, then the report will indicate "True" for that endpoint. If the endpoint is neither in a disconnected state nor in a notification state, but is in-service, then the report for that endpoint will indicate "False". If the endpoint is out-of-service, then the report for that endpoint will indicate "Out of Service".

In order to only determine whether an endpoint is in-service or out-of-service, the Call Agent should make a request with only the "I" `StateType` parameter.

2.1.1.3. `EndPointNameList` and `InstantiatedEndpointList` Parameters

`EndPointNameList` Parameter:

The `EndPointNameList` is a list of the endpoint names (i.e., the endpoint naming convention for the endpoints configured for service) supported by the gateway as qualified by the wildcarded `EndPointId`, and possibly `StartEndPointName` and `MaxNumEndpoints` parameters. This list can include one or more lines in the following ABNF format:

```
"BA/Z:" 0*WSP RangedLocalName 0*("," 0*WSP RangedLocalName)
```

where `RangedLocalName` is a `LocalEndpointName` that may include the ranged wildcard notation described in Appendix E (section E.5) of [3], i.e.,:

```
RangeWildcard = "[" NumericalRange *( "," NumericalRange ) "]"
NumericalRange = 1*(DIGIT) [ "-" 1*(DIGIT) ].
```

Example:

```
ba/z: ds/ds1-1/[1-24], ds/ds1-2/[1-24], ds/ds1-3/[1-24]
```

or simply:

```
ba/z: ds/ds1-[1-3]/[1-24]
```

Note that, since range wildcards use the character "[" to indicate the start of a range, the "[" character MUST NOT be used in endpoint names that use range wildcards.

Note that the ranged wildcard notation (RangeWildcard above) also allows commas between ranges like:

```
ba/z: ds/ds1-1/[1,3-5,8-24]
```

For virtual endpoints, that are automatically created and deleted on the fly by the gateway, there is a difference between reporting the endpoint names (i.e., the "naming convention") used in describing the endpoints and reporting the actual endpoints that are instantiated at the time the request is made. For this case:

- * EndPointNameList is a request to return the naming convention and
- * InstantiatedEndpointList is a request to return the "real" (or instantiated) endpoints.

InstantiatedEndpointList Parameter:

The syntax of the InstantiatedEndpointList value is the same as the EndPointNameList value returned with EndPointNameList, i.e., a number of lines may be returned with the following syntax:

```
"BA/X:" 0*WSP RangedLocalName 0*("," 0*WSP RangedLocalName)
```

In the case of hard-wired/physical endpoints (such as DSO's) or other persistent endpoints, the InstantiatedEndpointList would normally not be requested. However, if it is requested, the InstantiatedEndpointList and the EndPointNameList will be the same.

For virtual endpoints that are not persistent, an "all of" wild card ("*") is returned for the leftmost term of the name, which is dynamically assigned in the EndPointNameList to indicate that arbitrary names apply, and that the endpoints are virtual and non-persistent. The "all of" wild card notation MUST NOT be used when returning the EndPointNameList for persistent endpoints however. The following example illustrates this:

```
ba/z: announcement/*
ba/z: foo/bar/*
ba/z: foo/foo/*
```

The "all of" wildcard tells us, that "announcement" is simply the leftmost term for a dynamic set of non-persistent virtual endpoints. To instantiate one of these endpoints, we would include the "any of" wildcard (e.g., "announcement/\$") as the LocalEndpointName in the EndpointId of a request (e.g., NotificationRequest or CreateConnection). The response would then include the SpecificEndpointId indicating the instantiated endpoint. Also, note in the above example that "foo" defines two different levels of non-persistent virtual endpoints.

2.1.1.4. ConnectionCountList

The ConnectionCountList indicates the number of connections on a series of endpoints. It consists of a number of lines with the following ABNF syntax:

```
"BA/C:" 0*WSP NumConnections 0*(NumConnections)
```

where NumConnections is either:

- * a hexadecimal digit indicating the number of connections on the endpoint corresponding to the position on the list, or
- * the letter "Z" indicating that there are more than 15 connections on this endpoint.

2.1.1.5. ConnectionModeList

The ConnectionModeList indicates the connection modes for all the connections on a series of endpoints. It consists of a number of lines with the following ABNF syntax:

```
"BA/M:" 0*WSP ModeOrCount 0*(ModOrCount)
```

```
ModeOrCount = ConnCount / ConnMode
```

```
ConnMode = "I" / "S" / "R" / "B" / "C" / "L" / "T" / "N" / "U"
```

where ConnCount is either hexadecimal value corresponding to 0-15 connections on an endpoint or the value "Z", indicating that more than 15 connections are present.

ConnMode indicates the connection mode where:

- * "I" indicates "inactive" connection mode
- * "S" indicates "sendonly" connection mode
- * "R" indicates "recvonly" connection mode
- * "B" indicates "sendrecv" connection mode
- * "C" indicates "confrnce" connection mode
- * "L" indicates "loopback" connection mode
- * "T" indicates "conttest" connection mode
- * "N" indicates "netwloop" connection mode
- * "U" indicates some other connection mode

For a definition of MGCP connection modes, refer to section 3.2.2.6 of [3].

If an endpoint has no connections on it, ModeOrCount is given the value "0". If there is one connection associated with the endpoint, the symbol for the connection mode (ConnMode) is provided. If, on the other hand, there are from 2 to 15 connections, a symbol representing the number of connections (ConnCount) is provided followed by a list of symbols indicating the connection mode (ConnMode) for each connection. If there are more than 15 connections, "Z" is indicated for ConnCount and no connection modes are provided for the connections on that endpoint.

2.1.1.6. EndpointStateList Parameter

The EndpointStateList gives an overview of the endpoint state for a series of endpoints. It consists of a number of lines with the following ABNF syntax:

```
"BA/S:" 0*WSP EndPointState 0*(EndPointState)

EndPointState = "T" / "F" / "O"
```

where:

- * "T" indicates "True"
- * "F" indicates "False"
- * "O" indicates "Out of Service"

The "True" or "False" determination is based on the criteria supplied in StateType parameters when the request is made.

Note that the EndPointState indicator does not say anything about the connection state of the endpoint.

2.1.1.7. NextEndpointName Parameter

The NextEndpointName parameter will be included in the return, if there are additional endpoints in this gateway covered by the wildcarded endpoint name that were not reported, but for which information was available to be reported.

Note that the NextEndpointName is the LocalEndpointName (as opposed to EndpointName) of the next endpoint after the last endpoint reported. The syntax is as follows:

```
"BA/NE" ":" 0*WSP LocalEndpointName
```

A gateway may supply a report that is shorter than the request if the resulting report would have resulted in a message that would be too large (i.e., such that the report is larger than the maximum datagram size). In the case where the gateway supplied a response for less endpoints than requested, the gateway MUST supply NextEndpointName in the response.

In order to continue the audit on a following set of endpoints, the Call Agent can make a further request by using the NextEndpointName as the starting point (e.g., as the StartEndpointName in a following request).

2.1.1.8. ReportedEndpointList Parameter

A ReportedEndpointList MUST be provided in a response line before list(s) of EndpointStateList and/or ConnectionCountList in order to clearly specify the list of endpoints that are being reported. The ABNF syntax is as follows:

```
"BA/EL:" 0*WSP LimitedRangedName 0*("," 0*WSP LimitRangedName)
```

where LimitedRangedName is a LocalEndpointName that may include a ranged wildcard notation (RangeWildcard syntax indicated earlier). However, unlike the RangedLocalName that allows the range wildcard notation to be used on multiple terms of the local name at the same time, LimitedRangedName only allows the range notation to be used for the last term, i.e., the following is valid:

```
ba/el: ds/ds1-1/[1,3-5,8-24]
```

or

```
ba/el: ds/ds1-1/[1-24], ds/ds1-2/[1-24], ds/ds1-3/[1-24]
```

However, the following is not valid:

```
ba/el: ds/ds1-[1-3]/[1-24]
```

Note that a single bulk audit request may include a request to return both ConnectionCountList and EndpointStateList. However, the resulting report that includes both MUST cover the same endpoints.

A single bulk audit request may also include a request to return both EndPointNameList and InstantiatedEndpointList. However, requests for either an EndPointNameList and/or an InstantiatedEndpointList MUST NOT include a request for either ConnectionCountList or EndpointStateList.

2.1.2. Bulk Auditing of Non-persistent Virtual Endpoints

Note that gateways that have non-persistent virtual endpoints may have instantiated endpoints that are disjoint with respect to the name space. The ReportedEndpointList in front of a ConnectionCountList and/or EndpointStateList describes exactly which endpoints are being reported.

Example:

A Call Agent requests to know about the EndPointNameList for the endpoints on a conference bridge:

```
AUEP 1200 *@gw1.x.net MGCP 1.0
BA/F: BA/Z
```

Response:

```
200 1200 OK
ba/z: cnf/*
```

This indicates the naming convention but in fact not all of these endpoints are instantiated. A request for the list of instantiated endpoints, i.e.,:

```
AUEP 1201 cnf/*@gw1.x.net MGCP 1.0
BA/F: BA/X
```

might yield:

```
200 1201 OK
ba/x: cnf/[1-3]
ba/x: cnf/[6-12]
```

indicating that only these particular endpoints are instantiated.

Suppose the Call Agent now asks for the ConnectionCountList i.e.,:

```
AUEP 1202 cnf/*@gw1.x.net MGCP 1.0
BA/F: BA/C
```

The resulting instantiated virtual endpoints may be disjoint, which would be indicated by the ReportedEndpointList in front of the ConnectionCountList, e.g.,:

```
200 1202 OK
ba/el: cnf/[1-3]
ba/c: 035
ba/el: cnf/[6-12]
ba/c: 3450333
```

or alternatively:

```
200 1202 OK
ba/el: cnf/[1-3], ba/el: cnf/[6-12]
ba/c: 035
ba/c: 3450333
```

or

```
200 1202 OK
ba/el: cnf/[1-3], ba/el: cnf/[6-12]
ba/c: 0353450333
```

2.1.3. Package Specific Return Codes

The following return codes are specific to this package:

800	Invalid NextEndpointName
801	Invalid StartEndpointName
802	Invalid or unsupported BulkRequestInfo Parameter
803	Invalid or unsupported StateType
804	Bulk Audit Type not supported
805	Incorrectly specified endpoint range
806	Requested StartEndpoint unknown or unavailable

Note that package specific error codes includes the package name following the error code. For example, if error code 801 occurs in response to a request with a transaction ID of 1001 it would be sent as:

```
801 1001 /BA
```

2.2. Examples of Package Use

2.2.1. Endpoint List

This section contains examples of how to obtain a list of endpoints.

Example 1: This is an example of a gateway that contains a single OC3 that contains a single level of hierarchy at the T1 level.

The request is made:

```
AUEP 1200 *@gw1.x.net MGCP 1.0
BA/F: BA/Z
```

This may result in a single "BA/Z" term with ranges specifying all of the endpoints.

```
200 1200 OK
ba/z: ds/ds1-[1-84]/[1-24]
```

Example 2: In this example the gateway has 10 analog lines and a single T1. The same request is made as in example 1, but now the response is:

```
200 1200 OK
ba/z: aaln/[1-10]
ba/z: ds/ds1-1/[1-24]
```

2.2.2. Connection Count List

Example1: Audit the number of connections on endpoints of a single E1:

```
AUEP 2111 ds/e1-3/*@gw1.net
BA/F: BA/C
```

Response:

```
200 2111 OK
BA/EL: ds/e1-3/[1-30]
BA/C: 0121112100010000010000010000010
```

Example 2: Audit the number of connections on endpoints of a DS3:

```
AUEP 1144 ds/ds3-1/*@gateway.net
BA/F: BA/C
```

Response:

```
200 1144 OK
BA/EL: ds/ds3-1/[1-192]
BA/C: 010000010001000001000001
BA/C: 001000000101000000001001
:
BA/C: 011000100010000010000010
BA/C: 011111010001000001000001
BA/C: 011000001100000001000001
BA/NE: ds/ds3-1/193
```

In this case, the response provided by the gateway contained information about the first 192 endpoints. If the ds-3 contained a T1 hierarchy, the "BA/EL" and "BA/NE" values would indicate that hierarchy e.g.,:

```
200 1144 OK
BA/EL: ds/ds3-1/ds1-1/[1-24]
BA/C: 010000010001000001000001
BA/EL: ds/ds3-1/ds1-2/[1-24]
BA/C: 001000000101000000001001
:
BA/EL: ds/ds3-1/ds1-6/[1-24]
BA/C: 011000100010000010000010
BA/EL: ds/ds3-1/ds1-7/[1-24]
BA/C: 011111010001000001000001
BA/EL: ds/ds3-1/ds1-8/[1-24]
BA/C: 011000001100000001000001
BA/NE: ds/ds3-1/ds1-9/1
```

The Call Agent could continue to request endpoints by indicating the starting endpoint where it left off, i.e., simply using the returned "BA/NE" value as the "BA/SE" value for the next request:

```
AUEP 1145 ds/ds3-3/*@gw1.net
BA/F: BA/C
BA/SE: ds/ds3-1/ds1-9/1
```

Example 3: In this case, the Call Agent wants to know about the connection state of 12 DS0's starting with the endpoint with the LocalEndpointName "ds/ds3-1/ds1-6/4":

```
AUEP 1146 ds/ds3-1/*@gw1.net
BA/F: BA/C
BA/SE: ds/ds3-1/ds1-6/4
BA/NU: 12
```

Response:

```
200 1144 OK
BA/EL: ds/ds3-1/ds1-6/[4-15]
BA/C: 011000010001
BA/NE: ds/ds3-1/ds1-6/16
```

2.2.3. Connection Mode List

Example: Audit the connection modes for connections on the endpoints of a single E1:

```
AUEP 2111 ds/e1-3/*@gw1.net
BA/F: BA/M
```

Response:

```
200 2111 OK
BA/EL: ds/e1-3/[1-30]
BA/M: 0R2BRBBB2RRB0000B000000B000000B00000B0
```

This shows that:

- * Endpoint ds/e1-3/1 has no connections
- * Endpoint ds/e1-3/2 has one connection and it is in "recvonly" mode.
- * Endpoint ds/e1-3/3 has two connections which are in "sendrecv" and "recvonly" mode
- * Endpoints ds/e1-3/4 to ds/e1-3/6 each have one connection - in "sendrecv" mode in all cases
- * Endpoints ds/e1-3/7 has two connections, both in "recvonly" mode
- * etc.

2.2.4. Endpoint State

Endpoint state requests and responses are similar. An example of requesting endpoint state similar to example 3 in the previous section:

```
AUEP 1150 ds/ds3-1/*@gw1.net
BA/F: BA/S(I)
BA/SE: ds/ds3-1/ds1-6/4
BA/NU: 12
```

Response:

```
200 1150 OK
BA/EL: ds/ds3-1/ds1-6/[4-15]
BA/S: TOOTTOOTTOOT
BA/NE: ds/ds3-1/ds1-6/16
```

The request for in-service endpoints returns "True" for all endpoints in-service, and "0" for all endpoints "Out of Service".

A similar request but with additional parameters might be:

```
AUEP 1151 ds/ds3-1/*@gw1.net
BA/F: BA/S(H,N)
BA/SE: ds/ds3-1/ds1-6/4
BA/NU: 12
```

Response:

```
200 1151 OK
BA/EL: ds/ds3-1/ds1-6/[4-15]
BA/S: FFFFTFFFFFFFFFO
BA/NE: ds/ds3-1/ds1-6/16
```

This indicates that at least one of the StateType parameters "H" (off-hook) and "N" (notification state) evaluated to true for the endpoints that have a "T" associated with them (i.e., ds/ds3-1/ds1-6/7 and ds/ds3-1/ds1-6/16 since the request started from ds/ds3-1/ds1-6/4). All other endpoints are neither off-hook nor in the "notification state". Note that endpoint ds/ds3-1/ds1-6/15 is marked as being out-of-service.

It is possible to request both connection state and endpoint state in the same request such as:

```
AUEP 1151 ds/ds3-1/*@gw1.net
BA/F: BA/S(H,N), BA/C
BA/SE: ds/ds3-1/ds1-6/4
BA/NU: 12
```


In this case, the response might be:

```
200 1151 OK
BA/EL: ds/ds3-1/ds1-6/[4-15]
BA/S:  FFFFFFFF0
BA/C:  011000010001
BA/NE: ds/ds3-1/ds1-6/16
```

3. IANA Considerations

The MGCP package title, "Bulk Audit", with the name, "BA", has been registered with IANA as indicated in Appendix C.1 in [3].

4. Security Considerations

Section 5 of the base MGCP specification [3] discusses security requirements for the base protocol, which apply equally to the package defined in this document. Use of a security Protocol such as IPsec [4, 5] that provides per message authentication and integrity services is required in order to ensure that requests and responses are obtained from authenticated sources and that messages have not been modified. Without such services, gateways and Call Agents are open to attacks.

For example, although audit requests from unauthorized sources will not modify media gateway state, the information provided could be used to locate idle endpoints, which could then lead to making unauthorized calls. Similarly, an attack that modifies a response to an audit returned to a Call Agent could lead to a denial of service attack in which a Call Agent that is provided misinformation as to endpoint state could take some incorrect action such as taking valid calls out of service.

5. References

- [1] Crocker, D. and P. Overell, "Augmented BNF for Syntax Specifications: ABNF", RFC 2234, November 1997.
- [2] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [3] Andreasen, F. and B. Foster, "Media Gateway Control Protocol (MGCP) Version 1.0", RFC 3435, January 2003.
- [4] Kent, S. and R. Atkinson, "Security Architecture for the Internet Protocol", RFC 2401, November 1998.
- [5] Kent, S. and R. Atkinson, "IP Encapsulating Security Payload (ESP)", RFC 2406, November 1998.

6. Authors' Addresses

Flemming Andreasen
Cisco Systems
499 Thornall Street, 8th Floor
Edison, NJ 08837

EEmail: fandreas@cisco.com

David Auerbach
Cisco Systems Inc.
170 W. Tasman Drive
San Jose, CA, 95134

EEmail: dea@cisco.com

Bill Foster
Cisco Systems

EEmail: bfoster@cisco.com

7. Full Copyright Statement

Copyright (C) The Internet Society (2003). All Rights Reserved.

This document and translations of it may be copied and furnished to others, and derivative works that comment on or otherwise explain it or assist in its implementation may be prepared, copied, published and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and this paragraph are included on all such copies and derivative works. However, this document itself may not be modified in any way, such as by removing the copyright notice or references to the Internet Society or other Internet organizations, except as needed for the purpose of developing Internet standards in which case the procedures for copyrights defined in the Internet Standards process must be followed, or as required to translate it into languages other than English.

The limited permissions granted above are perpetual and will not be revoked by the Internet Society or its successors or assignees.

This document and the information contained herein is provided on an "AS IS" basis and THE INTERNET SOCIETY AND THE INTERNET ENGINEERING TASK FORCE DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Acknowledgement

Funding for the RFC Editor function is currently provided by the Internet Society.