

Configure the BW AS to Proxy Unknown Headers and Options

Contents

[Introduction](#)

[Prerequisites](#)

[Requirements](#)

[Components Used](#)

[Background Information](#)

[Configure](#)

[Verify](#)

[Troubleshoot](#)

Introduction

This document describes how to configure Proxy Policies on the BW AS, to proxy unknown SIP headers or Require/Supported parameters.

Prerequisites

- Basic Session Initiation Protocol (SIP) knowledge
- Basic Application Server (AS) knowledge
- Basic Broadworks (BW) bwcli knowledge

Requirements

Cisco recommends that you have knowledge of these topics:

- Be able to use the AS bwcli as an admin user
- Be able to review the AS XSLogs
- Generate an INVITE (via a client or simulator) that allows to configure Headers and Require/Supported parameters.

Components Used

The information in this document is based on these software and hardware versions:

- Cisco BW AS

The information in this document was created from the devices in a specific lab environment. All of the devices used in this document started with a cleared (default) configuration. If your network is live, ensure that you understand the potential impact of any command.

Background Information

As a Back-to-Back User Agent (B2BUA), the AS does not normally proxy SIP Headers or parameters in the Require and Supported headers, if they are considered unknown.

For example, if the AS receives this SIP INVITE, it discards the whole header MyUnknownHeader and the parameter MyUnknownParameter in the Supported header.

```
INVITE sip:+1555123456@cisco.com;user=phone SIP/2.0
Via: SIP/2.0/TCP 10.1.2.3:5060;branch=z9hG4bKva8in830cgv4i2mj6m20.1
Max-Forwards: 69
To: <sip:+1555123456@cisco.com;user=phone>
From: <sip:+1555654321@cisco.com;user=phone>;tag=SDq4k7b01-1
Contact: <sip:+155565432@172.16.0.1:5060;transport=tcp>
Call-ID: Testcall-456000001
CSeq: 101 INVITE
Supported: 100rel,MyUnknownParameter
MyUnknownHeader: MyStuff
Allow: INVITE, OPTIONS, BYE, CANCEL, ACK, PRACK, UPDATE, REFER, SUBSCRIBE, NOTIFY, INFO, REGISTER
Content-Length: 0
```

The ProxyPolicy settings on the AS allow you to configure unknown headers and parameters to be transparently proxied by the AS.

The [Cisco BroadWorks SIP Access Interface Interworking Guide](#) contains a list of the known headers and parameters, that can never be transparently proxied, as the AS processes them.

The known headers and parameters are processed and can or cannot be proxied (depending on the result of the AS processing), and cannot be controlled with the configuration explained in this article.

Configure

All the settings needed for the ProxyPolicy can be found in the bwcli at **AS_CLI/Interface/SIP/ProxyPolicy>**.

In order to allow the AS to proxy the header MyUnknownHeader, first navigate to **AS_CLI/Interface/SIP/ProxyPolicy/Rule>** from the AS bwcli and add a **new rule**:

```
<#root>
AS_CLI>
cd /Interface/SIP/ProxyPolicy/Rule

AS_CLI/Interface/SIP/ProxyPolicy/Rule>
add MyUnknownHeaderRule true true true true true true true true

...Done
```

Run the **get** command in order to show the current configuration.

The configuration now looks like this (due to the many options, the width of the output of this command is very large; horizontally scroll to fully view it):

```
<#root>
```

```
AS_CLI/Interface/SIP/ProxyPolicy/Rule>
```

```
get
```

```
Rule Name Keep On Redirection Keep On Egress Access Keep On Egress Network On-net Keep On Egress Network
=====
MyUnknownHeaderRule          true                               true                               true
```

Once this is completed, you must create the **HeaderPolicy**. Navigate to **AS_CLI/Interface/SIP/ProxyPolicy/HeaderPolicy>** and run this command:

```
<#root>
```

```
AS_CLI/Interface/SIP/ProxyPolicy/HeaderPolicy>
```

```
add MyUnknownHeader MyUnknownHeaderRule
```

```
...Done
```

In this example, the rule has been configured to preserve the header in all directions, for all services, and in redirect uri's, but this setting depends on your real life case.

The available options are explained in this list:

- **keepOnRedirection**: This parameter determines whether the header is kept in the case of a redirection.
- **keepOnEgressAccess**: This parameter determines whether the header is kept for outgoing messages to the access-side.
- **keepOnEgressNetworkOnNet**: This parameter determines whether the header is propagated for outgoing on-net messages bound to the network.
- **keepOnEgressNetworkOffNet**: This parameter determines whether the header is propagated for outgoing off-net messages bound to the network.
- **keepOnEgressAccessSharedCallAppearance**: This parameter determines whether the header is kept for outgoing messages to a Shared Call Appearance secondary location.
- **keepOnEgressNetworkBroadWorksAnywhereOnNet**: This parameter determines whether the header is kept for outgoing on-net messages to a BroadWorks Anywhere location.
- **keepOnEgressNetworkBroadWorksAnywhereOffNet**: This parameter determines whether the header is kept for outgoing off-net messages to a BroadWorks Anywhere location.
- **acceptFromRedirectToURI**: This parameter determines if an unknown header embedded in a redirect destination URI may be accepted for insertion into an outbound message.

The unknown parameter can be configured in the same way, but the parameter is configured in **AS_CLI/Interface/SIP/ProxyPolicy/OptionTagPolicy>** .

First, create a rule.



Note: For this example, the first switch is set to **false**.

```
<#root>
```

```
AS_CLI/Interface/SIP/ProxyPolicy/Rule>
```

```
add MyUnknownParameterRule false true true true true true true true
```

This is the resulting rule, with Keep On Redirection set to **false**, so the MyUnknownHeader is not added to the outgoing INVITE in case of redirections.

```
<#root>
```

```
AS_CLI/Interface/SIP/ProxyPolicy/Rule>
```

```
get
```

```
Rule Name Keep On Redirection Keep On Egress Access Keep On Egress Network On-net Keep On Egress Network
=====
```

MyUnknownHeaderRule	true	true	true
MyUnknownParameterRule	false	true	true

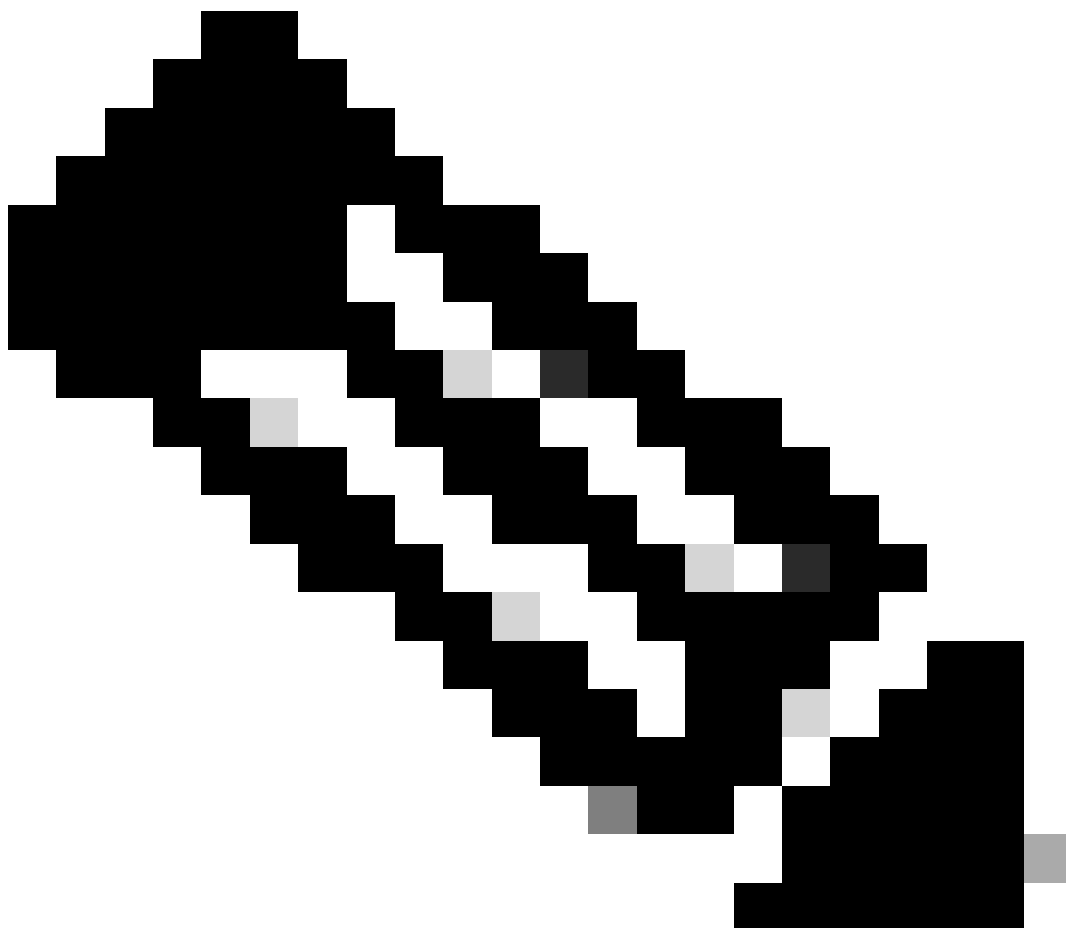
To complete the configuration, add the **OptionTagPolicy**.

```
<#root>
```

```
AS_CLI/Interface/SIP/ProxyPolicy/OptionTagPolicy>
```

```
add MyUnknownParameter MyUnknownParameterRule
```

The AS now proxies the unknown header MyUnknownHeader, and the unknown parameter MyUnknownParameter.



Note: The AS does not understand the semantic nor processes in any way the unknown headers and parameters.

Verify

Send to the AS the INVITE containing the unknown SIP header and option tag that you have configured in the Proxy Policy, and ensure that the outgoing INVITE has preserved the unknown header, according to the Rule configuration.

Troubleshoot

In case the outgoing INVITE does not contain the Header or Option Tag as expected, you can check this list:

- Ensure that the Header or Parameter you have configured is not included in the list of known headers/parameters.
- Check that the Header or Parameter in the INVITE exactly matches the ones configured in the bwcli (case insensitive).
- Check that the configured Rule allows the Header or Parameter to be proxied.