Available mechanisms to modify HTTP data within IBM ® Security Access Manager

—

ISAM Level II
Thursday, 5 September 2019

IBM Security

Announcing IBM VIP Rewards

Engage. Earn points. Get Rewards.

Learn more...
ibm.biz/vip-rewards

Join IBM VIP Rewards for Security...
ibm.biz/JoinIBMVIPRewards-Security

IBM

IBM VIP Rewards for **Security**

IBM Security

- Courses
- Videos
- Hands-on Labs
- Live Events
- Badges

Learning at no cost.

New content published daily.

IBM

Panel

Presenters

- Sten Drescher
- Steve Hughes
- Scott Stough

Panelist

- Nick Lloyd

Learn about the mechanisms available to modify HTTP data

This session will cover HTTP transformation updates, junction handling of credential attributes, and various configuration stanzas.

We see many Cases in Support with questions such as:

- Can we send a header to a junction server with xyz?
- Can I pull the value from a Cookie and send it as a header?
- Can I send error codes back to the client?
- How can credential attributes added via EAI be sent to the backend?

Agenda

- HTTP Transformation Rules (ISAM 9.0.6.0/9.0.7.0 Updates)
- HTTP-Tag-Value Extended Attributes
- Reverse Proxy Config File stanza
- Internet Content Adaptation Protocol (ICAP) Support

HTTP Transformation Rules (ISAM 9.0.6.0/9.0.7.0 Updates)

Existing assets

ISAM HTTP Transformation Open Mic from 2017 (Thanks Reagan)

More in-depth coverage of configuration and debugging.

ISAM Support's GitHub Examples

Many real-world examples derived from Support Cases. We encourage customers to add their own examples to the repository.

ISAM 9.0.6.0/9.0.7.0 Updates

9.0.6.0

Case-insensitive rules. Matching on resource URIs is case sensitive. To have a transformation do an insensitive match set:

```
[http-transformations:<resource-name>]
match-case-insensitive = true
```

ISAM 9.0.6.0/9.0.7.0 Updates

9.0.7.0

The XML representation of the HTTP Response object now makes available all of the elements of the HTTP Request object to be used in the response transformation.

HTTP Transformation Rules now support regular expressions which can be used to match and replace strings.

HTTP Transformation rules can now be used to specify the ACL bits which are used in the authorization decision for the request.

ISAM 9.0.6.0/9.0.7.0 Updates

Response object access to Request info.

Dynamic CORS Headers at GitHub (Thanks Jack)

```
<xsl:template match="//HTTPResponse/Headers">
    <xsl:variable name="originvalue" select="//HTTPResponse/HTTPRequest/Headers/Header[@name='origin']/node()" />
    <xsl:choose>
      <xsl:when test="Header/@name='access-control-allow-origin'"/>
      <xsl:otherwise>
        <Header action="add" name="access-control-allow-origin"><xsl:value-of select="$originvalue" /></Header>
      </xsl:otherwise>
    </xsl:choose>
</xsl:template>
```

ISAM 9.0.6.0/9.0.7.0 Updates

Regular expression match and find-and-replace. Quick example to change v1 to v2 if web services are being deprecated.

```
 <xsl:template match="//HTTPRequest/RequestLine">
    <xsl:choose>
      <xsl:when test="external:matches(URI, '^/dashboard/cgi-
bin/service/v1/.*')">
        <URI><xsl:value-of select="external:replace(URI, '/dashboard/cgi-
bin/service/v1/(.*)', '/dashboard/cgi-bin/service/v2/$1')"/></URI>
      </xsl:when>
    </xsl:choose>
  </xsl:template>
```

ISAM 9.0.6.0/9.0.7.0 Updates

Note: A brower’s address bar will still show v1:

```
/dashboard/cgi-bin/service/v1/date-and-time.sh
```

but we see from request.log the API called was v2:

```
192.168.61.1 - iuser 01/Sep/2019:14:08:22 -0500 "GET /dashboard/cgi-bin/service/v2/date-and-time.sh HTTP/1.1" 200 187
```

The response verifies as well:

```
{
  “version”:”2.0”,
  “full”:”Sun Sep  1 19:08:23 UTC 2019”
}
```

ISAM 9.0.6.0/9.0.7.0 Updates

ISAM ACL perms can be set for a new protected object for authorization.

Remember, this is not the URI but rather the protected object from the ISAM policy db to use for an authz decision.

```
<xsl:template match="/">
  <HTTPRequestChange>
    <xsl:apply-templates />  ← Set after the other templates have fired.  Note obj name, not URI.
            <ObjectName>/default-webseald-isam9070.level2.org/dashboard/new.html</ObjectName>
            <AclBits>r</AclBits>
  </HTTPRequestChange>
 </xsl:template>
```

HTTP-Tag-Value Extended Credential Attributes

HTTP-Tag-Value and Extended Credential Attributes

The HTTP-Tag-Value mechanism allows credential attributes to be inserted into HTTP request headers.

Commonly, these are extended credential attributes, added using the registry attribute entitlement service.

Registry Credential Attribute Service

The attributes to be added to the credential must be specified in the WebSEAL configuration file.

```
[TAM_CRED_ATTRS_SVC]
eperson = azn_cred_registry_id # this is a standard credential attribute
group = cn=enterprise, o=tivoli # this is the DN to be found in the user registry or federated directory


[TAM_CRED_ATTRS_SVC:eperson]
tagvalue_credattrs_lastname = sn
tagvalue_credattrs_employeetype = employeetype
tagvalue_credattrs_address = homepostaladdress
tagvalue_credattrs_email = mail


[TAM_CRED_ATTRS_SVC:group]
tagvalue_credattrs_businesscategory = businesscategory
```

Adding HTTP-Tag-Value attribute (1)

The HTTP-Tag-Value attribute must be added to the junction object in the object space.

```
pdadmin> object modify /WebSEAL/WS1/junctionA set attribute HTTP-Tag-Value credattrs_lastname=surname
pdadmin> object modify /WebSEAL/WS1/junctionA set attribute HTTP-Tag-Value credattrs_email=email
```

By default, ‘tagvalue_’ is prepended to the credential name, so the tagvalue_credattrs_lastname and tagvalue_credattrs_email attributes will be added to all requests passing through the /junctionA junction.

```
surname:Drescher
email:stend@tivoli.com
```

Adding HTTP-Tag-Value attribute (2)

The HTTP-Tag-Value attributes added to child objects will be ignored.

```
pdadmin> object modify /WebSEAL/WS1/junctionA/hr set attribute HTTP-Tag-Value

credattrs_employeetype=type
```

The tagvalue_credattrs_employeetype credential attribute will not be added to requests.

Adding extended credential attributes with EAI

An EAI may add extended attributes to the credential, using the am-eai-xattrs header (by default).

```
am-eai-xattrs: zipcode, tagvalue_always
zipcode: 78741
tagvalue_always: zipcode:zip
```

tagvalue_always is a special credential attribute, to specify attributes added to all forwarded requests.

Extended attributes added by EAI may also be added using the HTTP-Tag-Value attribute.

Reverse Proxy Config File Stanza

[acnt-mgt]

This stanza is used to set options for Reverse Proxy management operations. Specifically for this topic is the setting:

```
http-rsp-header = header-name:macro
```

This setting is global only. It cannot be set on a per-junction.

For example, set the following:

```
[acnt-mgt]
http-rsp-header = errorcode:ERROR_CODE
http-rsp-header = errortext:ERROR_TEXT
```

and when a management page is return, such as for a 403 Forbidden, the following headers are sent:

```
errorcode: 0x38cf0427
errortext: Forbidden
```

[acnt-mgt] (Macros)

TAM_OP

AUTHNLEVEL

ERROR_CODE

ERROR_TEXT

CREDATTR{*<name>*}, where *<name>* is the name of the credential attribute.

USERNAME

TEXT{*<value>*}, where *<value>* is the static header to include in the response header.

[acnt-mgt] (Default setttings)

```
http-rsp-header = x-frame-options:TEXT{DENY}
http-rsp-header = content-security-policy:TEXT{frame-ancestors 'none'}
http-rsp-header = x-content-type-options:TEXT{nosniff}
http-rsp-header = x-xss-protection:TEXT{1}
http-rsp-header = errorcode:ERROR_CODE
http-rsp-header = errortext:ERROR_TEXT
```

[filter-request-headers]

Use this stanza to configure HTTP headers for WebSEAL to filter before it sends a request to a junctioned server. The default setting is:

```
[filter-request-headers]
header = accept-encoding
```

This can be set per-junction:

```
[filter-request-headers:/junction]
header = someheader
```

The global will still apply to all junctions.

A common question is why is the default to filter accept-encoding? The reason is that we do not want compressed responses from the backend because the contents may need to be filtered depending on configuration.

[header-names]

Controls the addition of HTTP headers into the request that is passed to junctioned applications. The default setting is:

```
[header-names]
server-name = iv_server_name
```

This setting is global only.

A common use case we see is sending the hostname used in the original request in the

X-Forwarded-Host header to the backend:

```
httphdr{host} = X-Forwarded-Host
```

[header-names] (Macros)

The <header-data> may be one of the following values:

server-name: The ISAM authorization server name for the WebSEAL server. This is the name which is used in the "server task“ commands.

client-ip-v4: The IPv4 address of the client of this request.

client-ip-v6: The IPv6 address of the client of this request.

client-port: The port which is used by the client of this request.

host-name: The host name of the WebSEAL server. The host name will be obtained from the web-host-name configuration entry within the [server] stanza (if specified), or the host name of the machine itself.

httphdr{<name>}: A HTTP header from the request, as specified by the <name> field. If the HTTP header is not found in the request the value contained within the [server] tag-value-missing-attr-tag configuration entry will be used as the value for the header.

[rsp-header-names]

Defines static HTTP headers that are added to every HTTP response from the WebSEAL server.

The default setting is:

```
[rsp-header-name]
strict-transport-security = max-age=31536000; includeSubDomains
```

This can be set per-junction:

```
[rsp-header-names:/junction]
someheader: somevalue
```

Note:

These are not sent on management responses ([acnt-mgt]http-rsp-header)

If you do not want to stomp on a matching header from a backend set [server] replace-rsp-headers = no

[rsp-header-names] (Macros)

The %SESSION_EXPIRY% macro can be used to designate a header which will contain the remaining length of time in seconds before the current local session expires. This value does not include the overall session timeout for sessions which are managed by the distributed session cache (DSC), but just the length of time before the session expires in the local cache.

```
[rsp-header-name]
session-timeout = %SESSION_EXPIRY%
```

Note:

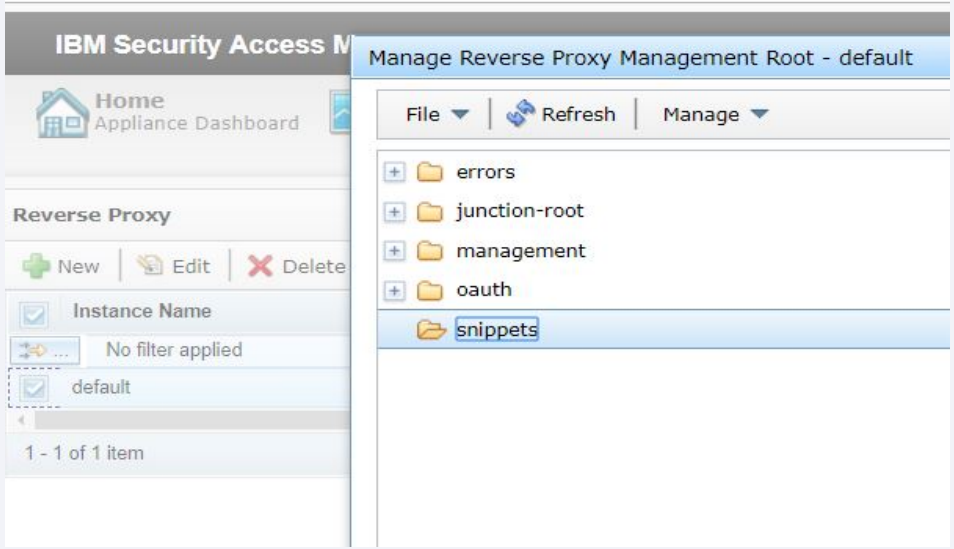
This does not work in the way that customers assume. It basically sends back in seconds the inactive-timeout whenever a resource is accessed. This is because as soon as the access occurred the timeout was reset. If you want to use it to throw up a session timer dialog then the application can query a resource set as below and now the header will have the actual seconds until inactive-timeout:

```
[session]
preserve-inactivity-timeout = /junction/timer-check.html     ← Standard Junction
preserve-inactivity-timeout = [www.vhj.org]/timer-check.html ← Virtual Host Junction
```

[snippet-filter]

This filter allows snippets to be inserted into the response for the specified URI. It can be used to insert a call to the AAC info.js page needed for device registration. Here are the the steps:

Navigate to the reverse proxy Management Root:



[snippet-filter]

Create a new snippet which is just the JavaScript for calling info.js

[snippet-filter]

Here are the contents of our main application dashboard page:

```
<head>
<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Level II Application Dashboard</title>
<link rel="stylesheet" type="text/css" href="index.css">
</head>
```

[snippet-filter]

Add the following to the reverse proxy config:

```
[snippet-filter:/dashboard/]
*Level*II*Application*Dashboard* = infojs

Where:
/dashboard/:                    The decoded URI for which the snippet substitution will take place.
*Level*II*Application*Dashboard*:   The location at which the snippet is inserted.
infojs:                         The name of the snippet file.
```

[snippet-filter]

Here are the contents of our main application dashboard page returned to the client:

```
<head>
<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<script src="https://isam9070-web.level2.org/mga/sps/ac/js/info.js"></script>
<title>Level II Application Dashboard</title>
<link rel="stylesheet" type="text/css" href="index.css">
</head>
```

Internet Content Adaptation Protocol (ICAP) Support

ICAP

- The Internet Content Adaptation Protocol (ICAP) is designed to offload the processing of Internet-based content to dedicated servers. ICAP helps free up resources and standardize how features are implemented.
- A proxy server, such as WebSEAL, can be configured to pass client requests and responses through ICAP servers. These ICAP servers can focus on specific, value-added services, and therefore be more efficient. For example, if an ICAP server handles language translation only, it might be more efficient than a web server that performs many additional tasks.
- ICAP is a "lightweight" HTTP-like protocol. ICAP clients can pass HTTP-based (HTML) messages or content to ICAP servers for adaptation. Adaptation refers to performing the particular value added service, such as content manipulation, for the associated client request or response.
- WebSEAL supports both TCP connections and SSL connections to the ICAP server.
- For more information, see Request For Comments (RFC) 3507 - Internet Content Adaptation Protocol (ICAP): http://www.ietf.org/rfc/rfc3507.txt.

ICAP (Common Use Cases)

- Integrate with 3rd party ICAP products which function as virus/malware scanners.
- Write a quick ICAP Docker Container to check response for “IBM Confidential”

ICAP (Fun with Docker and ICAP)

Update the config file:

```
[ICAP:isamdocker]
URL = icaps://isamdocker:9999/filter?mode=strict
transaction = rsp
timeout = 120
```

Create and attach the POP:

```
pdadmin sec_master> pop create icap-pop
pdadmin sec_master> pop modify icap-pop set attribute ICAP isamdocker
pdadmin sec_master> pop attach /WebSEAL/isam9070.level2.org-default/dashboard/icap.html icap-pop
pdadmin sec_master> server replicate ← Just in case auto-update is disabled.
```

ICAP (Fun with Docker and ICAP)

Before attaching the POP:

ICAP (Fun with Docker and ICAP)

After attaching the POP, responses are sent to my ICAP server which scans the body for “IBM CONFIDENTIAL”:

```
icap    | ProcessConnection
icap    | RESPONSE BODY
icap    | <HTML>
icap    | <HEAD>
icap    |   <META http-equiv="Content-Type" content="text/html; charset=UTF-8">
icap    |   <TITLE>ICAP</TITLE>
icap    | </HEAD>
icap    |
icap    | <BODY>
icap    | <P>
icap    | THIS IS IBM CONFIDENTIAL
icap    | </BODY>
icap    | </HTML>
icap    | Found a confidential doc
icap    | Exception in run: Socket is closed
```

ICAP (Fun with Docker and ICAP)

For the purposes of this presentation I just send back a 400:

```
(69) 2019-09-03-16:45:50.599 BackEnd (X.X.X.X:9999) to WebSEAL (X.X.X.X:50814) sending 130 bytes
ICAP/1.0 400 BAD REQUEST
Date: Mon,  2 Sep 2019  18:55:21 GMT
Server: icap
Connection: close
Encapsulated: res-hdr=0, res-body=0
```

And now the client receives:

```
(69) 2019-09-03-16:45:50.602 WebSEAL (X.X.X.X:443) to Client (X.X.X.X:55359) sending 2795 bytes
HTTP/1.1 500 Internal Server Error
```

Document Links

HTTP Transformations

HTTP-Tag-Value

[acnt-mgt]http-rsp-header

[filter-request-headers]header

[header-names]

[rsp-header-names]

[snippet-filter]

Configuring the attribute collection service

ICAP

Questions for the panel

Ask the panelists a question now

Enter your question in the Q&A panel

Type your question in the box below the ‘Ask’ drop-down option in the Q&A panel.

Select ‘All Panelists’ in the ‘Ask’ drop-down option.

Click Send. Your message is sent and appears in the Q&A panel.

To ask a question after this presentation:

You are encouraged to ask follow-up questions in the forums

https://www.ibm.com/mysupport/s/forumshome

For more information

- **Product Forum: IBM Support Forum for Access Manager**
- **Security Learning Academy: Access Manager Subjects and Courses**
- **IBM Knowledge Center: Access Manager V9.0.7 documentation**
- **Useful links:**

Get started with IBM Security Support **IBM Support**
Sign up for My Notifications **IBM Security Community**

Follow us:

www.youtube.com/user/IBMSecuritySupport

twitter.com/askibmsecurity

http://ibm.biz/ISCS-LinkedIn

Thank you

Follow us:

securitylearningacademy.com

youtube/user/IBMSecuritySupport

@AskIBMSecurity

ibm.biz/IBMSecurityClientSuccess-LinkedIn

securityintelligence.com

xforce.ibmcloud.com

ibm.com/security/community