FORUM SYSTEMS™

MANAGING & ACCELERATING XML WEB SERVICES SECURITY

FORUM SYSTEMS HANDS-ON TRAINING

LAB 14. IDENTITY – PASSWORD AUTHENTICATION ON A REST POLICY

FORUMSYSTEMS

A Crosscheck Networks Company

Legal Marks

Published: September 2014
Forum Systems Hands-on Training – Lab 14. Identity – Password Authentication on a REST Policy
D-ASF-SE-010029

Contents

Introduction

Lab 14. Identity – Password Authentication on a REST Policy

Skill Level

This lab is Intermediate Skill Level. Some existing experience with Forum Sentry and SOAPSonar or completion of the Beginner Level labs is assumed.

Prerequisites

This lab requires the Forum Sentry Training Image, with a licensed copy Forum Sentry and SOAPSonar Enterprise.

Refer to the “FS_Training_Labs_v8-1_Introduction” document for information on the Forum Sentry Training Image and licensing SOAPSonar Enterprise Edition.

This lab assumes a basic understanding of building and testing a REST Policy in Sentry, as outlined in Lab 6.

This lab utilizes the Sentry LDAP Policies built in Lab 13. If these do not exist, follow the instructions in Lab 13 to build the Runtime User LDAP Policy.

This lab utilizes SSL Termination and so requires a key pair in Sentry. If there are no key pairs in your Sentry configuration, either follow the instructions in Lab 7 to generate a key pair or import a key pair (see Lab 7).

Lab Overview

This lab focuses on adding a common form of Password Authentication (HTTP Basic Authentication) to a Sentry REST Policy.

Sentry supports multiple Password Authentication options, with the most commonly used being HTTP Basic Authentication and Form Post Authentication. This lab will use HTTP Basic Authentication.

Password Authentication methods require the client, whether an app or a browser, provide a username and password to a service to authenticate and gain access.

As an API Gateway, Sentry resides upstream from the services and APIs it secures. This is the ideal place to handle the authentication and authorization of the user credentials being provided by the client. This approach removes the user identity and access control processing from the back-end application tier, centralizing it at the gateway layer for improved security, performance, and management.

While Sentry will consume the runtime user credentials, it will typically rely on an external user store to validate the credentials. The most common integration for this is LDAP.

Password Authentication should not be used without SSL as the credentials are not encrypted in any way. Therefore, we’ll also use SSL on the REST policy built in this lab.

In this lab we’ll first build a new REST Policy for the OpenWeatherMap API. Next we’ll build a User ACL and add the runtime user LDAP policy built in Lab 13 to this User ACL. Lastly, we’ll associate the User ACL to the REST policy to require a username and password to access the API.

This lab will provide instructions for enabling HTTP Basic Authentication on a REST policy in Forum Sentry. Topics will include:

1. Building a REST Policy
2. Creating a User ACL
3. Enabling and Requiring HTTP Basic Authentication on a REST Policy

Create a User ACL

In this step we will build a new User ACL in Sentry. This User ACL (access control list) will be applied to the virtual directory of the REST policy and will dictate which user “groups” will have access to the virtual directory.

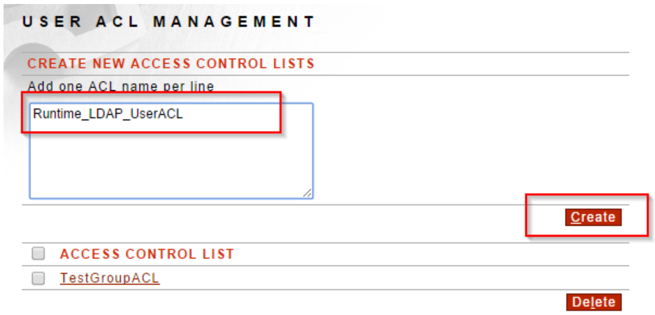
The runtime user LDAP Policy “FS_LDAP_Server_Runtime” will be added to the new User ACL.

User ACLs in Sentry - A User ACL is comprised of Sentry user groups. These user groups can be groups of local users in Sentry referred to as “User Groups” or they can be access control policies that integrate with an external user store (e.g. an LDAP Policy, a SiteMinder policy, etc…).

When viewing a User ACL in Sentry, all user groups in the system are available to select. You add a user group to a User ACL by giving the group Execute privileges.

Follow the steps below to build a new User ACL and add the runtime user LDAP policy to the User ACL.

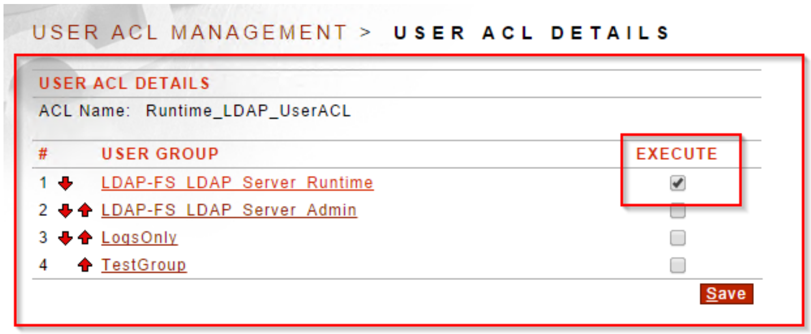
1. Navigate to the Access→Runtime Access→User ACLs page.
2. To create a new User ACL, enter “Runtime_LDAP_UserACL” in the white box and click Create.



Runtime_LDAP_UserACL

□ ACCESS CONTROL LIST
□ TestGroupACL

3. Once created, click on the name of the User ACL to open it.
4. You’ll notice that the LDAP Policies in the system are listed as User Groups, along with any other local User Groups and access control policies. The prefix “LDAP-“ is added to the name of the LDAP Policies.
5. Add the “LDAP-FS_LDAP_Server_Runtime” user group to the User ACL by clicking the box to the right. You’ll notice that the policy will move to #1 in the list. It is possible to have more than 1 group in a User ACL and the groups can be ordered. This tells Sentry which user store to check first when validating credentials.



ACL Name: Runtime_LDAP_UserACL

#	USER GROUP	EXECUTE
1	LDAP-FS_LDAP_Server_Runtime	☑
2	LDAP-FS_LDAP_Server_Admin	□
3	LogsOnly	□
4	TestGroup	□

6. Click Save.

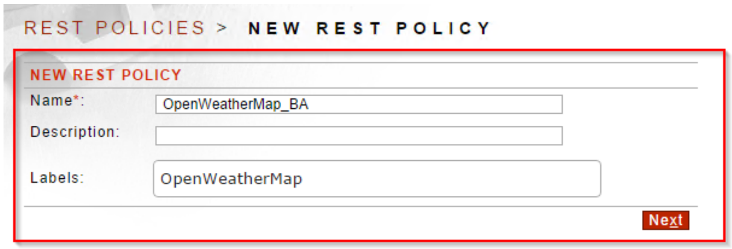
Building a REST Policy with Password Authentication

In this step we will build a new REST Policy in Sentry for the OpenWeatherMap API. This REST Policy will use a new network listener policy. For detailed instructions on building a REST Policy in Sentry, refer to Lab 6 of this series.

In addition to building the REST Policy, we'll apply Password Authentication to the REST Policy.

Follow the steps below to build a REST Policy with HTTP Basic Authentication

1. Navigate to the Gateway→Content Policies→REST Policies page
2. Click New and create a REST Policy with the following criteria:
    a. Name: OpenWeatherMap_BA
    b. Label: OpenWeatherMap



REST POLICIES > NEW REST POLICY

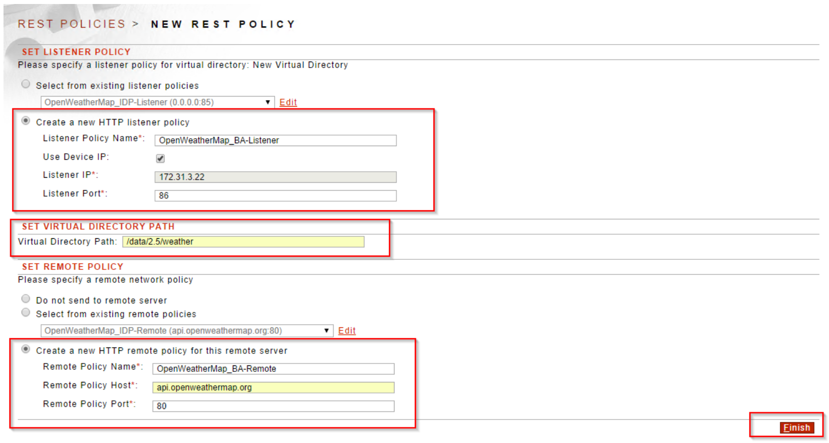
NEW REST POLICY

Name*: OpenWeatherMap_BA

Description:

Labels: OpenWeatherMap

3. On the next screen, you are prompted to select existing remote policies or to build new listener and remote policies, and to enter the virtual path. Create new network policies rather than reusing any existing policies. Enter the following information to build the REST Policy for the OpenWeatherMap API:
    a. Check the "Use Device IP" box
    b. Set the listener port to: 86
    c. Set the Virtual Directory Path to: /data/2.5/weather
    d. Set the Remote Policy Host to: api.openweathermap.org
    e. Set the Remote Policy Port to: 80
    f. Click Finish



REST POLICIES > NEW REST POLICY

SET LISTENER POLICY

Please specify a listener policy for virtual directory: New Virtual Directory

○ Select from existing listener policies

OpenWeatherMap_IDP-Listener (0.0.0.0:85) ▾ Edit

● Create a new HTTP listener policy

Listener Policy Name*: OpenWeatherMap_BA-Listener

Use Device IP: ☑

Listener IP*: 172.31.3.22

Listener Port*: 86

SET VIRTUAL DIRECTORY PATH

Virtual Directory Path: /data/2.5/weather

SET REMOTE POLICY

Please specify a remote network policy

○ Do not send to remote server

○ Select from existing remote policies

OpenWeatherMap_IDP-Remote (api.openweathermap.org:80) ▾ Edit

● Create a new HTTP remote policy for this remote server

Remote Policy Name*: OpenWeatherMap_BA-Remote

Remote Policy Host*: api.openweathermap.org

Remote Policy Port*: 80

4. Access the Virtual Directory page by clicking the link "New Virtual Directory" and configure with the following criteria:

a. Name: OpenWeatherMap_BA

b. Filter Expression: .*

c. ACL Policy: Runtime_LDAP_UserACL

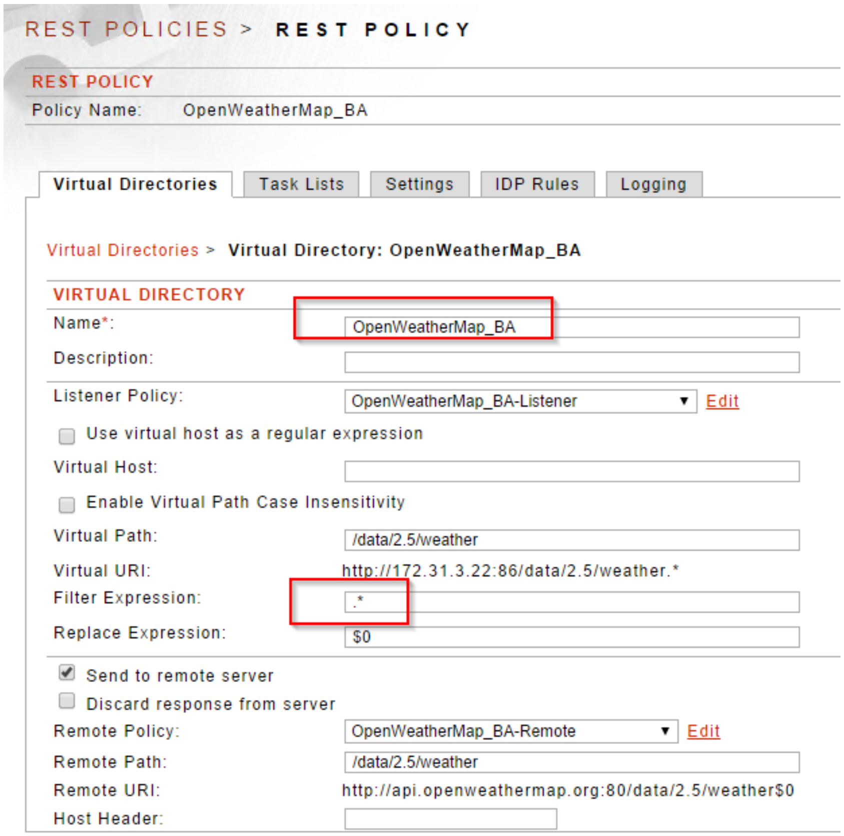
d. Password Authentication: Specify

e. Use basic authentication: Enabled

f. Require password authentication (any type): Enabled

g. Leave all other default values

h. Click Save



REST POLICIES > REST POLICY

REST POLICY

Policy Name: OpenWeatherMap_BA

Virtual Directories > Virtual Directory: OpenWeatherMap_BA

VIRTUAL DIRECTORY

Name*: OpenWeatherMap_BA

Description:

Listener Policy: OpenWeatherMap_BA-Listener ▾ Edit

☐ Use virtual host as a regular expression

Virtual Host:

☐ Enable Virtual Path Case Insensitivity

Virtual Path: /data/2.5/weather

Virtual URI: http://172.31.3.22:86/data/2.5/weather.*

Filter Expression: .*

Replace Expression: $0

☑ Send to remote server

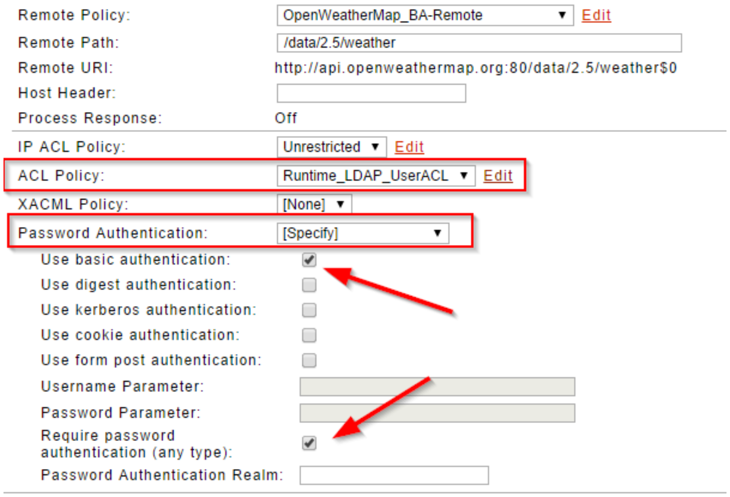
☐ Discard response from server

Remote Policy: OpenWeatherMap_BA-Remote ▾ Edit

Remote Path: /data/2.5/weather

Remote URI: http://api.openweathermap.org:80/data/2.5/weather$0

Host Header:



Remote Policy: OpenWeatherMap_BA-Remote ▾ Edit

Remote Path: /data/2.5/weather

Remote URI: http://api.openweathermap.org:80/data/2.5/weather$0

Host Header:

Process Response: Off

IP ACL Policy: Unrestricted ▾ Edit

ACL Policy: Runtime_LDAP_UserACL ▾ Edit

XACML Policy: [None] ▾

Password Authentication: [Specify] ▾

Use basic authentication: ☑

Use digest authentication: ☐

Use kerberos authentication: ☐

Use cookie authentication: ☐

Use form post authentication: ☐

Username Parameter:

Password Parameter:

Require password authentication (any type): ☑

Password Authentication Realm:

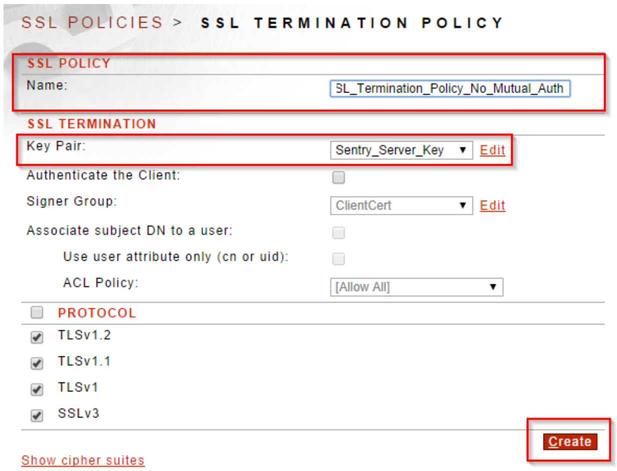
Enable SSL on the REST Policy

Password Authentication should not be used without SSL as the credentials are not encrypted in any way.

In this step, we'll first build an SSL Termination Policy, then we'll change the network listener policy associated with the "OpenWeatherMap_BA" REST policy to use HTTPS.

Follow the steps below to change the REST Policy to use HTTPS.

1. Navigate to the Resources>>PKI>>SSL Policies page.
2. Click New and create a new SSL Termination Policy with the following criteria:
   a. Name: SSL_Termination_Policy_No_Mutual_Auth
   b. Key Pair: Sentry_Server_Key
   c. Leave all defaults
   d. Click Create



SSL POLICIES > SSL TERMINATION POLICY

SSL POLICY

Name: SL_Termination_Policy_No_Mutual_Auth

SSL TERMINATION

Key Pair: Sentry_Server_Key ▾ Edit

Authenticate the Client: ☐

Signer Group: ClientCert ▾ Edit

Associate subject DN to a user: ☐

Use user attribute only (cn or uid): ☐

ACL Policy: [Allow All] ▾

☐ PROTOCOL

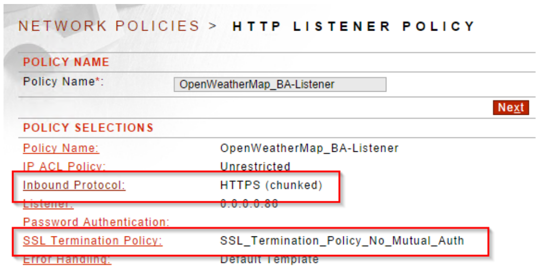
☑ TLSv1.2

☑ TLSv1.1

☑ TLSv1

☑ SSLv3

3. To change the network listener policy from HTTP to HTTPS, navigate to the Gateway→Network Policies→Network Policies page.
4. Click the name of the HTTP Listener "OpenWeatherMap_BA-Listener" and modify as follows:
   a. Click the Inbound Protocol Link: Change to HTTPS
   b. Click Next until you come to the SSL Termination option: Change to "SSL_Termination_Policy_No_Mutual_Auth"
   c. Click Next
   d. Click Finish



NETWORK POLICIES > HTTP LISTENER POLICY

POLICY NAME

Policy Name*: OpenWeatherMap_BA-Listener

POLICY SELECTIONS

Policy Name: OpenWeatherMap_BA-Listener

IP ACL Policy: Unrestricted

Inbound Protocol: HTTPS (chunked)

Listener: 0.0.0.0:80

Password Authentication:

SSL Termination Policy: SSL_Termination_Policy_No_Mutual_Auth

Error Handling: Default Template

5. No further changes are required. Changing the listener updates the REST Policy automatically.

Testing the REST Policy

Now that “OpenWeatherMap_BA” REST Policy has been configured to require HTTP Basic Authentication, test the policy using SOAPSonar.

Follow the steps below to test the password protected REST Policy using SOAPSonar.

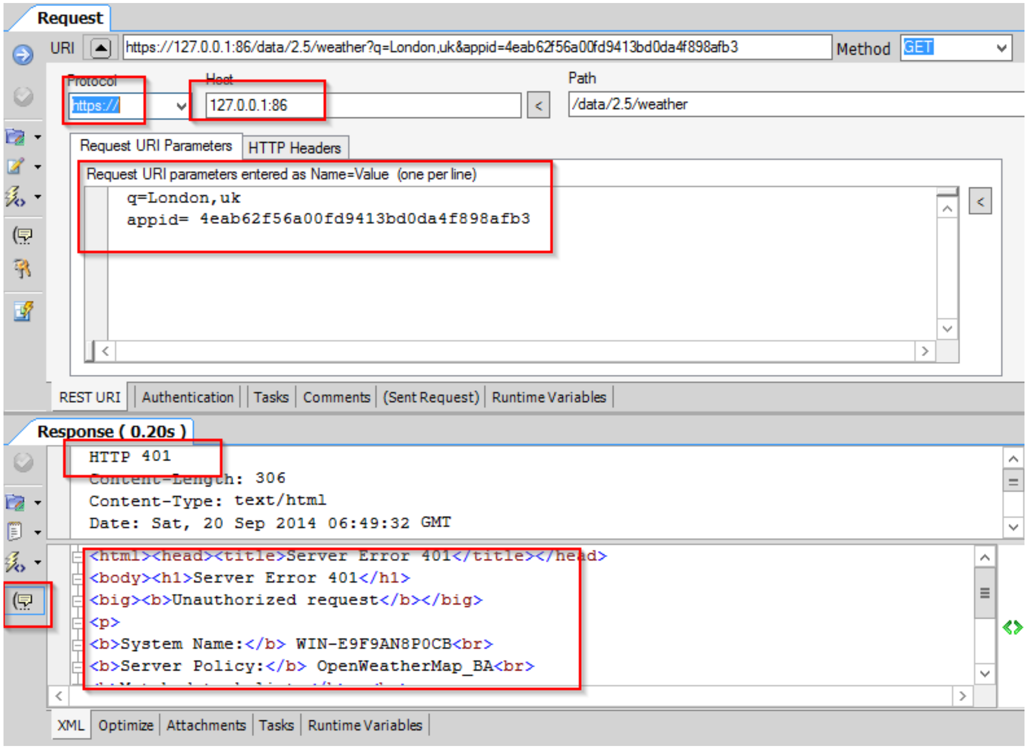
1. Launch SOAPSonar and build a new REST Project.
    a. In the Project QuickStart menu choose Create a REST Project
    b. Name the project “OpenWeatherMap_BA”

2. Build the REST Test case in SOAPSonar using the information listed below. You’ll notice the Request URI field will update automatically.
    a. **Protocol:** HTTPS
    b. **Host:** 127.0.0.1:86
    c. **Path:** /data/2.5/weather
    d. **Request URI Parameters / Name=Value pairs (one per line):**

        q=London,uk
        appid= 4eab62f56a00fd9413bd0da4f898afb3

3. Click the ✓ to commit the settings.

4. Test the policy by sending the request using the ➜ button. You should receive a 401 unauthorized request message. You can view the response headers by clicking the thought bubble icon on the left of the response.



URI: https://127.0.0.1:86/data/2.5/weather?q=London,uk&appid=4eab62f56a00fd9413bd0da4f898afb3
Method: GET

Protocol: https://
Host: 127.0.0.1:86
Path: /data/2.5/weather

Request URI Parameters | HTTP Headers

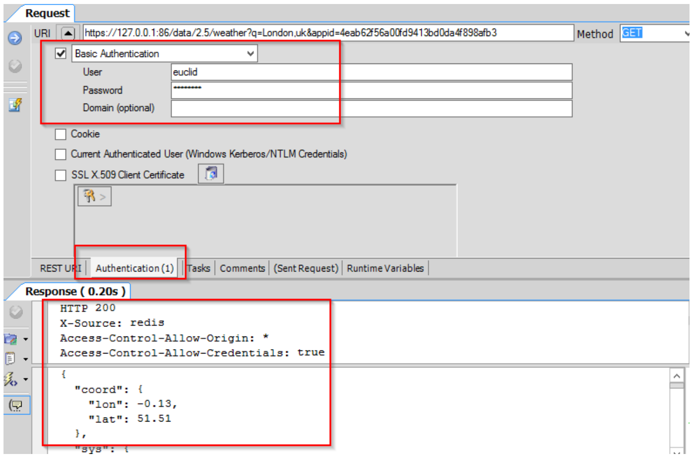
Request URI parameters entered as Name=Value (one per line)

```
q=London,uk
appid= 4eab62f56a00fd9413bd0da4f898afb3
```

```
HTTP 401
Content-Length: 306
Content-Type: text/html
Date: Sat, 20 Sep 2014 06:49:32 GMT
```

```
<html><head><title>Server Error 401</title></head>
<body><h1>Server Error 401</h1>
<big><b>Unauthorized request</b></big>
<p>
<b>System Name:</b> WIN-E9F9AN8P0CB<br>
<b>Server Policy:</b> OpenWeatherMap_BA<br>
```

5. In order to successfully access this API deployed through Sentry, we will need to provide the correct user credentials. To apply HTTP Basic Authentication credentials in SOAPSonar, click the Authentication tab at the bottom of the Request pane. Configure the authentication settings as follows:
   a. Enable Basic Auth – check the box
   b. User: euclid
   c. Password: password
   d. Commit and Test again
   e. Notice the 200 OK response



URI https://127.0.0.1:86/data/2.5/weather?q=London,uk&appid=4eab62f56a00fd9413bd0da4f898afb3 Method GET

☑ Basic Authentication

User euclid

Password ********

Domain (optional)

□ Cookie

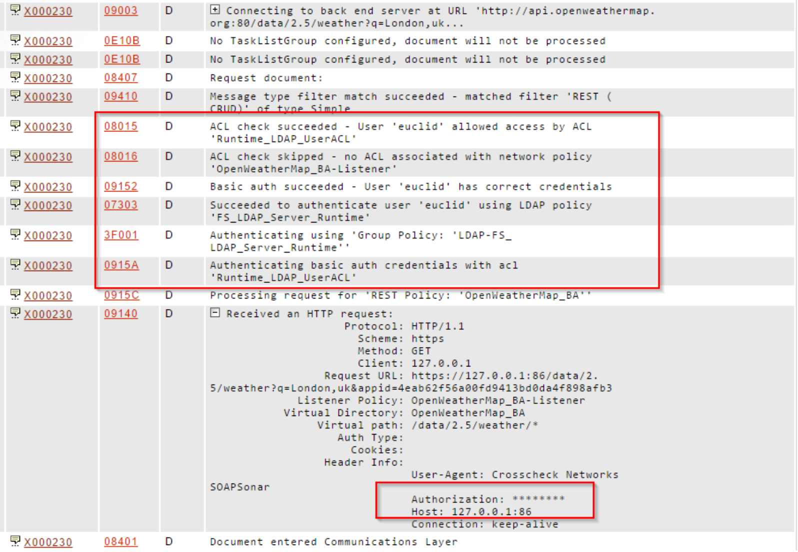
□ Current Authenticated User (Windows Kerberos/NTLM Credentials)

□ SSL X.509 Client Certificate

```
HTTP 200
X-Source: redis
Access-Control-Allow-Origin: *
Access-Control-Allow-Credentials: true
```

```
{
  "coord": {
    "lon": -0.13,
    "lat": 51.51
  },
  "sys": {
```

6. Check the Sentry System log (at DEBUG level) to validate the successful authentication occurred on the incoming request. Notice the following:
   a. The HTTP Authorization header which contains the username and password is obfuscated in the Sentry log
   b. The user is authenticated and authorized via LDAP



```
X000230   09003   D   ⊞ Connecting to back end server at URL 'http://api.openweathermap.
                      org:80/data/2.5/weather?q=London,uk...
X000230   0E10B   D   No TaskListGroup configured, document will not be processed
X000230   0E10B   D   No TaskListGroup configured, document will not be processed
X000230   08407   D   Request document:
X000230   09410   D   Message type filter match succeeded - matched filter 'REST (
                      CRUD)' of type Simple
X000230   08015   D   ACL check succeeded - User 'euclid' allowed access by ACL
                      'Runtime_LDAP_UserACL'
X000230   08016   D   ACL check skipped - no ACL associated with network policy
                      'OpenWeatherMap_BA-Listener'
X000230   09152   D   Basic auth succeeded - User 'euclid' has correct credentials
X000230   07303   D   Succeeded to authenticate user 'euclid' using LDAP policy
                      'FS_LDAP_Server_Runtime'
X000230   3F001   D   Authenticating using 'Group Policy: 'LDAP-FS_
                      LDAP_Server_Runtime''
X000230   0915A   D   Authenticating basic auth credentials with acl
                      'Runtime_LDAP_UserACL'
X000230   0915C   D   Processing request for 'REST Policy: 'OpenWeatherMap_BA''
X000230   09140   D   ⊟ Received an HTTP request:
                                          Protocol: HTTP/1.1
                                            Scheme: https
                                            Method: GET
                                            Client: 127.0.0.1
                                       Request URL: https://127.0.0.1:86/data/2.
                      5/weather?q=London,uk&appid=4eab62f56a00fd9413bd0da4f898afb3
                                   Listener Policy: OpenWeatherMap_BA-Listener
                                 Virtual Directory: OpenWeatherMap_BA
                                      Virtual path: /data/2.5/weather/*
                                         Auth Type:
                                           Cookies:
                                       Header Info:
                                                    User-Agent: Crosscheck Networks
                      SOAPSonar
                                                    Authorization: ********
                                                    Host: 127.0.0.1:86
                                                    Connection: keep-alive
X000230   08401   D   Document entered Communications Layer
```

END

Additional Testing and More Reading

BACK IT UP!

It is recommended that you export your REST Policy and/or your full Sentry configuration after completing this lab.

To export the REST policy, navigate to the REST Policies page, select the REST policy and use the GDM Export option to export the policy (and all dependencies) as a password encrypted FSG file. This can later be imported on the System→Configuration→Import/Export screen.

To export your full Sentry configuration, navigate to the System→Configuration→Import/Export screen and use the Export option in the center of the page to export the full Sentry configuration file as a password encrypted FSX file. This can later be imported on the same screen.

Backup your SOAPSonar project file by using the File→Save As option. All of your test cases will be saved in an .SSP file.

We recommend including the lab number in the name of the export files.

Additional Tests and Discussion Topics

1. What happens when you use the wrong user credentials?
2. Test the service with a browser. Are you challenged for credentials? What do you see in the log?
3. Build a task list that maps a user attribute retrieved from the LDAP server into an HTTP Header that is sent to the remote server.

Additional Information

Details of the online LDAP Server from Forum Systems:
http://www.forumsys.com/tutorials/integration-how-to/ldap/online-ldap-test-server/

For more information, review the following Forum Sentry Admin Guide:

1. Access Control Guide
2. REST Policies Guide

About Forum Systems

Forum Systems is the global leader in API and Cloud Security technology with industry-certified, patented, and proven products deployed in the most rigorous and demanding customer environments worldwide. Forum Systems has been an industry leader for over 12 years and has built the core architecture of its technology on the foundation of FIPS 140-2 and NDPP. Forum Systems security-first mindset enables trusted, network edge deployments of its technology for protecting critical enterprise transactions.

Our product technology is purpose-built and designed for mission-critical, enterprise-class scalable solutions where business solutions require the modern day security and identity enforcement protection, while enabling a scalable architecture and low-latency, high-volume throughput.

Forum Systems supports global enterprise customers across industries in commercial, government, and military sectors. Forum Systems technology provides the leading-edge of modern-day cyber-security innovation with integrated identity and SSO features that enable out-of-the box business solutions with point-and-click technology.

Forum's patented; FIPS 140-2 and NDPP certified hardware and virtual products make modern-day business communications secure by actively protecting and accelerating data exchange and API service access across networks and business boundaries. For more information, please visit www.forumsys.com.