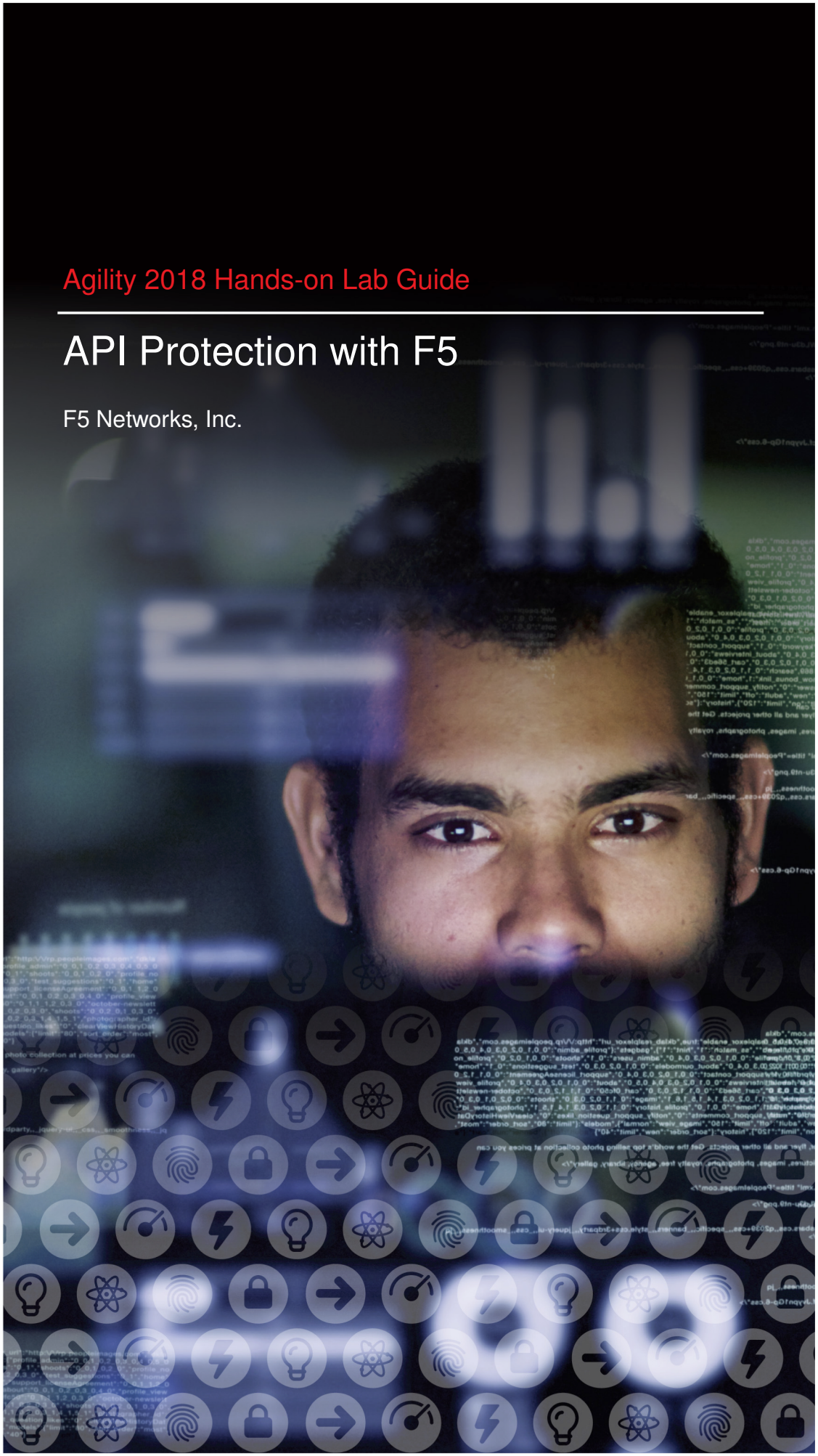


Agility 2018 Hands-on Lab Guide

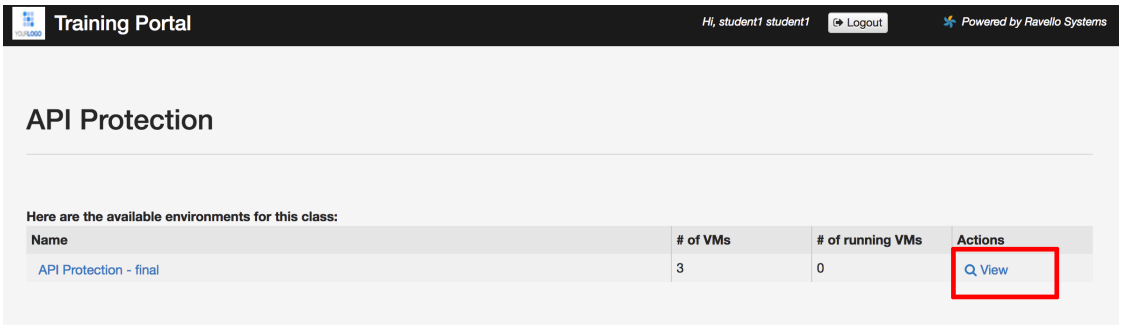
API Protection with F5

F5 Networks, Inc.

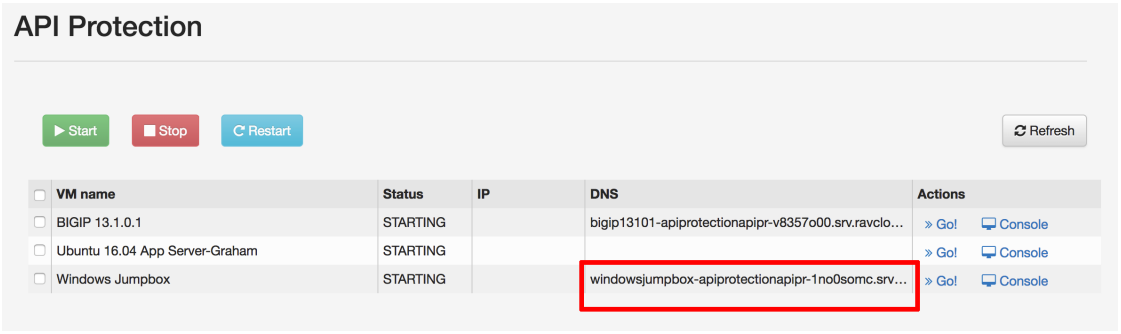
Contents:

1 Getting Started

Instructor will provide you with access credentials to training portal as well as login page.



Once logged in you should tap on “View”. You should be able to see Virtual Machines as shown below. Copy DNS name of Windows jumpbox to clipboard and use your Remote Desktop client to connect.



Note: All work for this lab will be performed exclusively from the Windows Jumpbox. No installation or interaction with your local system is required.

1.1 Lab Credentials

The following table lists access credential for all required components:

Component	Credentials
Windows Jumpbox	`admin/admin`
F5 BIG-IP VE	`admin/admin`

The BIG-IP VE is accessible from the Windows Jumpbox at https://192.168.1.5

1.2 Lab Topology

The following components have been included in your lab environment:

- 1 x F5 BIG-IP VE (v13.1)
- 1 x Linux Webserver (xubuntu 14.04)
- 1 x Windows Jumphost

On the picture below you can see network topology. Basically, you will be sending various API calls to API server proxied through BIG-IP VE.

Traffic from Windows Jumpbox will be proxied through the BIG-IP to API Server.

2 API Protection with F5

In this section you will find guidelines for completion API protection lab exercises.

2.1 Making API requests with POSTMAN

2.1.1 Using Postman to make API requests

In this module you will learn how to make API requests with the Postman client to simulate calls that might be made as part of an application, for instance, a mobile app, native client app, client side webapp, or server to server API request.

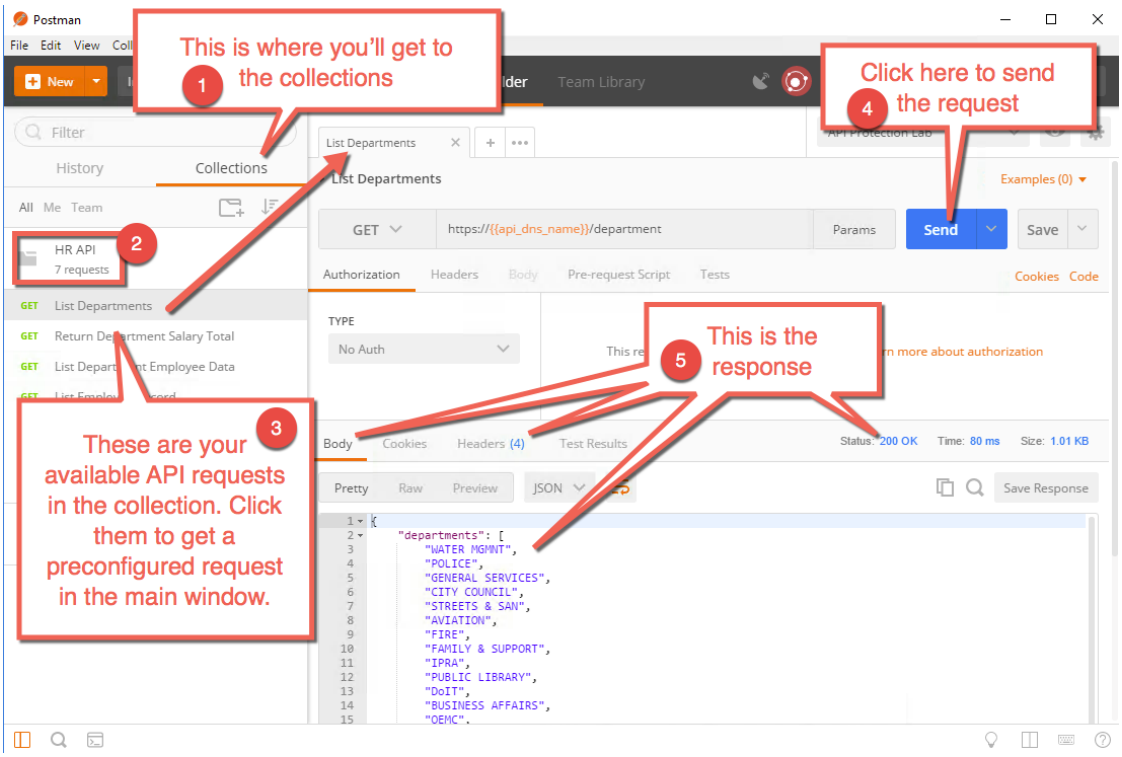
2.1.2 Connect to Client Jumphost and launch Postman

1. RDP to the client jumphost
2. Launch the Postman application. The icon looks like this:

2.1.3 Learn how to use the preconfigured API request collection

In this task you will learn how to use the preconfigured set of requests in the HR API collection.

1. Click Collections
2. Click HR API
3. Click List Departments
4. Click Send
5. Notice the returned list of departments



2.1.4 Learn how to change environment variables

In this task you will learn how to change the environment variables that are configured to alter which department you are querying data for. In this case the variables are used in the URI, but there are other variables used in some queries in the body as well.

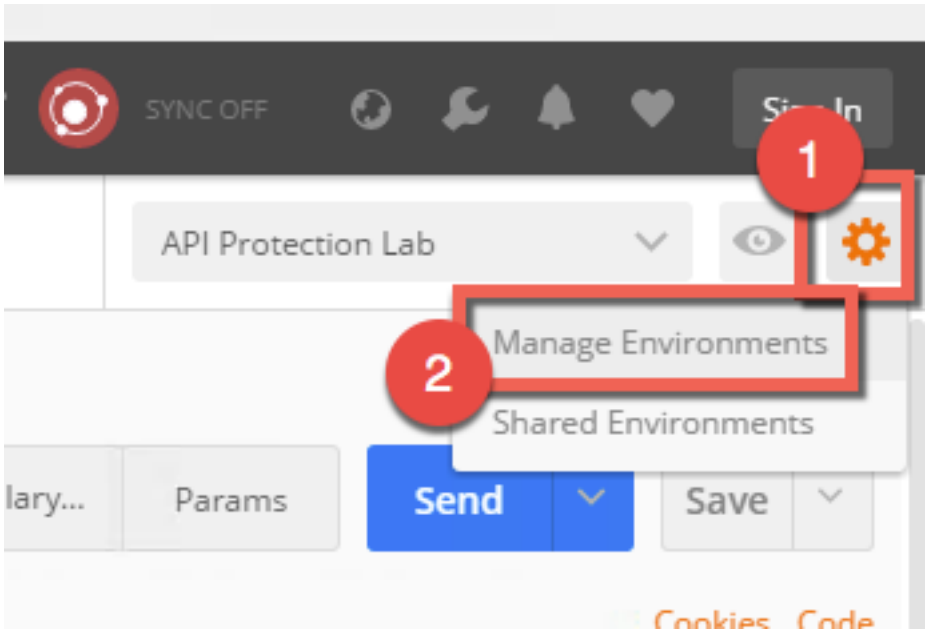
Determine Police Department Salary Total

1. Click on the Return Department Salary Total request in the collection
2. Click Send
3. Notice the total returned is 1106915639.7999947

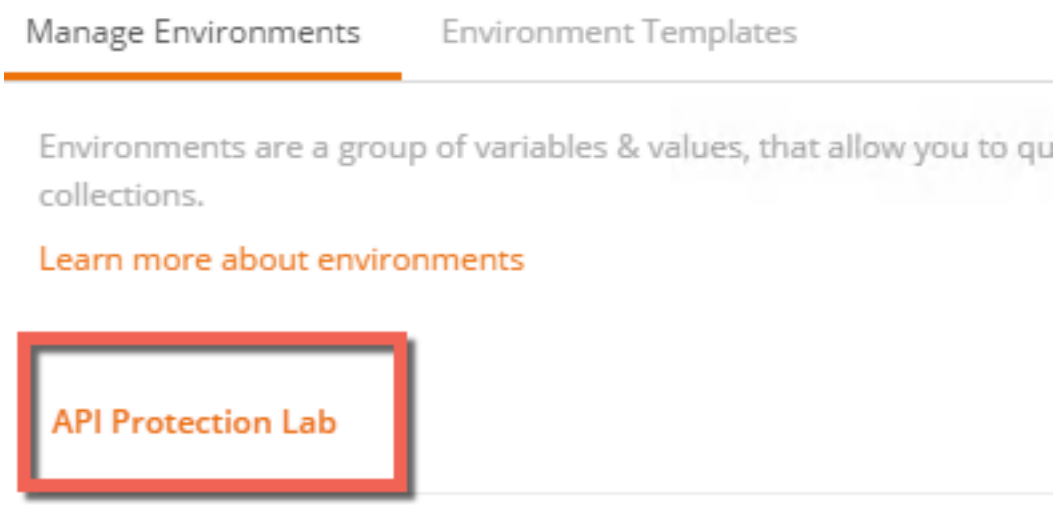
Change environment variable for department

1. Notice the GET request URI has a variable in it named {{department}}

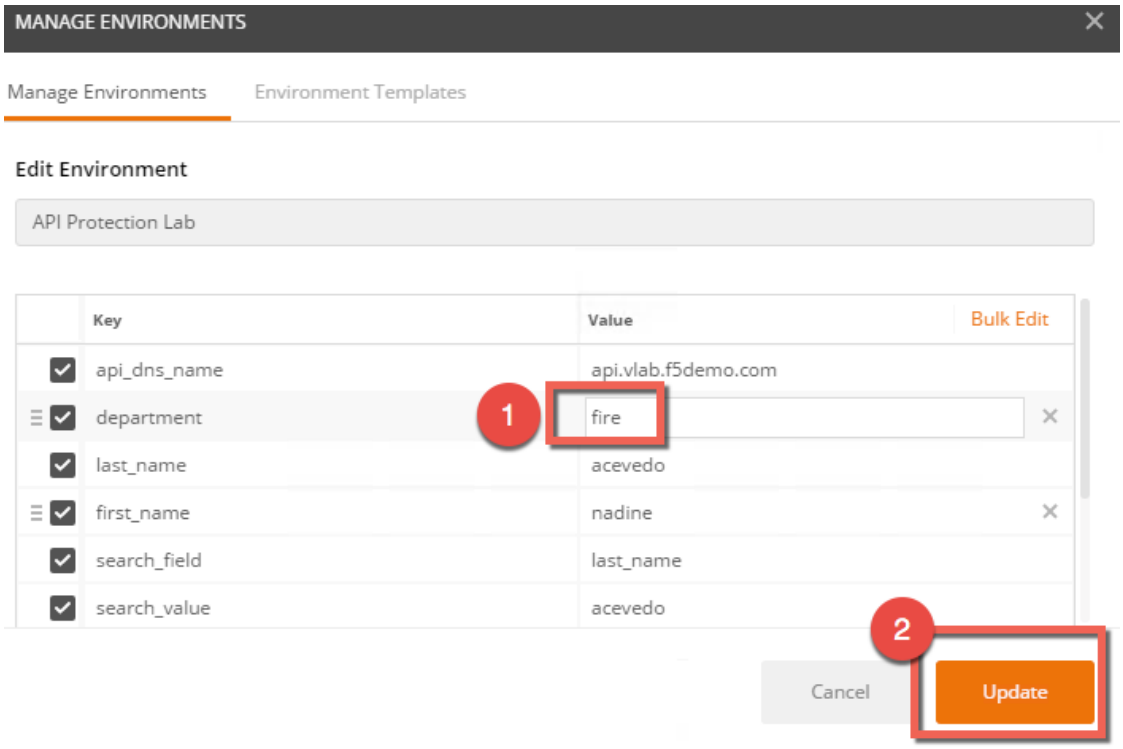
2. Notice in the top right we have an environment set named “API Protection Lab”.
3. Click the gear in the top right, then click “manage environments”.



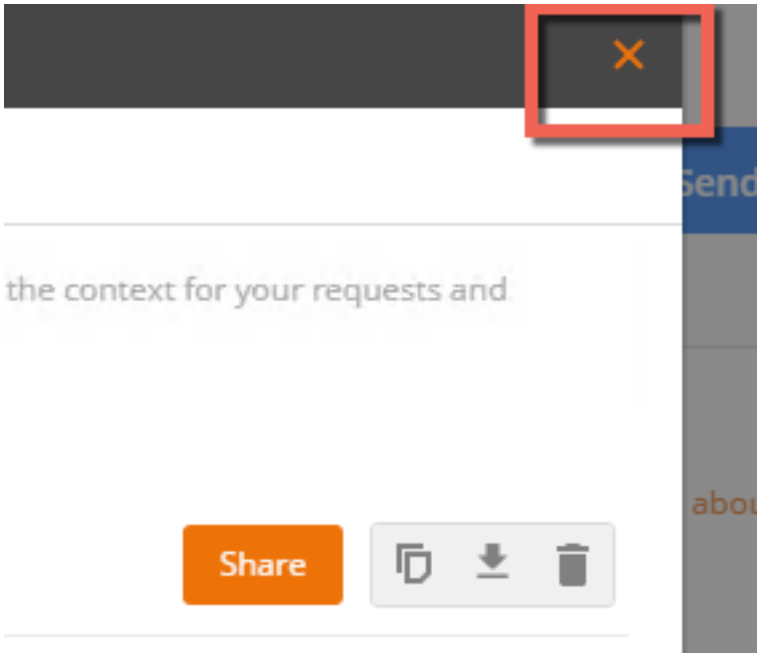
4. Click API Protection Lab



5. Change the value for department from “police” to “fire” then click update



6. Click the X in the top right to close the manage environments window



Determine Fire Department Salary Total

1. Click Send
2. Notice the total returned is now 457971613.68

2.1.5 Return the Environment variables to default

1. Change the department variable back to “police”

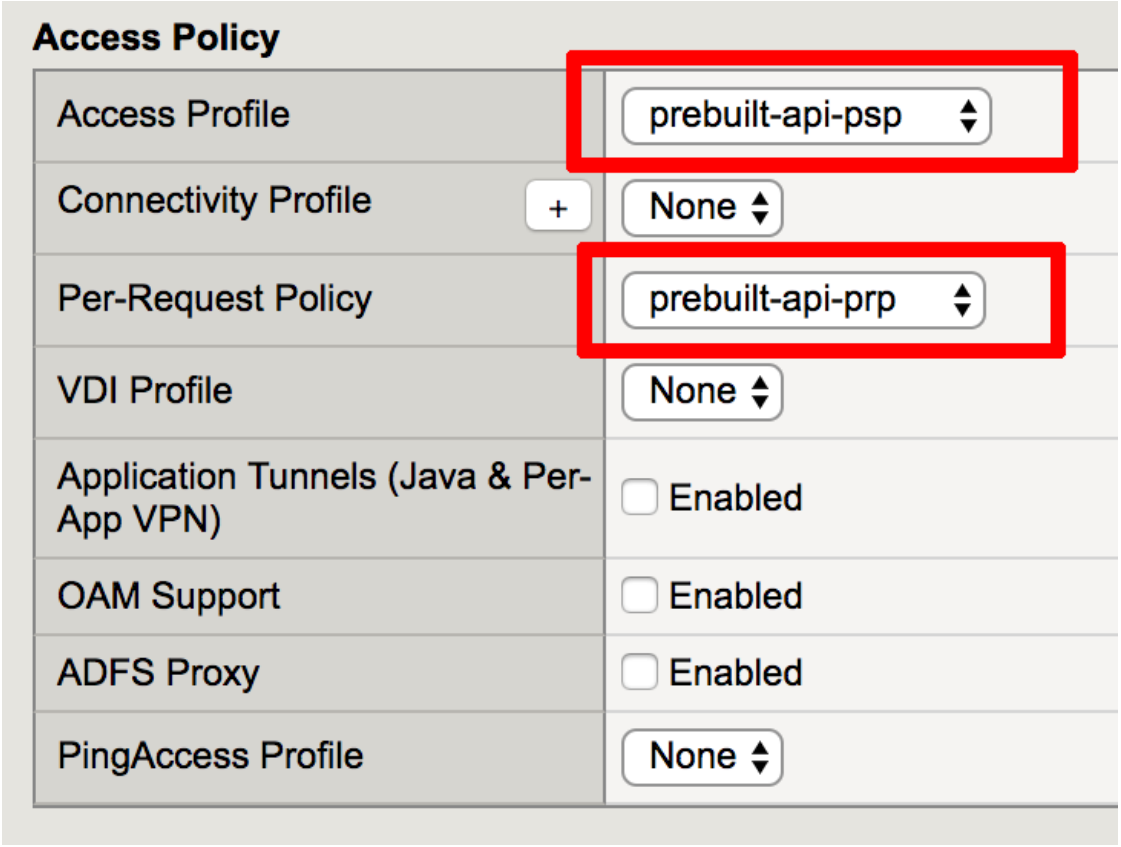
2.2 Implement Coarse-Grain Authorization

In this module you will implement authorization requirements. You will require a valid JWT (JSON Web Token) before a client can access the API. You will then gather a valid JWT and leverage it to make an API request.

2.2.1 Add the policies to the virtual server

In this task you will add the policies you created to the virtual server.

1. Open web-browser, connect to BIG-IP https://https://192.168.1.5 (login: admin, password: admin) and click Local Traffic -> **Virtual Servers**
2. Click **api.vlab.f5demo.com**
3. Change Access Profile from none to **prebuilt-api-psp**
4. Change Per Request Policy from none to **prebuilt-api-prp**

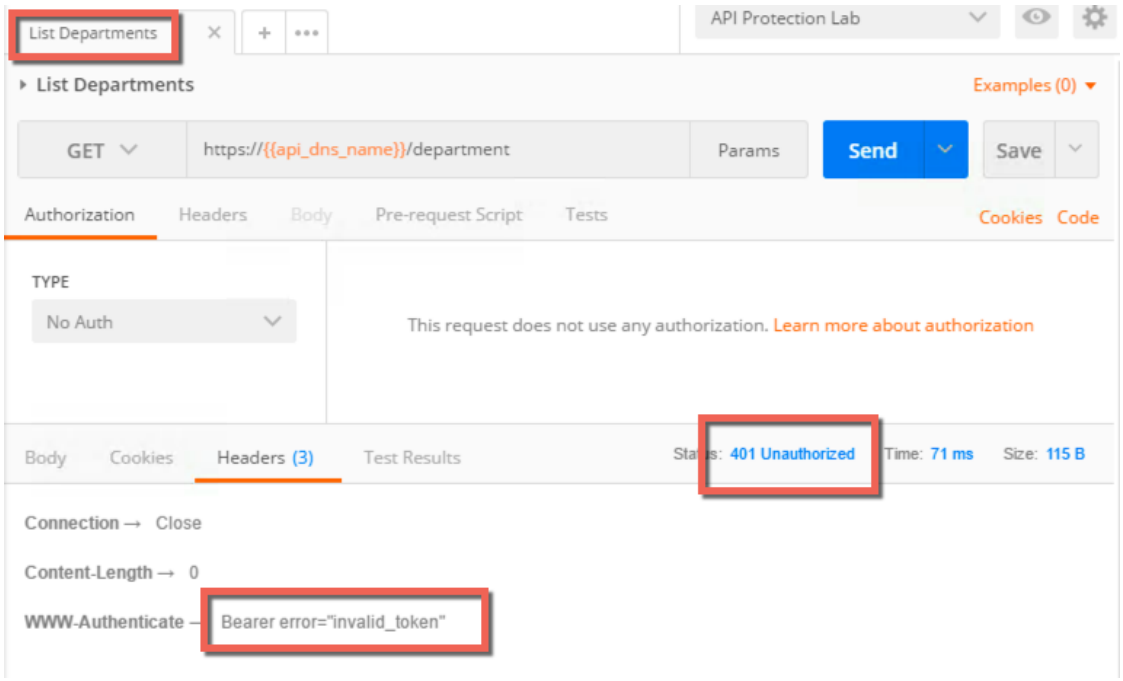


5. Click Update

2.2.2 Test access to the API

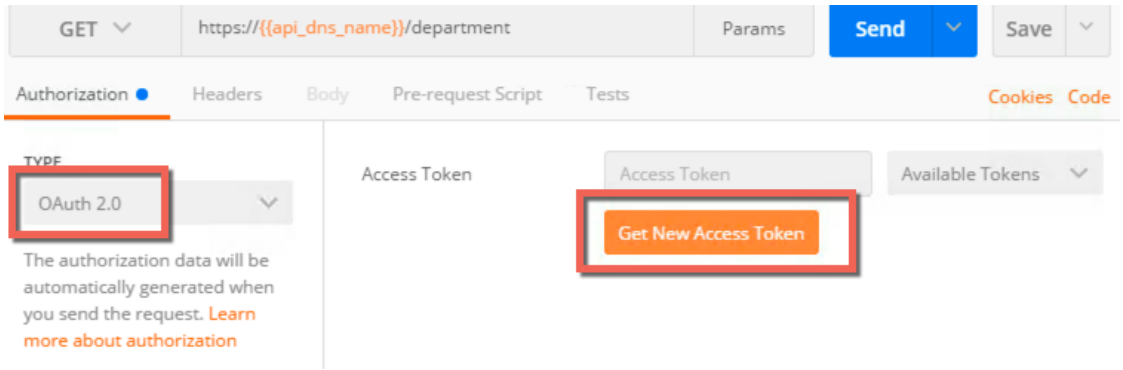
In this task you will test your access to the API and find it is blocked because you do not present a valid JWT.

1. Open Postman on the jumphost client
2. Select List Departments from the HR API collection and send the request
3. Review the response, note the 401 unauthorized and the header indicating you did not present a valid token



2.2.3 Get a JWT from the Authorization Server

1. Click the **type** drop down under the **authorization** tab.
2. Select **OAuth 2.0**
3. Click **Get New Access Token**



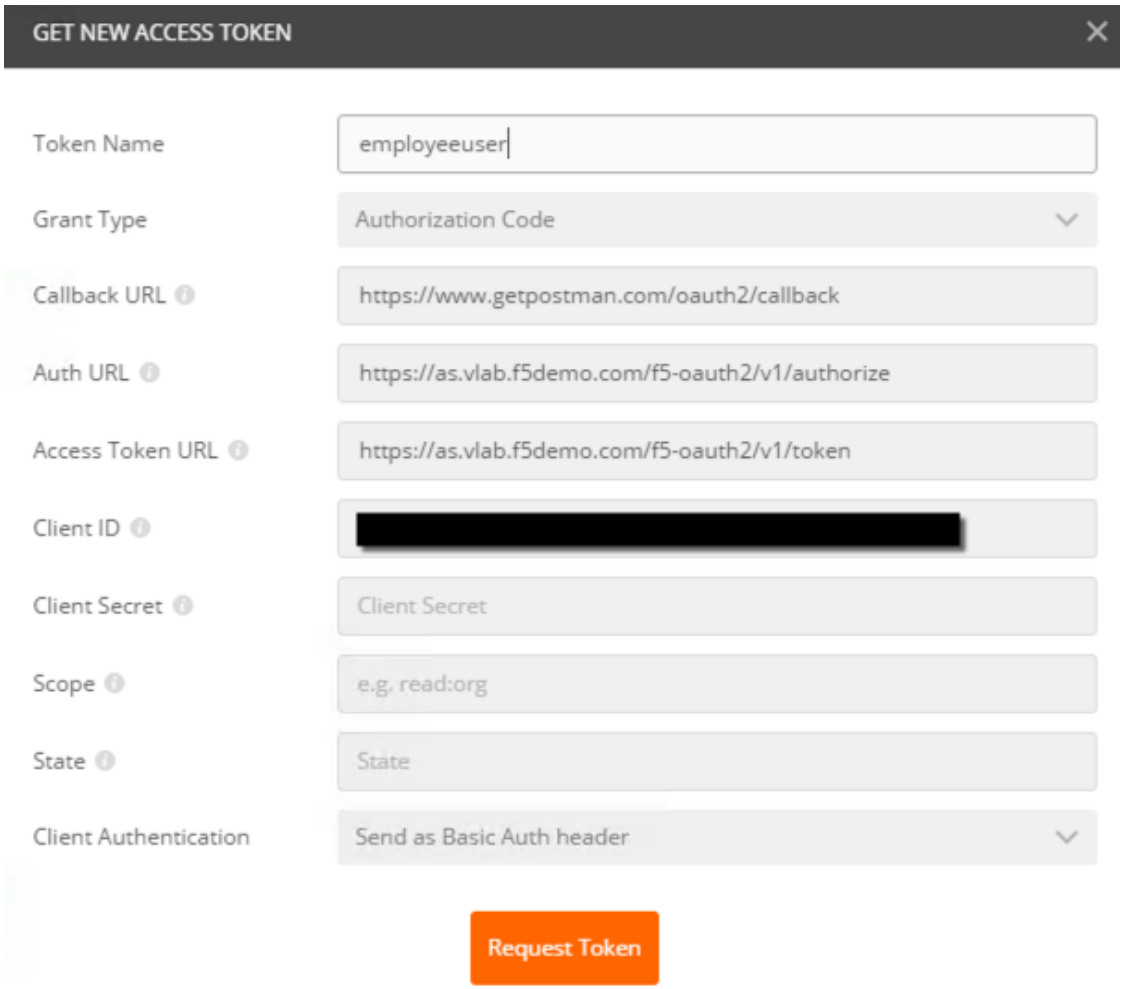
Postman provides a mechanism to handle the OAuth client workflow automatically. This means it will handle getting the authorization code and then exchange it for an access token, which you will use. Without this

you would make two separate requests, one to get an authorization code and another to exchange that for an access token.

1. Fields should be prefilled, but verify they match the below:

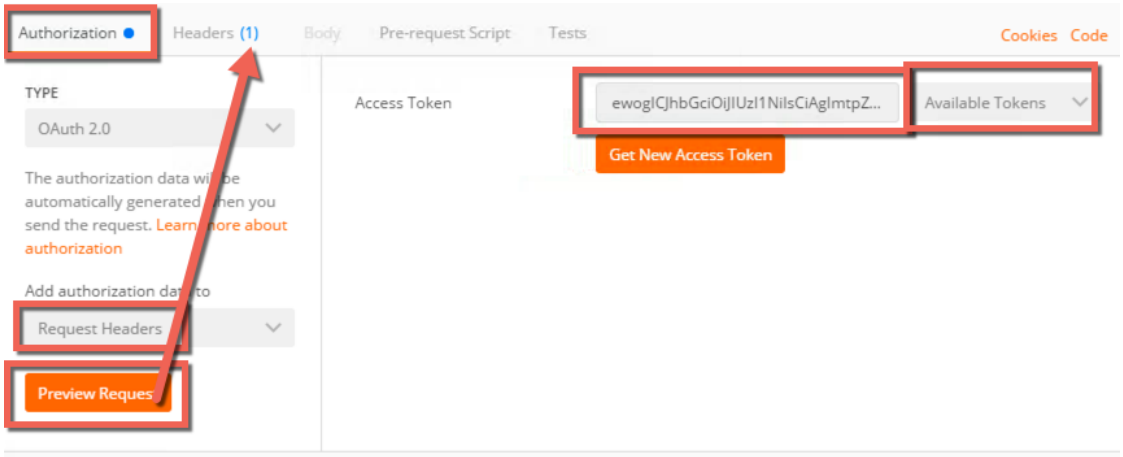
Field	Value
Token name	employeeuser
Grant Type	Authorization Code
Callback URL	https://www.getpostman.com/oauth2/callback
Auth URL	https://as.vlab.f5demo.com/f5-oauth2/v1/authorize
Access Token URL	https://as.vlab.f5demo.com/f5-oauth2/v1/token
Client ID	9f1d39a8255e066b89a51f56b27506d39442c4f608c2f859
Client Authenticatin	Send as Basic Auth header

Most of this data is provided by the authorization server. The callback URL specified here is a special callback URL that the Postman client intercepts and handles rather than calling out to the getpostman.com website.

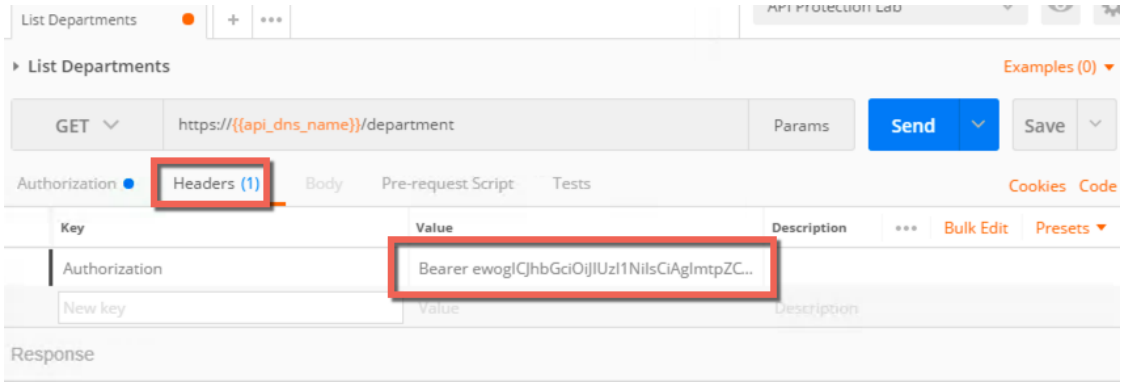


1. Click **Request Token**
2. Select **employeeuser** in the authentication window that pops up and click **Logon**
3. Click the **X** to close this window

4. Make sure **employeeuser** is selected under Available Tokens drop down
5. Select **Request Headers** from the Add Authorization Data To drop down
6. Click **Preview Request**, the result should be this:

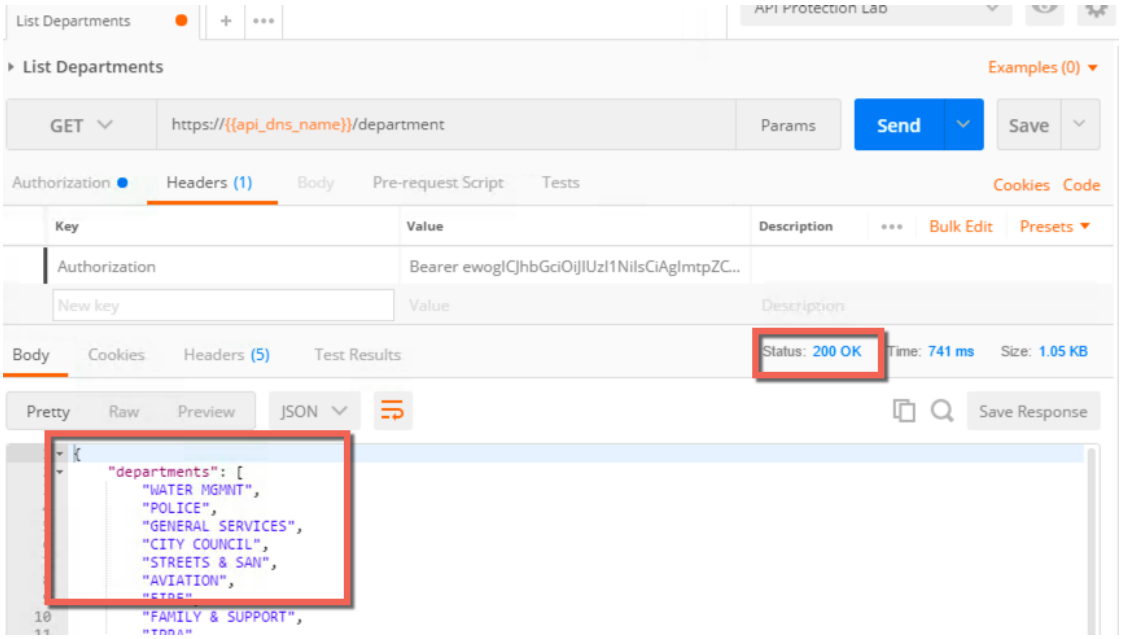


7. Go to the **Headers** tab and review the inserted **Bearer** token header:



2.2.4 Send the request with JWT and review response

1. Click **Send** and review the response.
2. Note that now it is a 200 OK instead of 401 Unauthorized and that you have response data in the body.



You have now implemented coarse grained authorization and are requiring clients to request a JWT from a trusted authorization server before allowing access to the API.

2.3 Adding Fine-Grain Authorization

2.3.1 Adding Fine-Grain Authorization

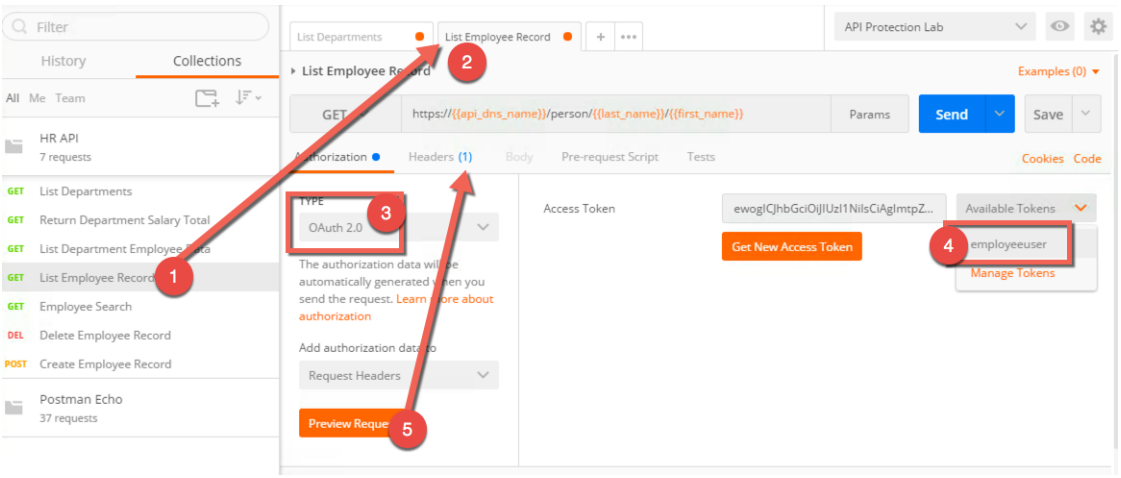
In this module you will add fine-grain controls to your policy to restrict access to parts of the API based on parameters in the JWT. The example will relate to user group membership, but it could be many parameters (e.g. company, user, group, as source, etc).

The goal is to restrict access to any person API requests to only members of the HR department.

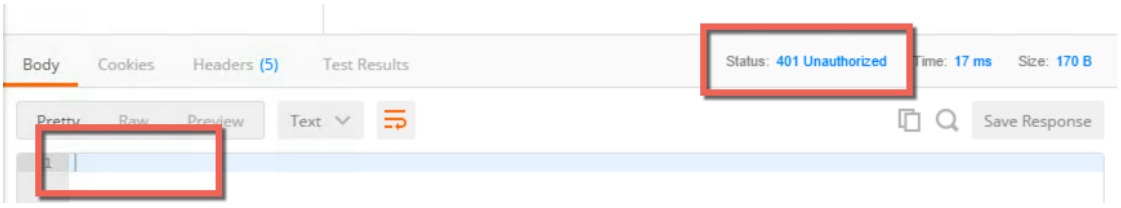
2.3.2 Validate your policy blocks access to person requests without HR group membership

In this task you will test the settings you just put in the per request policy. You are expecting to be denied access to the /person URL because employeeuser is not in the HR group that you have marked as a required value in the JWT.

1. On the left side, select **List Employee Record**. It will now appear in another tab in the middle section and you should select it if it is not already.
2. Under Authorization type select **OAuth 2.0** for the type
3. From the **Available Tokens** drop down, select **employeeuser**
4. Make sure Add Authorization Data is set to **Request Headers**
5. Click **Preview Request** and note the header has been inserted



6. Click **Send**
7. The result should be a **401 unauthorized** with no data in the body. The header will report an invalid token.

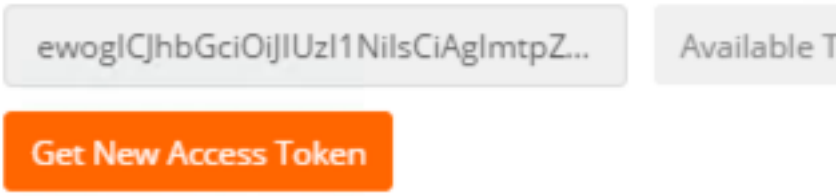


You were denied access because the JWT retrieved by this user is not allowed to access that data. We can resolve this by using credentials that will generate a JWT valid for this request.

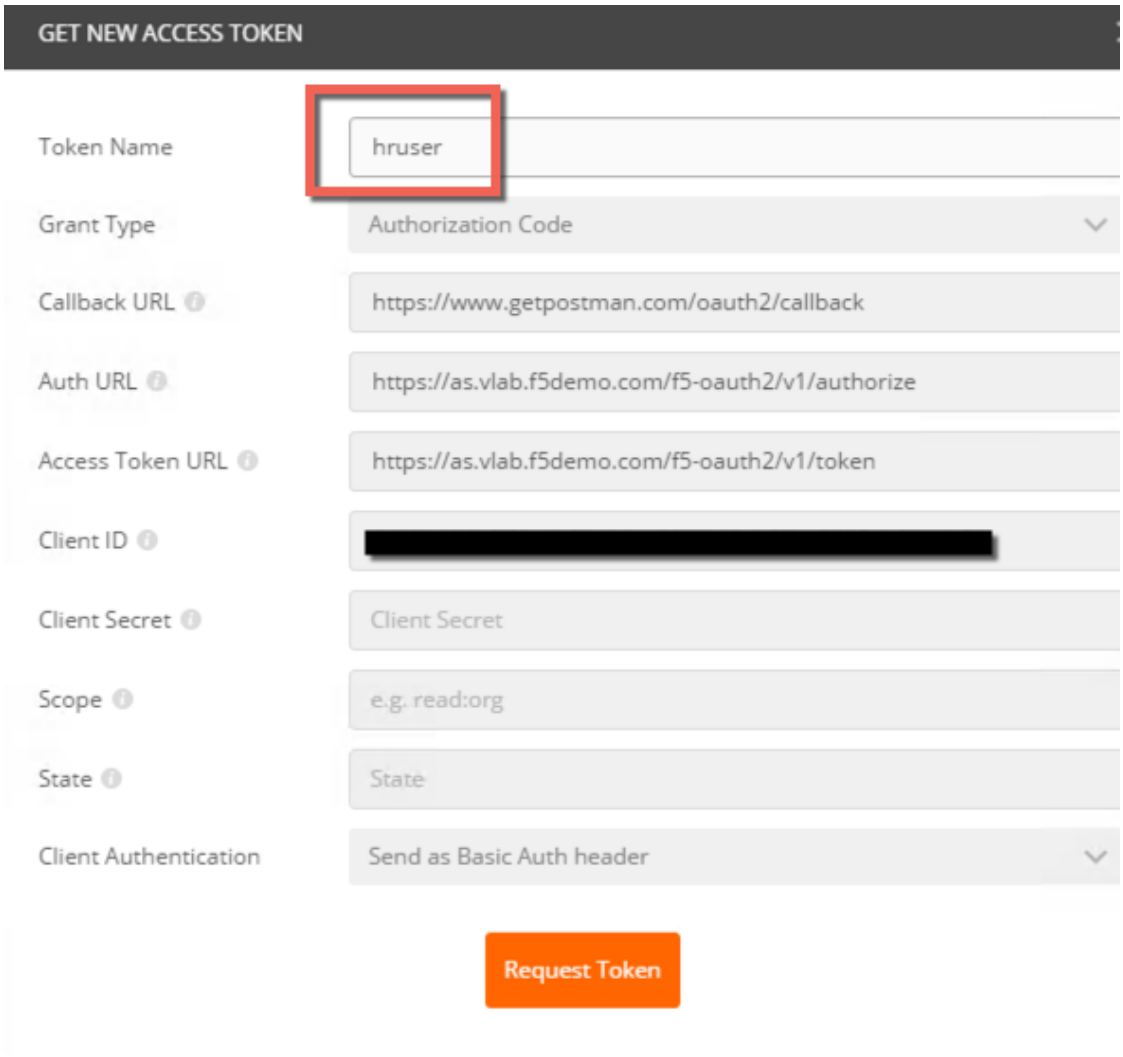
2.3.3 Acquire a JWT for hruser and validate it can access /person

In this task you will get another JWT and use that to gain access to the /person portion of the API.

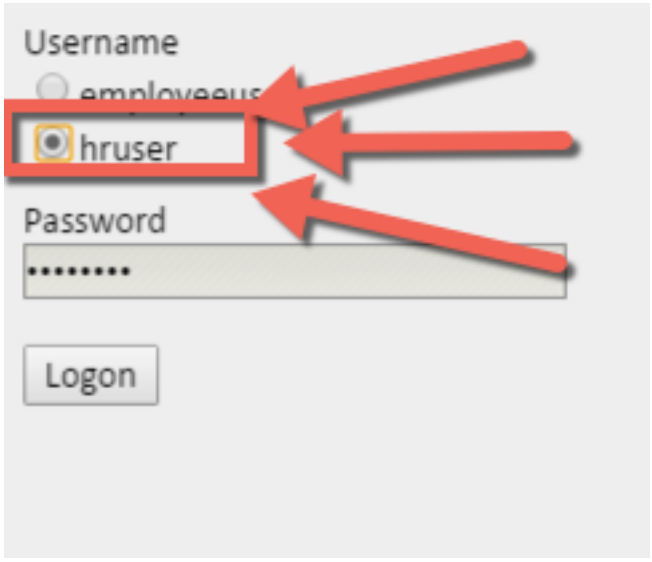
1. Click Get **New Access Token**



2. Change the token name to **hruser**, the rest of the settings should be already correct.

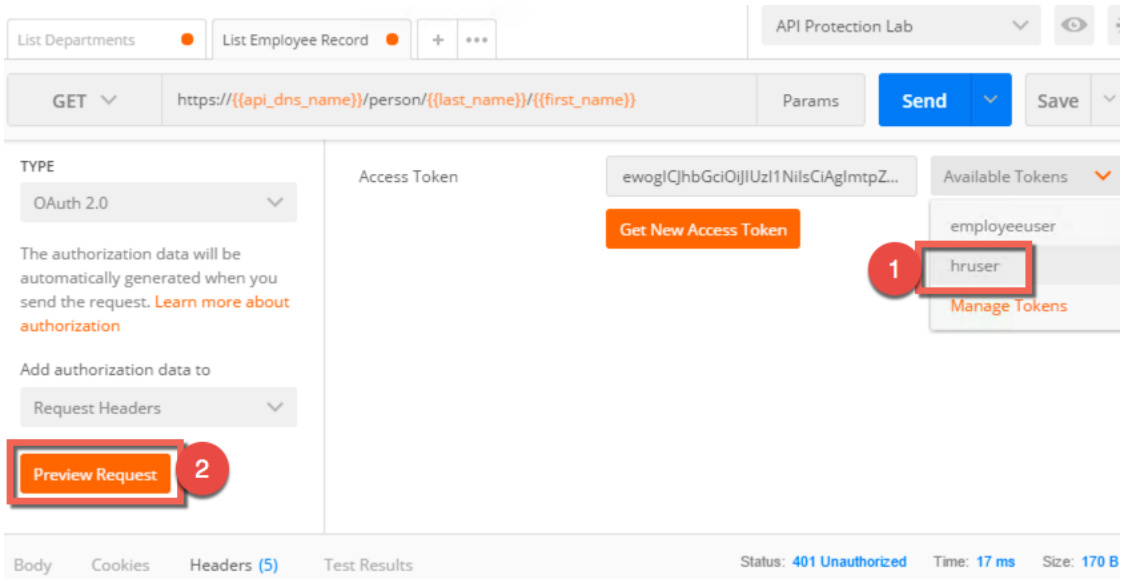


3. Click **Request Token**
4. Select **hruser** at the logon page and press logon.

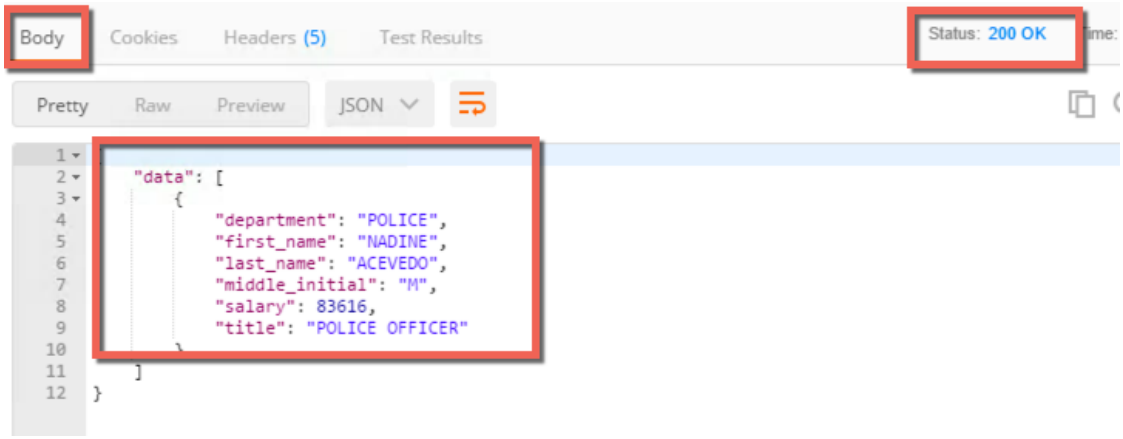


5. A JWT should be returned and your JWT management token window will look like this:

6. Notice you now have two tokens, and click the X to close the window
7. Select hruser from the Available Tokens drop down
8. Click Preview Request



9. Click Send, you should get a 200 OK response and data in the response body like this:



10. You can now change the token used on any request by using this process:
    (a) Select the request
    (b) Select the Authorization tab
    (c) Select OAuth 2.0 from the type drop down menu
    (d) Select the correct token from the Available Tokens drop down menu
    (e) Make sure Authorization Data is set to Request Headers
    (f) Click Preview Request to add the token to the headers
    (g) Click Send on the request

In this module we've used group membership to restrict access to particular URIs, but in production you may encounter many different variations. For example, an iRule can set an APM session variable equal to the request method (e.g. GET, POST, etc) and then in the Per Request Policy you can branch on method, only allowing POST from certain users, groups, IPs, etc

JWTs are typically short lived and may or may not use refresh tokens. In this lab the JWTs have been set as valid for several hours so you will not need to get new JWTs during the lab.

More information about BIG-IP configuration settings available here *F5 BIG-IP configuration deep dive*

2.4 API Protection By ASM

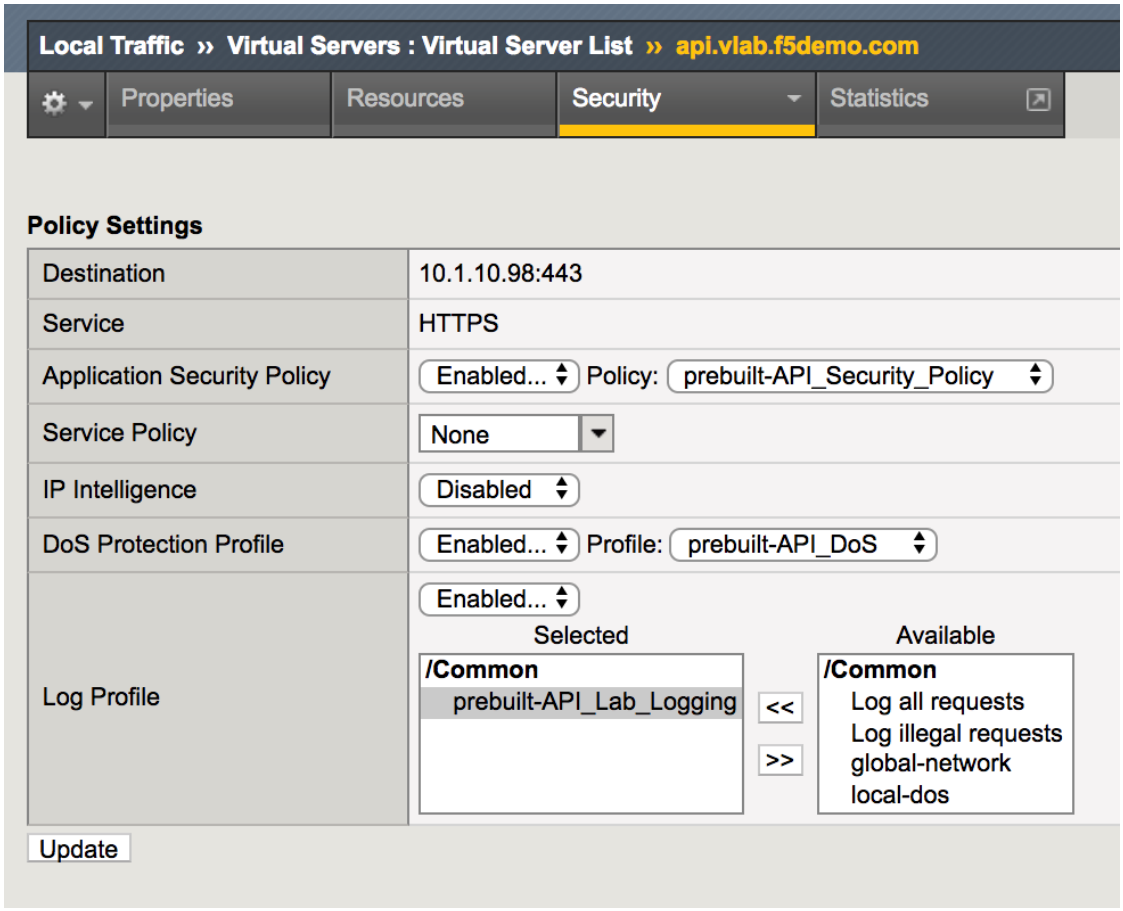
In this module, you will demonstrate how ASM can protect JSON API against JSON parser attack, SQL Injection, Directory Traversal and DoS attacks.

2.4.1 Apply API Protections Profiles

In this task, you will apply the DoS protection profile for the API interface that will protect against known bots and DoS attacks.

1. Navigate to Local Traffic->Virtual Servers->Virtual Server List-> api.vlab.f5demo.com
2. Open Security->Policies tab.
3. Change the state of **Application Security Policy** to Enable and make sure prebuilt-API_Security_Policy is selected.

4. Change the state of **DoS Protection Profile** to Enable and select **prebuilt-API_DoS** from the drop-down menu.
5. Change the state of **Log Profile** to Enable. Highlight **Prebuilt-API_Lab_Logging** from the list of available profiles and tap **<<** to move it into Selected.

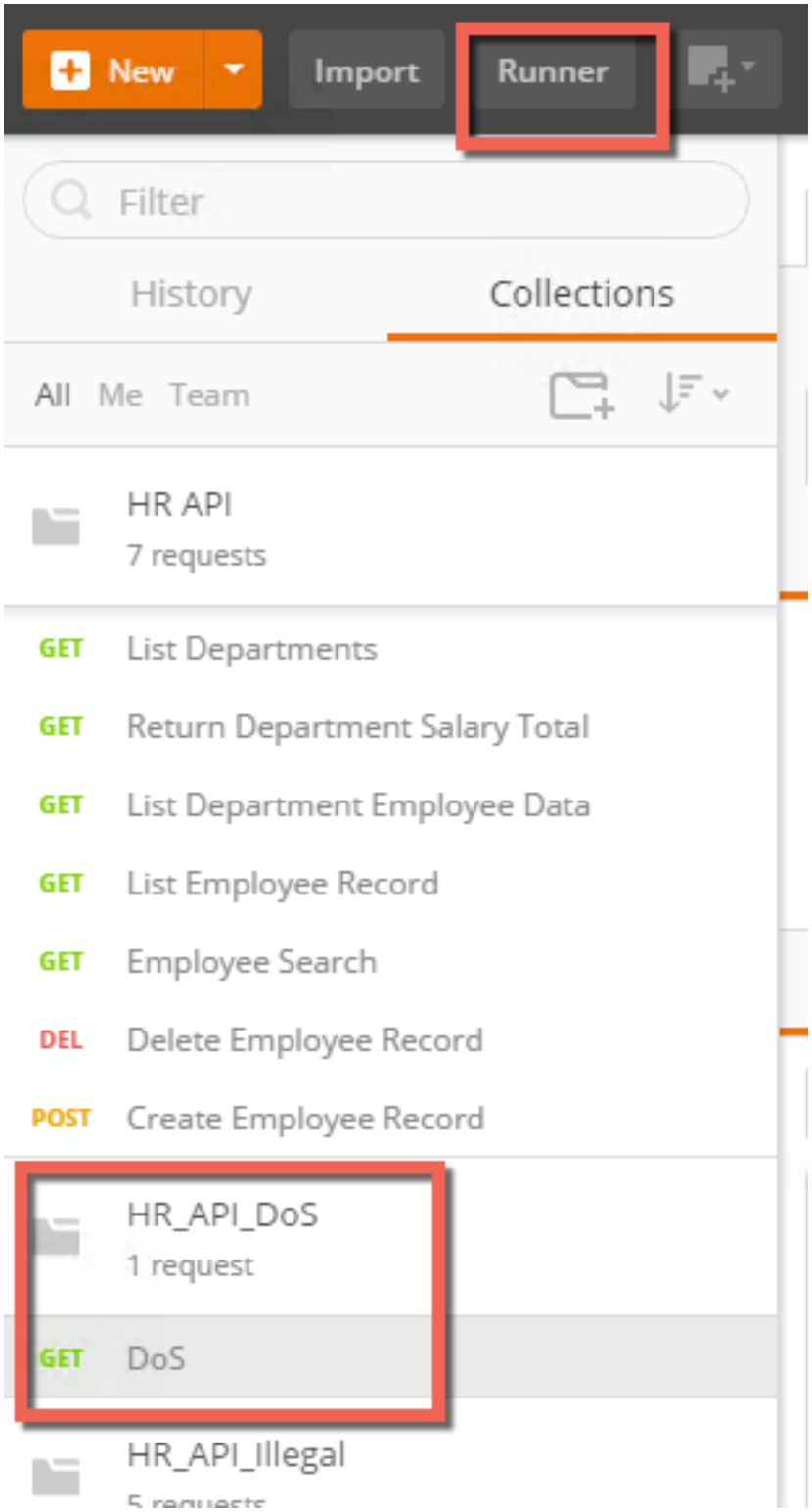


6. Click Update

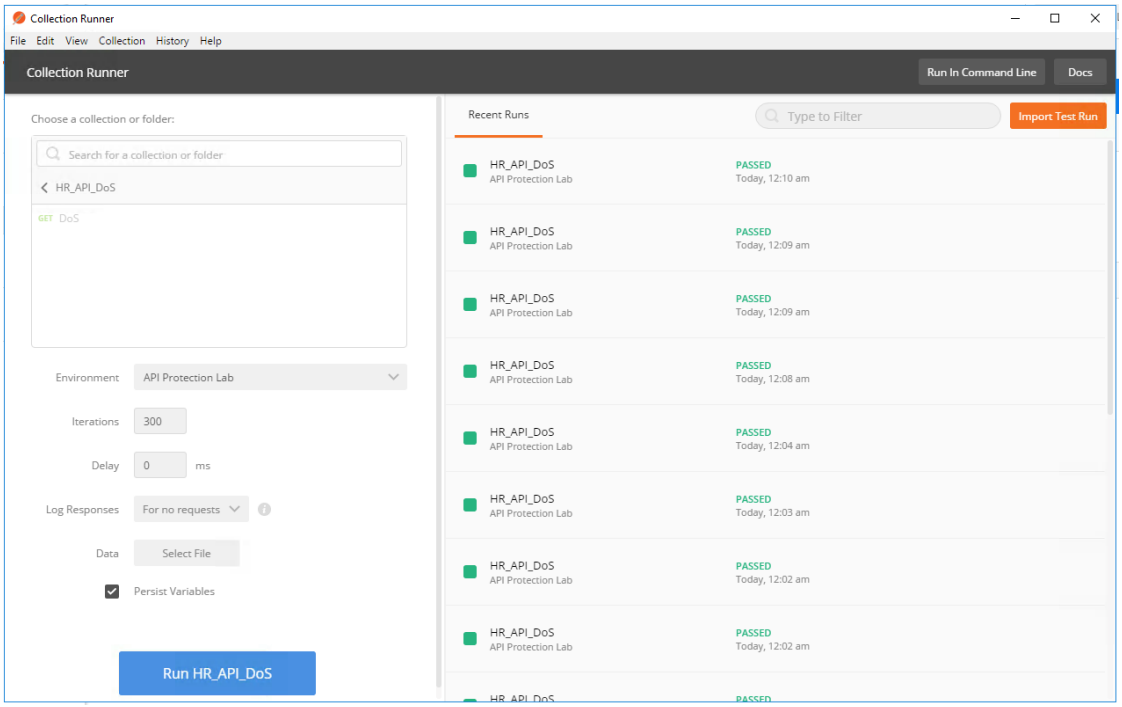
2.4.2 Running DoS Attack

In this task, you will simulate DoS attack with Postman Runner and observe how it's been mitigated by Application DoS profile.

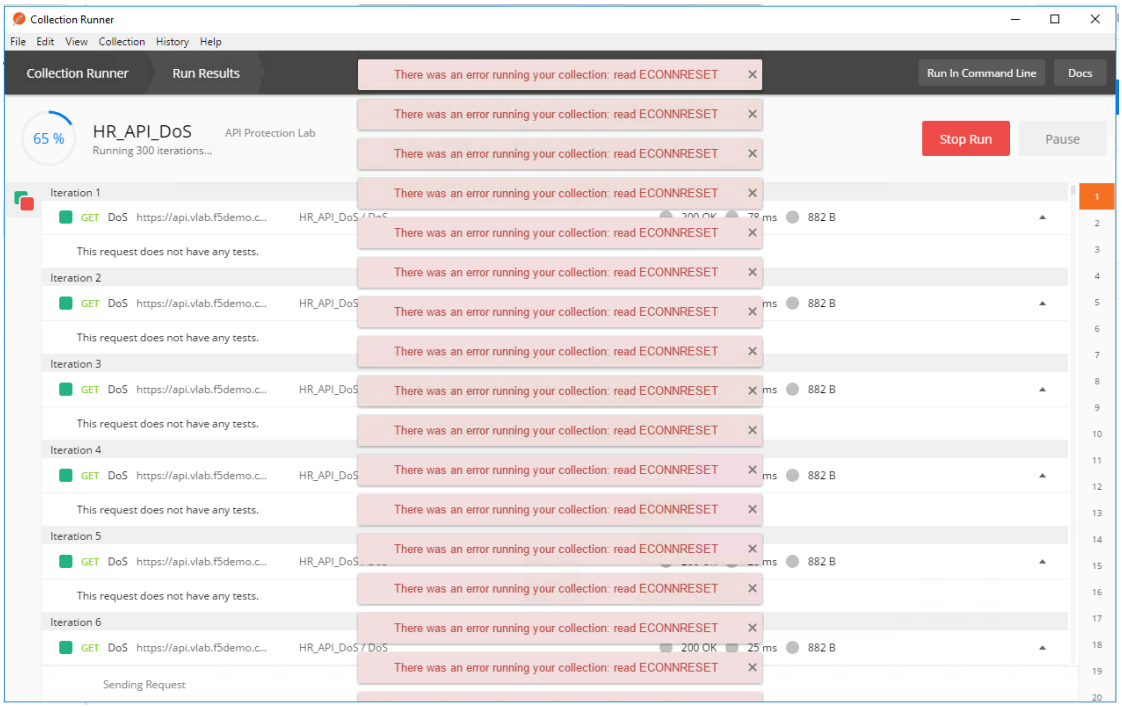
1. On the Windows client choose HR_API_DoS collection then DoS request.
2. Select token "employeeuser" from the list of available tokens and send the API call. Make sure you are getting response http code 200 and list of departments.
3. Click **Save** to save the request with new access token.
4. Click Runner on the top of the window:



5. Choose HR_API_DoS collection and use the screenshot below to configure parameters. Ensure right environment has been chosen, iterations updated and Log responses as “For no requests”. Once the parameters have been selected click Run HR_API_DoS.

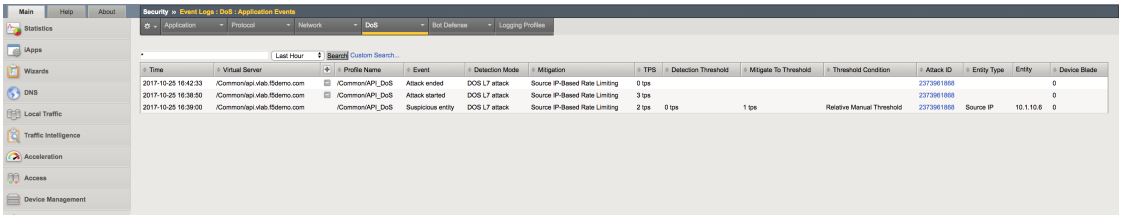


6. After a short period of time Postman Runner may report failing transactions (it may not and gracefully handle the rate limiting, proceed to check logs in next step anyway):



7. On the BIG-IP navigate to Security->Event Logs-> DoS -> Application Events

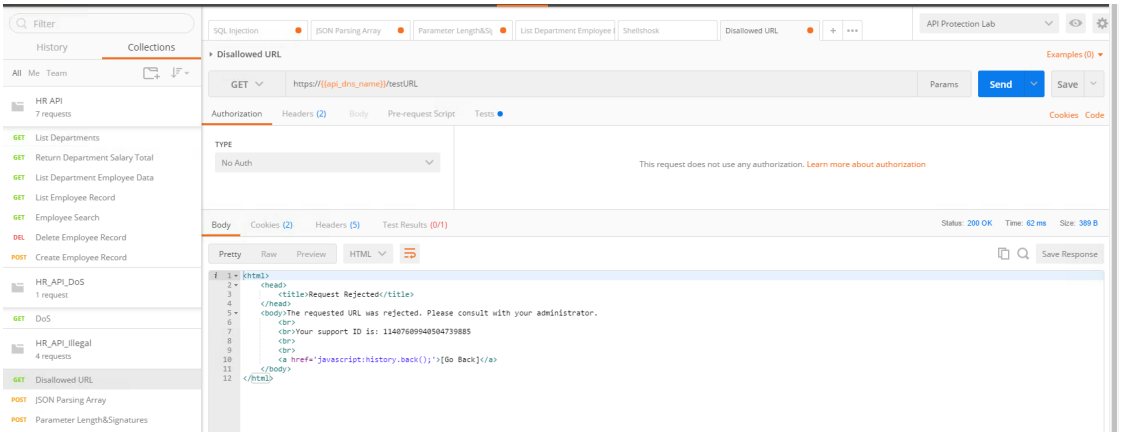
   Look at the details of the attack detection and mitigation applied.



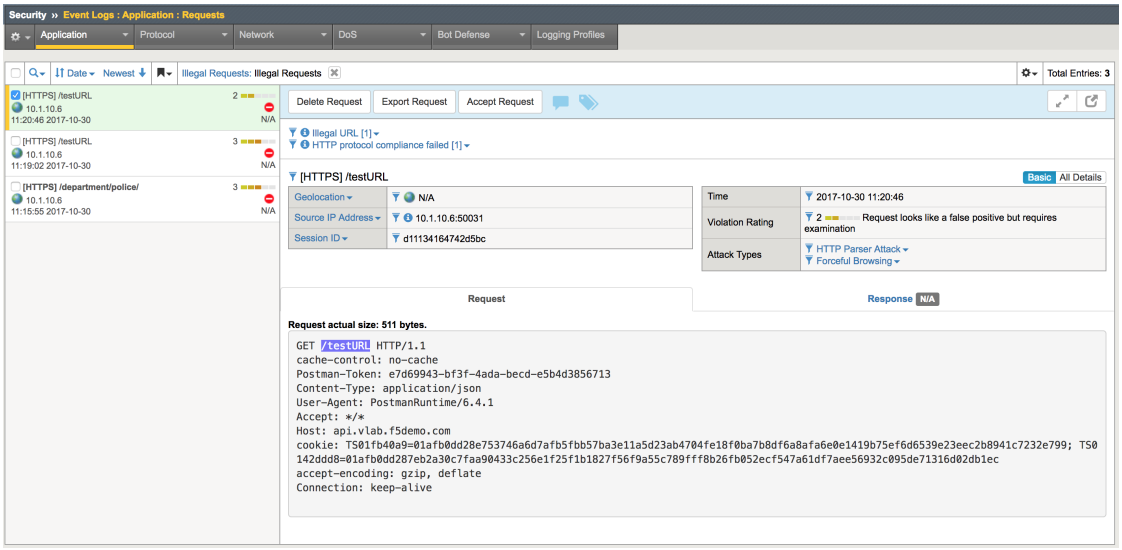
2.4.3 Accessing Disallowed URL

In this task, you will try accessing URL not allowed by ASM policy.

1. On the client machine Postman app proceed to “HR_API_Illigal” collection and select Disallowed URL request.
2. Use previously retrieved access token by setting Authorization Type to Oath 2.0 and Available Tokens to hruser (similar to Module 3 Task 5 without getting a new token).
3. Click Send. ASM should return a blocking page as below:



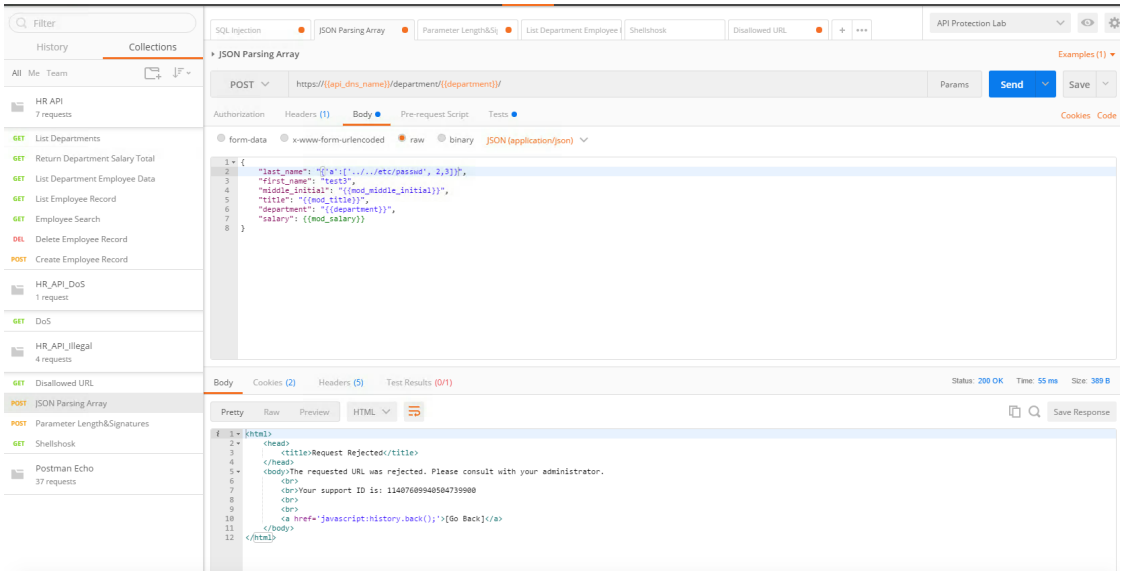
4. On the BIG-IP navigate to Security->Event Logs->Application ->Requests. Click on the last (top) request and look at the violation details:



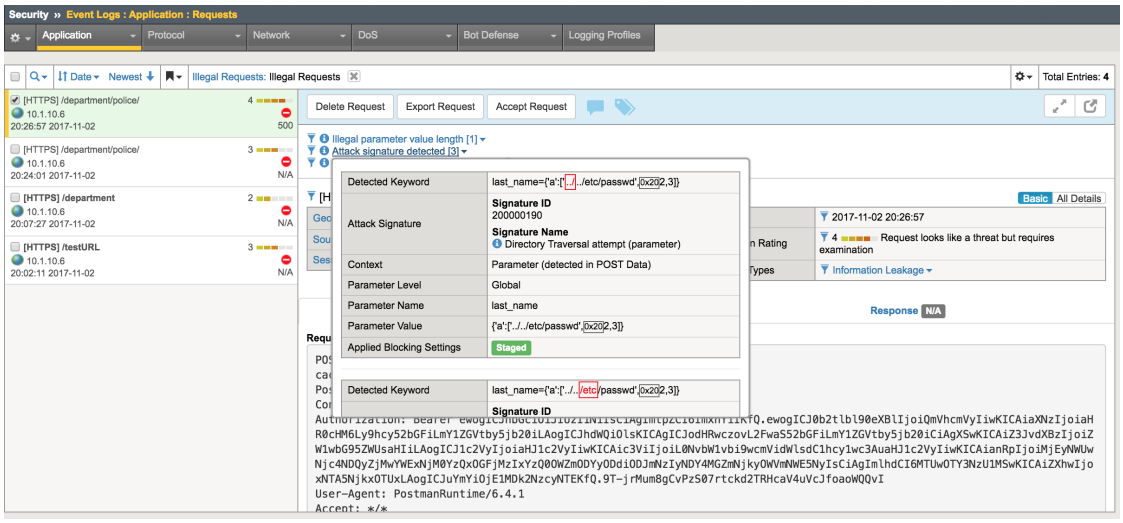
2.4.4 Illegal JSON parameter value - 1

In this task, you will simulate attack by running request with illegal JSON parameter.

1. On the client machine Postman app chose JSON Parsing Array request.
2. Use previously retrieved access token by setting Authorization Type to Oauth 2.0 and Available Tokens to hruser (similar to Module 3 Task 5 without getting a new token).
3. Click Send. ASM should return a blocking page as below:



4. On the BIG-IP navigate to Security->Event Logs->Application ->Requests. Click on the last (top) request and look at the violation details:

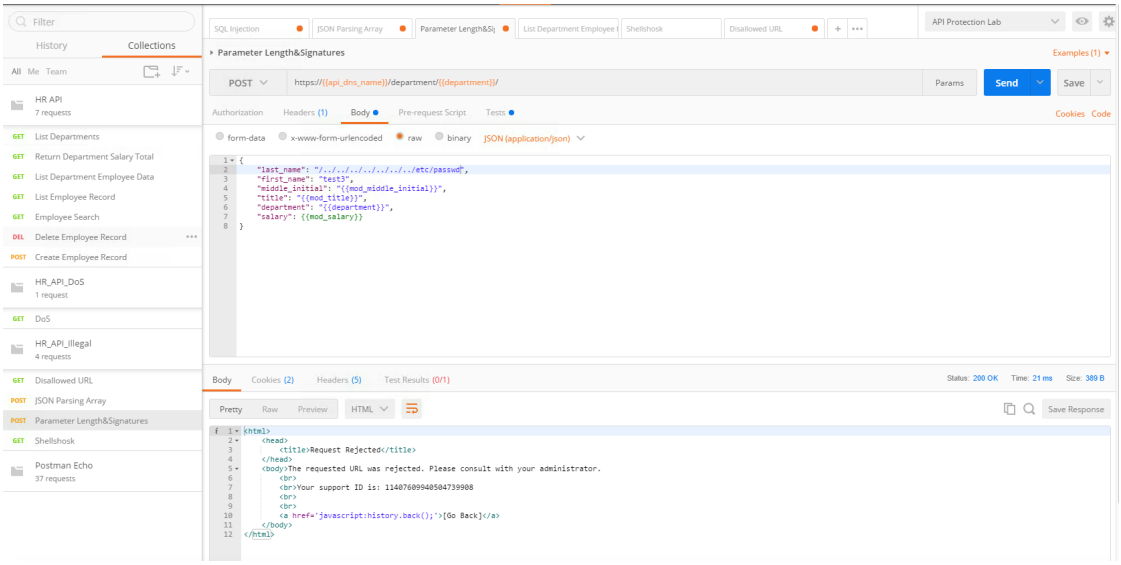


2.4.5 Illegal JSON parameter value - 2

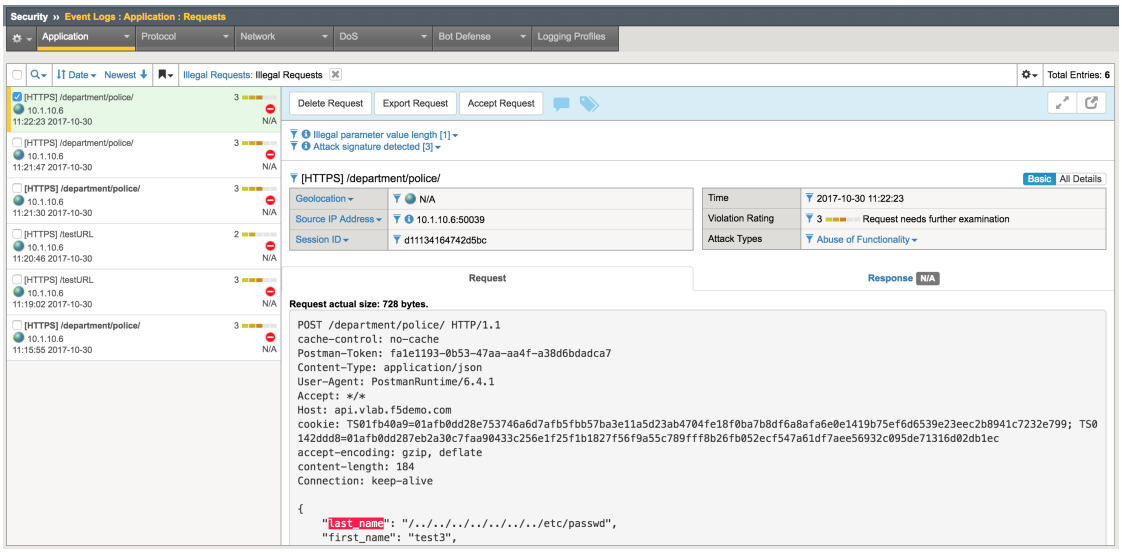
In this task, you will simulate another attack by running request with illegal JSON parameter.

1. On the client machine Postman app chose Parameter Length&Signatures request.

2. Use previously retrieved access token by setting Authorization Type to Oath 2.0 and Available Tokens to hruser (similar to Module 3 Task 5 without getting a new token).
3. Click Send. ASM should return a blocking page as below:



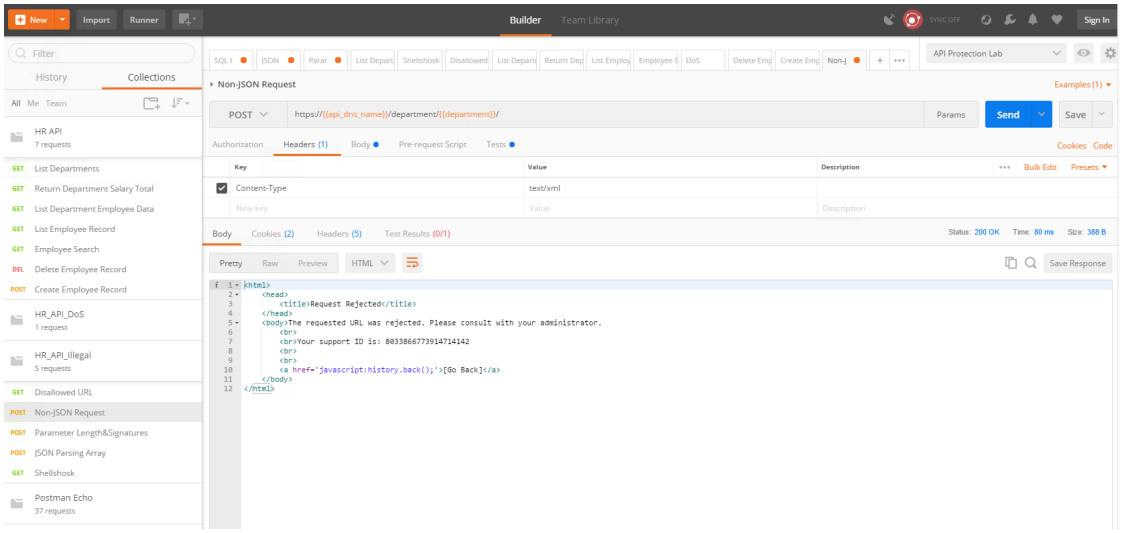
4. On the BIG-IP navigate to Security->Event Logs->Application ->Requests. Click on the last (top) request and look at the violation details:



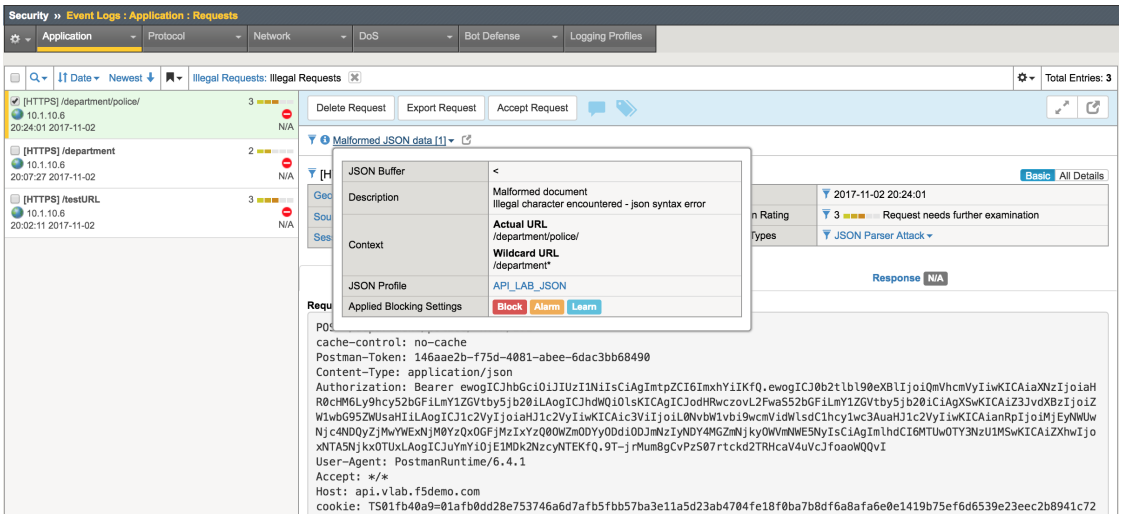
2.4.6 Non-JSON request

In this task, you will simulate attack by running non JSON request.

1. On the client machine Postman app chose Non-JSON request.
2. Use previously retrieved access token by setting Authorization Type to Oath 2.0 and Available Tokens to hruser (similar to Module 3 Task 5 without getting a new token).
3. Click Send. ASM should return a blocking page as below:



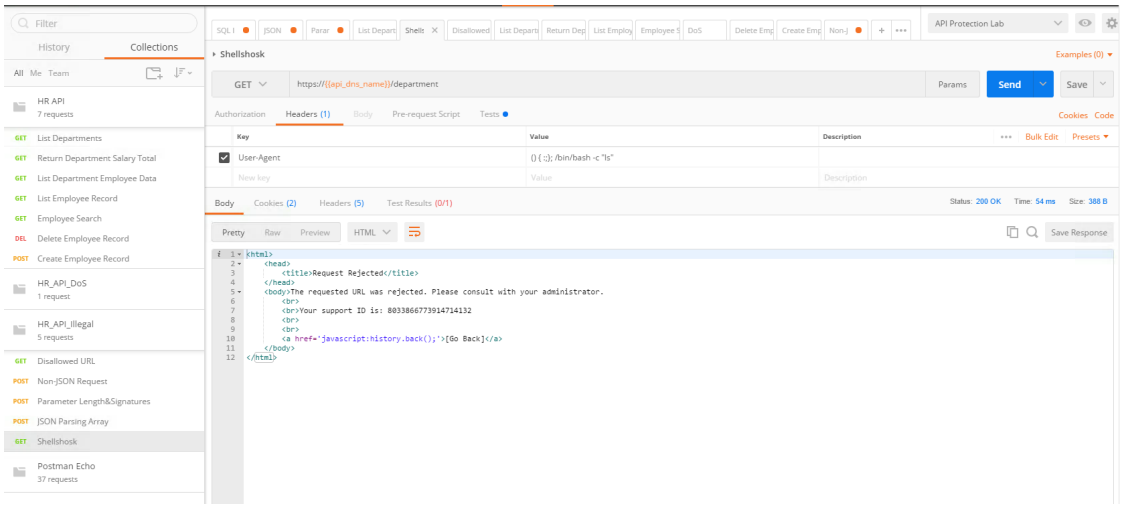
4. On the BIG-IP navigate to Security->Event Logs->Application ->Requests. Click on the last (top) request and look at the violation details:



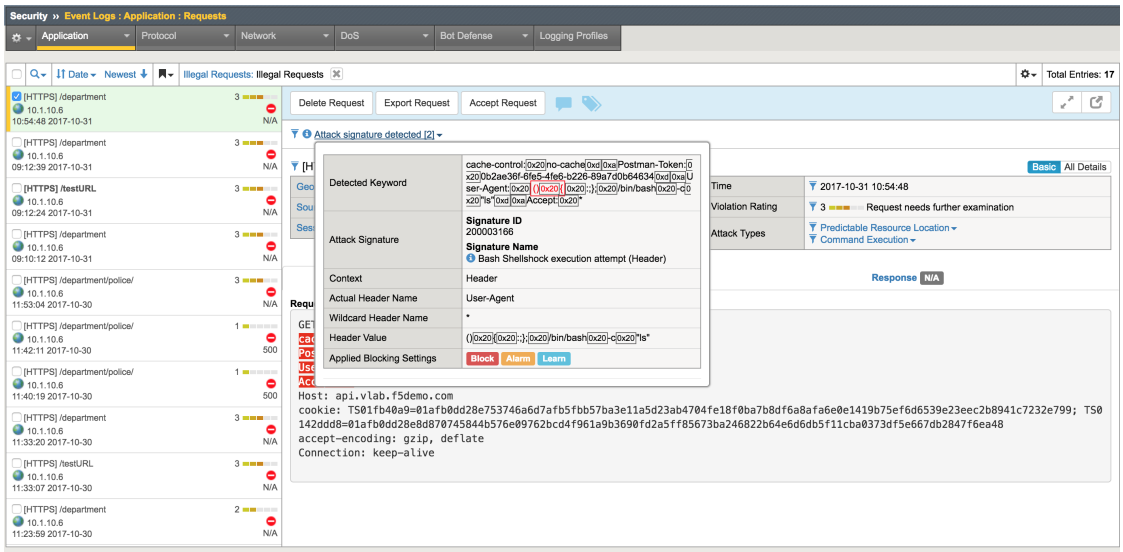
2.4.7 Shellshock request

In this task, you will simulate Shellshock attack by running request with specific header.

1. On the client machine Postman app chose Shellshock request.
2. Use previously retrieved access token by setting Authorization Type to Oath 2.0 and Available Tokens to hruser (similar to Module 3 Task 5 without getting a new token).
3. Click Send. ASM should return a blocking page as below:



1. On the BIG-IP navigate to Security->Event Logs->Application ->Requests. Click on the last (top) request and look at the violation details:



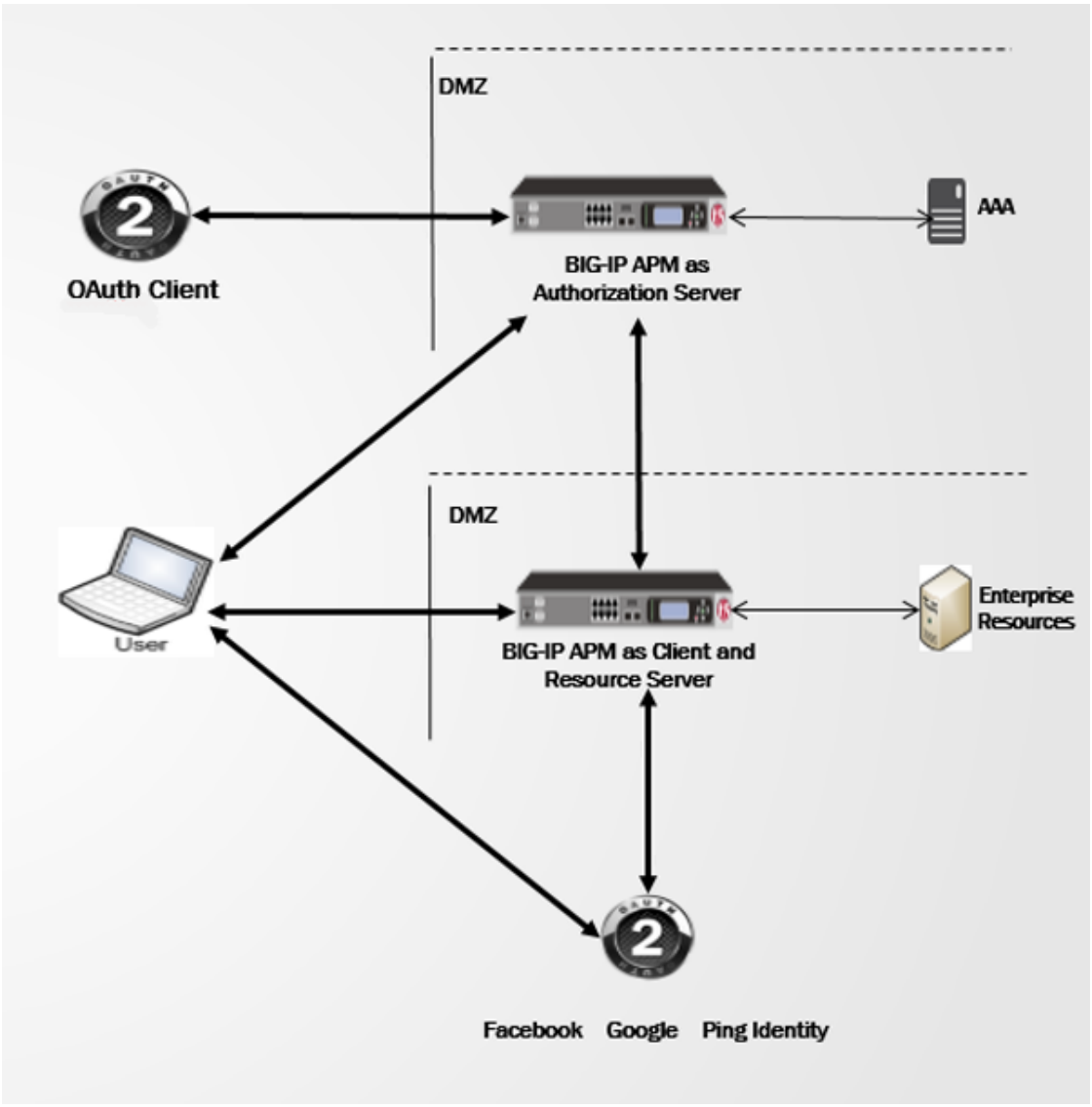
More information about BIG-IP configuration settings available here *F5 BIG-IP configuration deep dive*

3 F5 BIG-IP configuration deep dive

In this section you will find detailed information about BIG-IP configuration used in the lab exercises.

3.1 BIG-IP Authorization access control configuration

BIG-IP APM can be used as authorization server as well resource server. On the diagram below this is shown as separate hardware components.



Although hardware segregation is not a strict requirement and those components can be hosted on the same BIG-IP, only VIPs should be different. In this lab we have used **api.vlab.f5demo.com** as resource server and **as.vlab.f5demo.com** as an authorization server. Check the steps below in order to configure coarse-grain authorization.

3.1.1 Creation of JWK (JSON Web Key)

In this task you will check configuration settings for JWK which is used for validating the sent JWT.In this lab we have used Octet and a shared secret, but options includesolutions like public/private key pair as well.

Go to Access -> Federation -> JSON Web Token -> Key Configuration -> prebuilt-api-jwk

It is configured according to data below

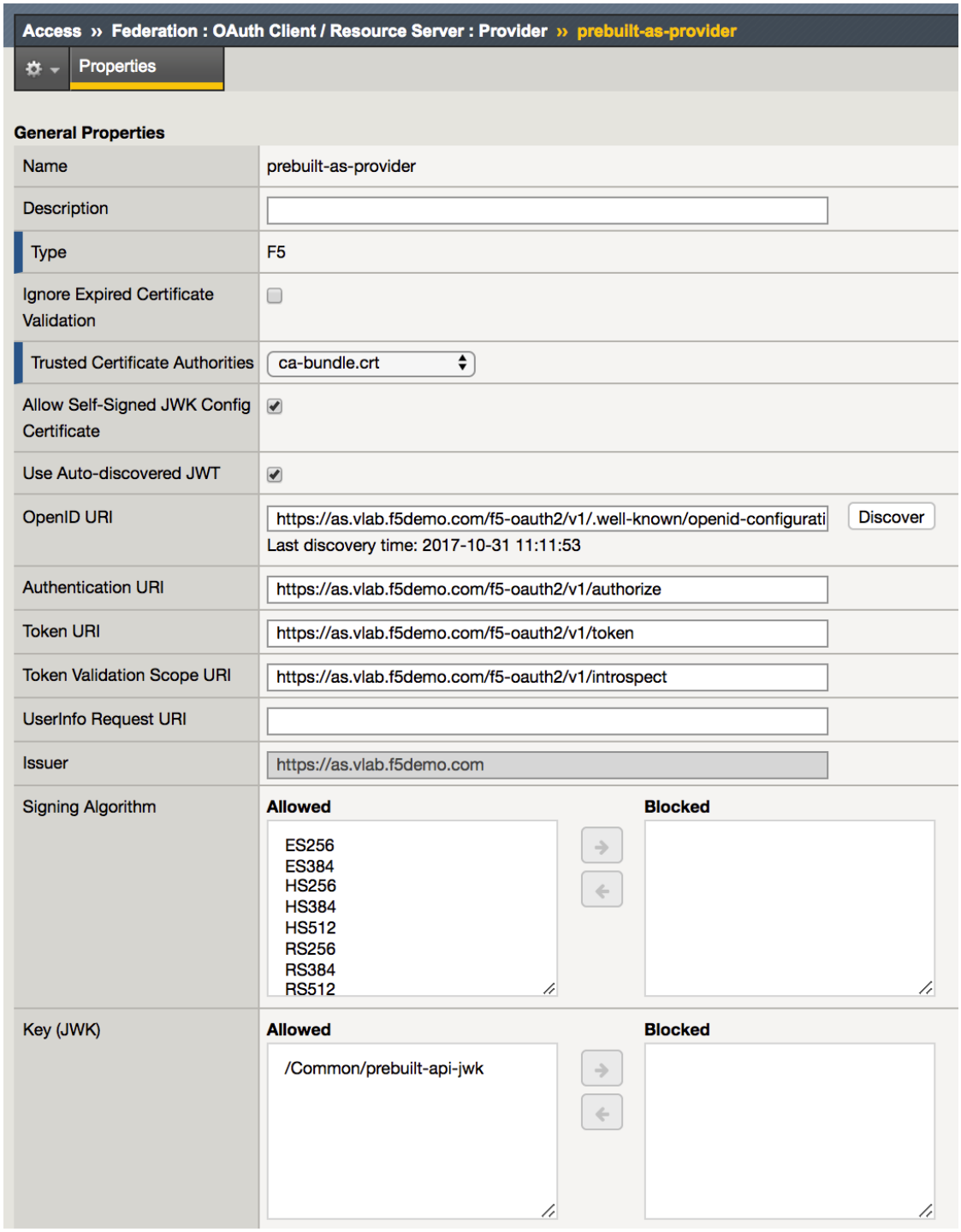
Field	Value
Name	api-jwk
ID	lab
Type	Octet
Signing Algorithm	HS256
Shared Secret	secret

3.1.2 OAuth provider

In this task you will check configuration settings for an OAuth provider so that JWT can be validated.

Go to Access -> Federation -> OAuth Client/Resource Server -> Provider -> prebuilt-as-provider

The configuration settings is shown below.

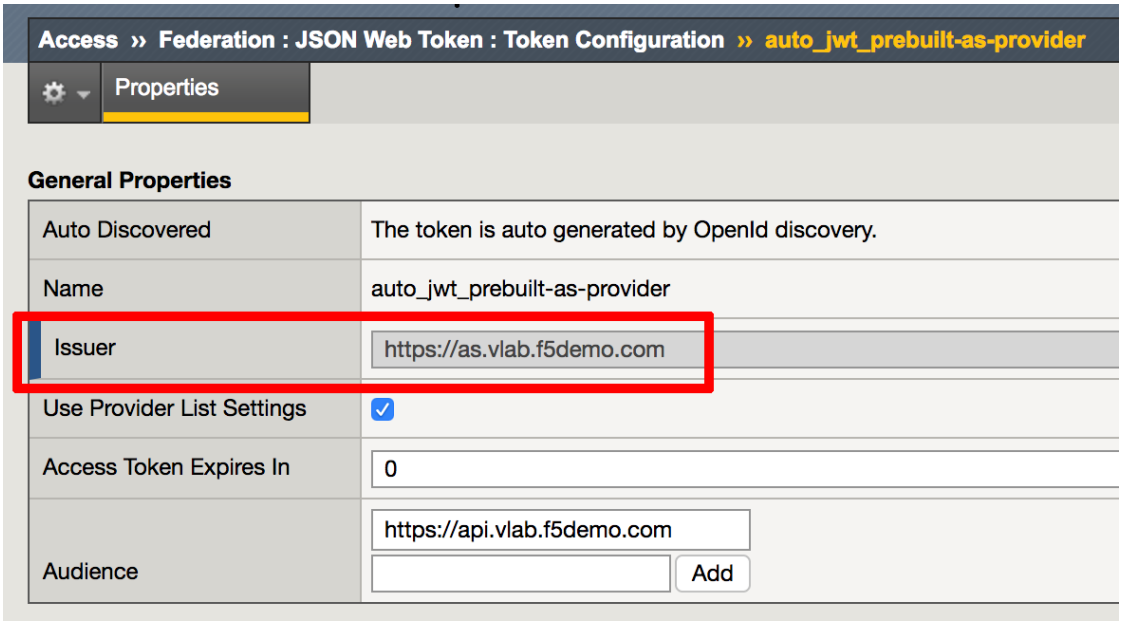


Most of these settings have been discovered automatically from Authorization Server with the use of OIDC.

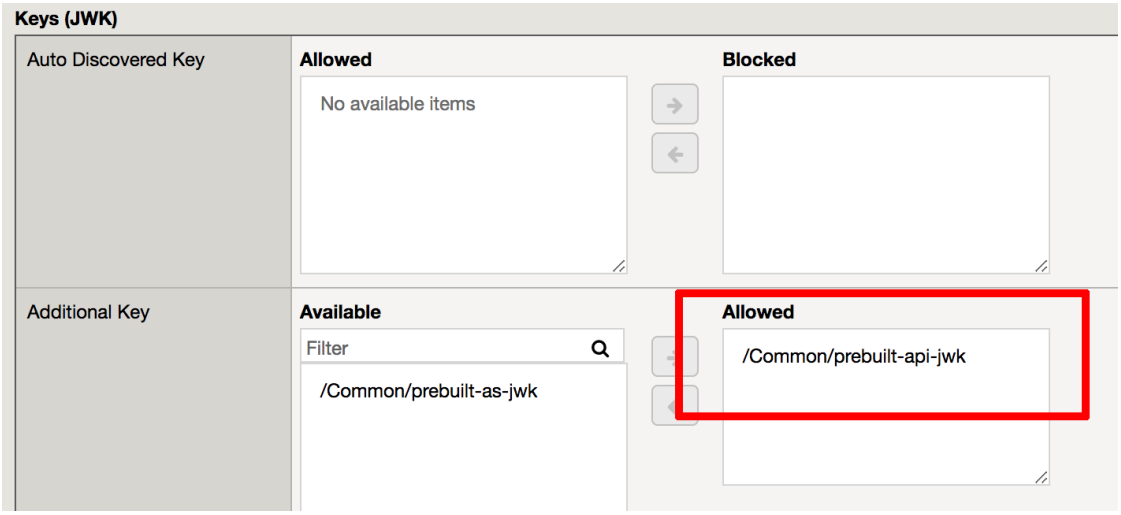
3.1.3 Token Configuration

In this task you will check the configuration of some of the values retrieved automatically via OIDC discover tool. Those values had to be adjusted manually because the OIDC AS cannot provide you with the values specific to your audience.

1. Go to Access -> Federation -> JSON Web Token -> Token Configuration -> auto_jwt_prebuilt-as-provider
2. Make sure **https://api.vlab.f5demo.com** have been defined as Issuer



3. Under Additional Key make sure prebuilt-api-jwk is added into Allowed



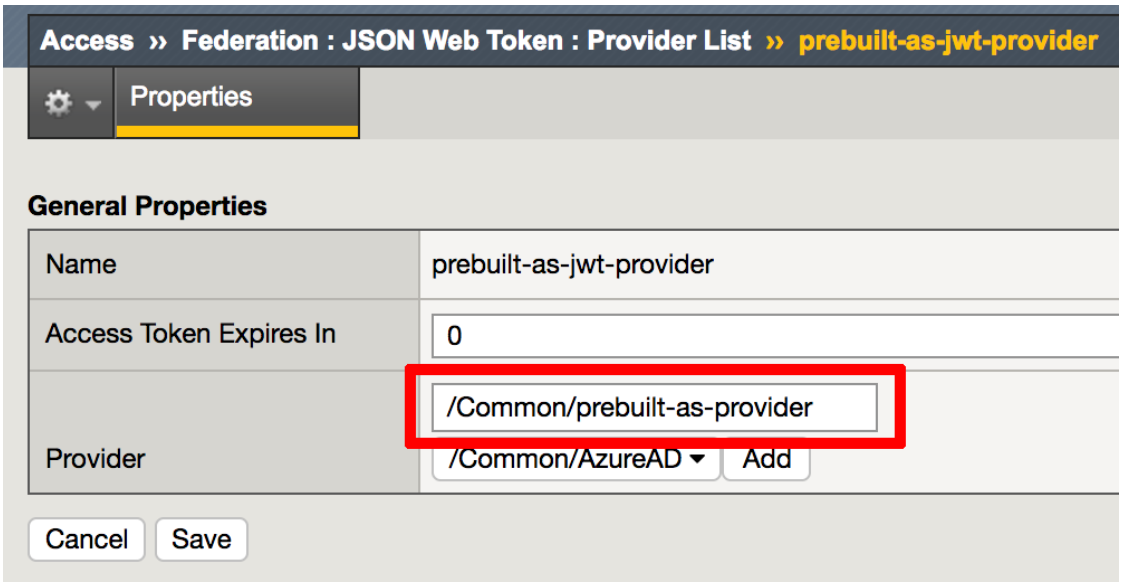
The object prebuilt-as-jwk was precreated for the authorization server function. It matches api-jwk (and prebuilt-api-jwk) exactly because a shared key is needed on both the authorization server and resource side. In this case you could have reused it instead of making a new one, but in production your authorization server may not be the Big-IP you are protecting the API server on, and you would need to create it as shown here.

3.1.4 JWT Provider

In this task you will check configuration for a JWT provider that can be selected in a per request or per session policy for JWT validation.

Go to Access -> Federation -> JSON Web Token -> Provider List -> prebuilt-as-jwt-provider

Make sure prebuilt-as-provider is selected

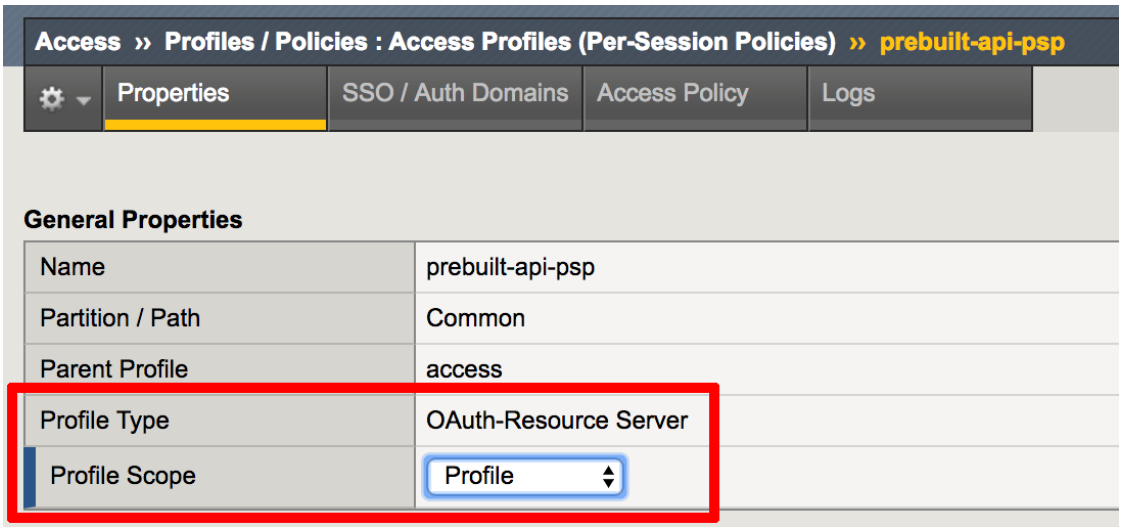


3.1.5 Per-session policy

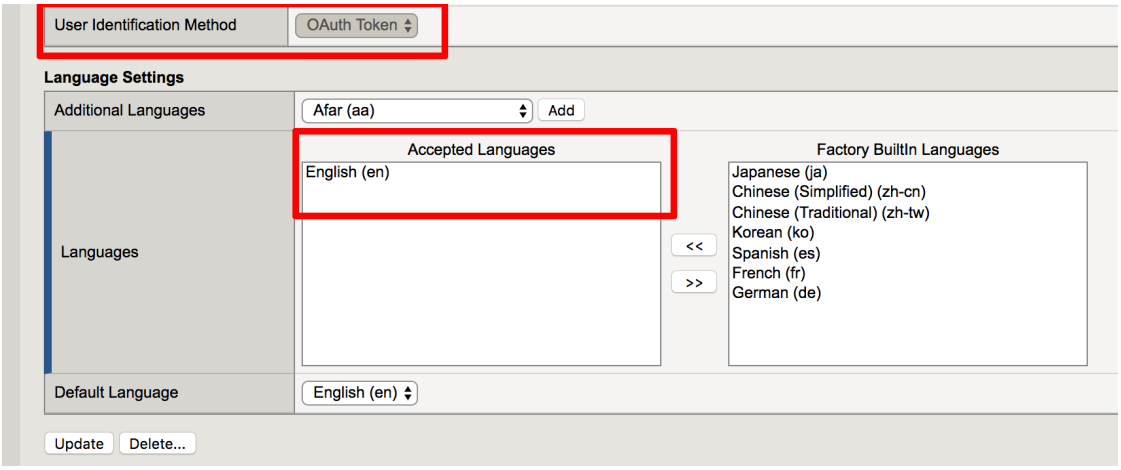
In this task you will check per session policy which validating the JWT token and collecting the claims data from parameters inside the JWT.

Go to Access -> Profiles/Policies -> Access Profiles (Per-Session Policies) -> prebuilt-api-psp

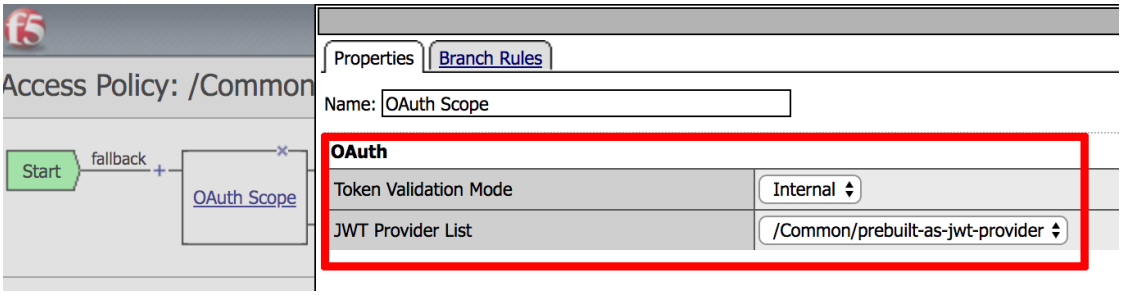
Make sure profile type is set to OAuth-Resource Server and Profile Scope is Profile.



Also make sure that the User Identification Method is set to OAuth token and default language is English.



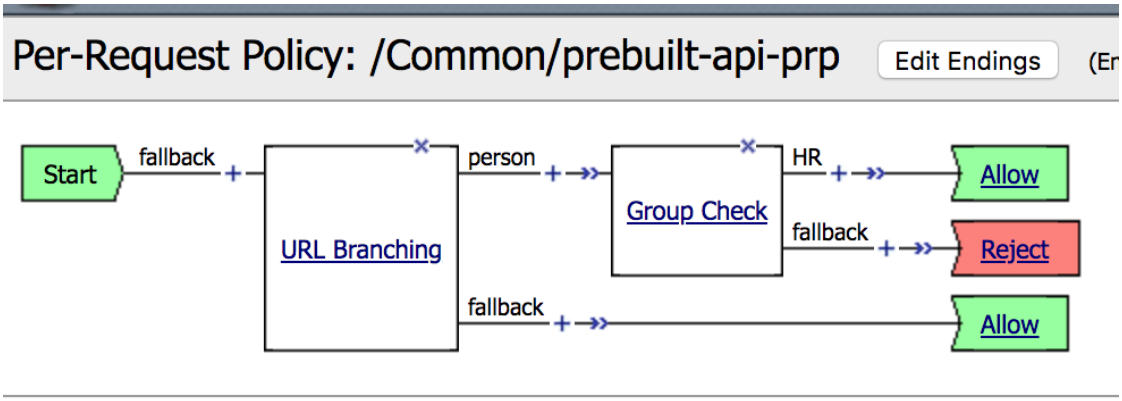
Click **Access Policy** tab and then click **Edit Access Policy for Profile "prebuilt-api-psp"**. Examine the policy and click on OAuth Scope. Make sure Token Validation Mode is set to "Internal" and JWT Provider List is set to as-jwt-provider. The validation mode is set to be internal because JWT token will be validated internally instead of making an external call.



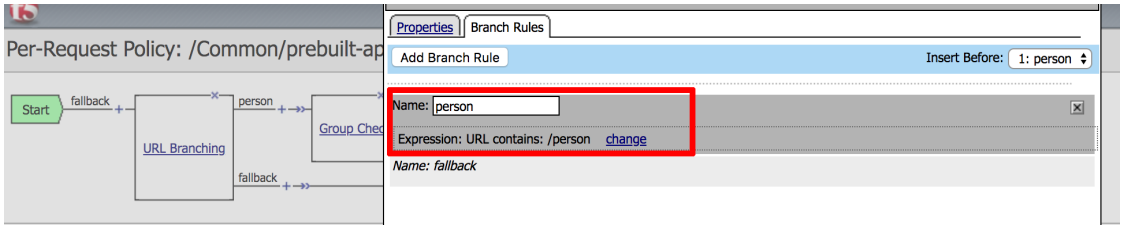
3.1.6 Per-request policy

You will check a per request policy to validate authorization on each request by checking for the presence and validity of a JWT.

Go to Access -> Profiles/Policies -> Per-Request Policies -> prebuilt-api-prp —> Click **Edit**



Click on **URL Branching** and and then **Branch Rules**. Note the URL expression.



Click on **Group Check** and and then **Branch Rules**. Note the expression.

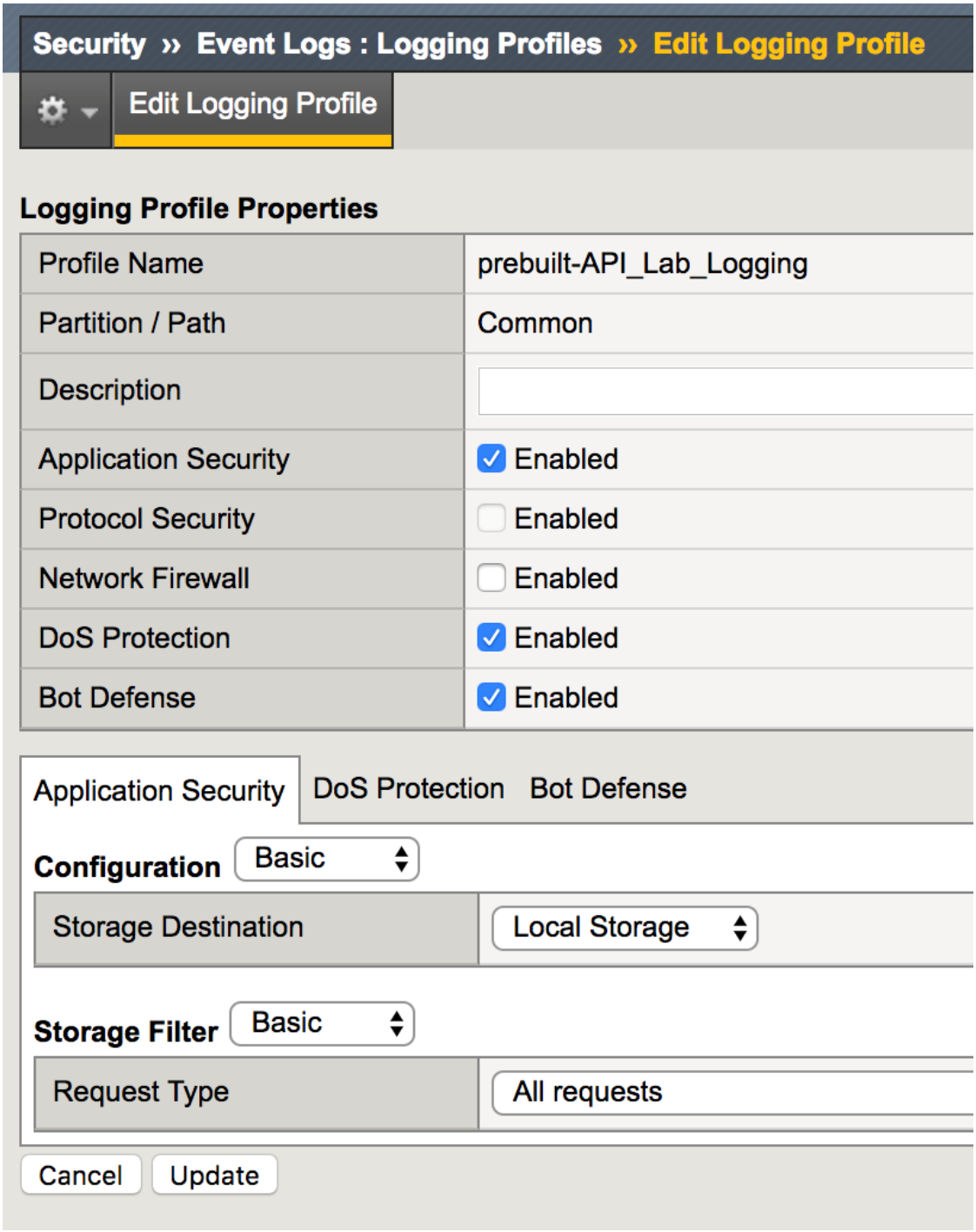
Close the tab in the browser

3.2 BIG-IP API requests protection

In this module you will examine security policies created for API server protection.

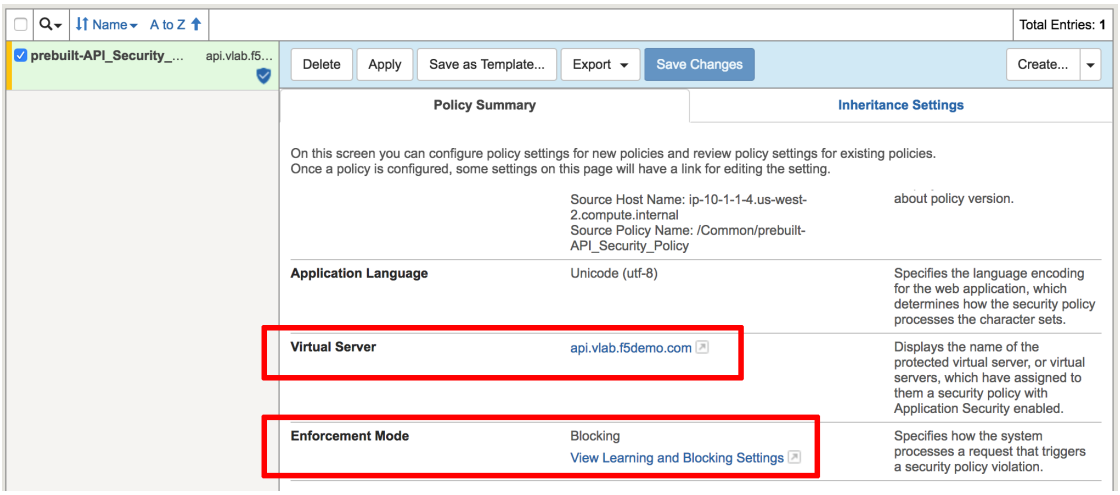
3.2.1 Logging profile

In order to check for logging configuration settings to Security -> Event Logs -> Logging Profiles —> prebuilt-API_Lab_Logging and check logging settings.

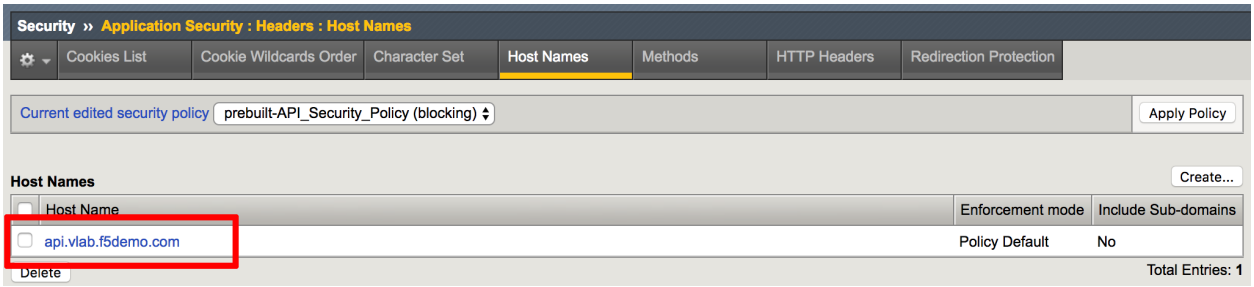


3.2.2 Application Security Policy

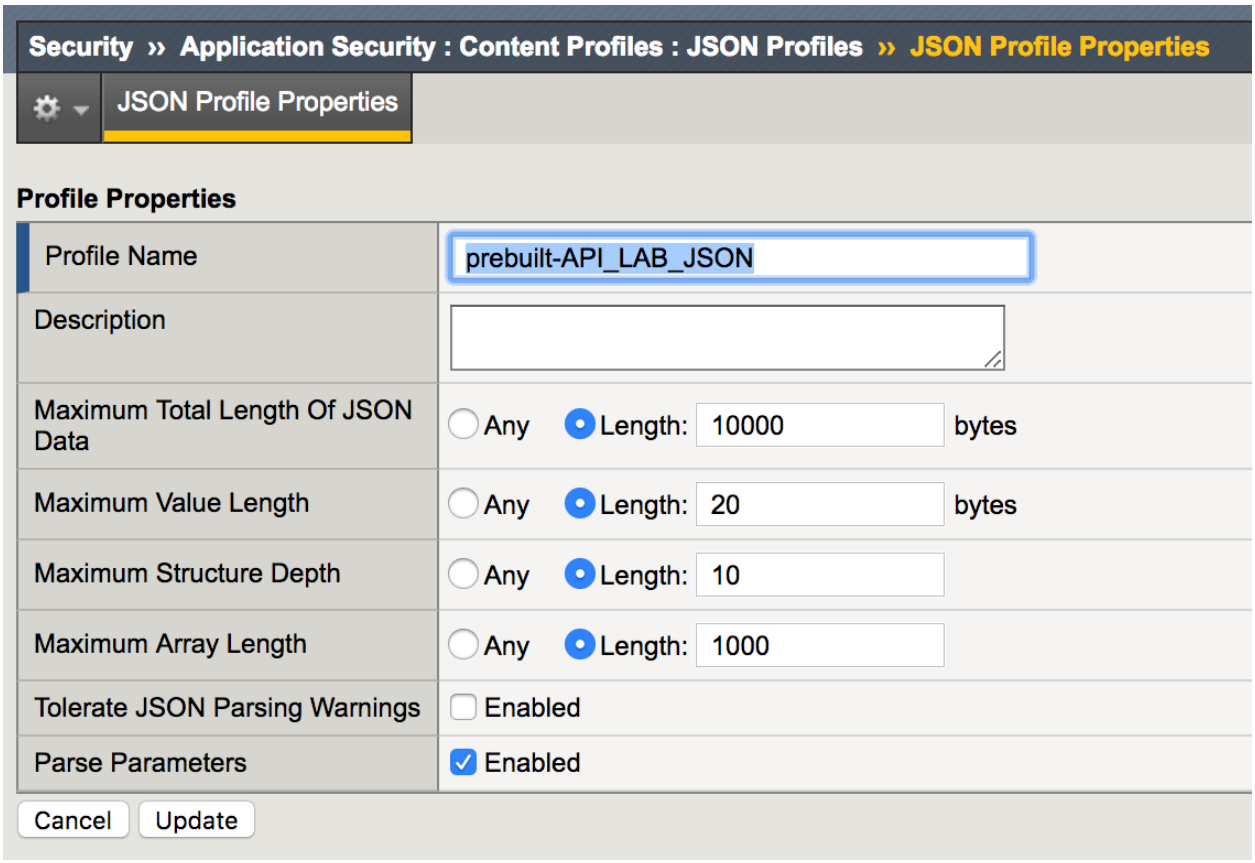
Go to Security -> Application Security -> Security Policies —> prebuilt-API_Security_Policy and make sure it is in blocking mode.



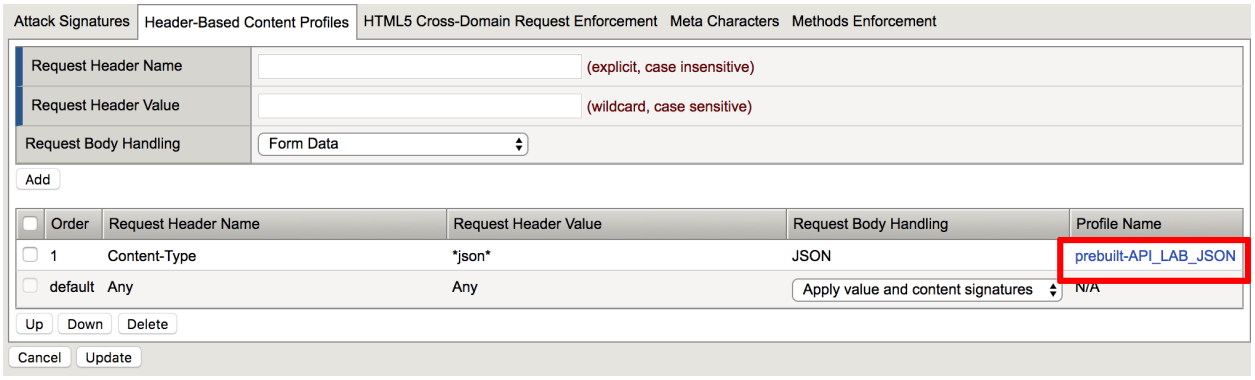
Check hostname configuration for requested secured by this policy. Go to Security -> Application Security -> Headers -> Hostnames and make sure api.vlab.f5demo.com is configured. Any request with different hostname will be blocked by the policy.



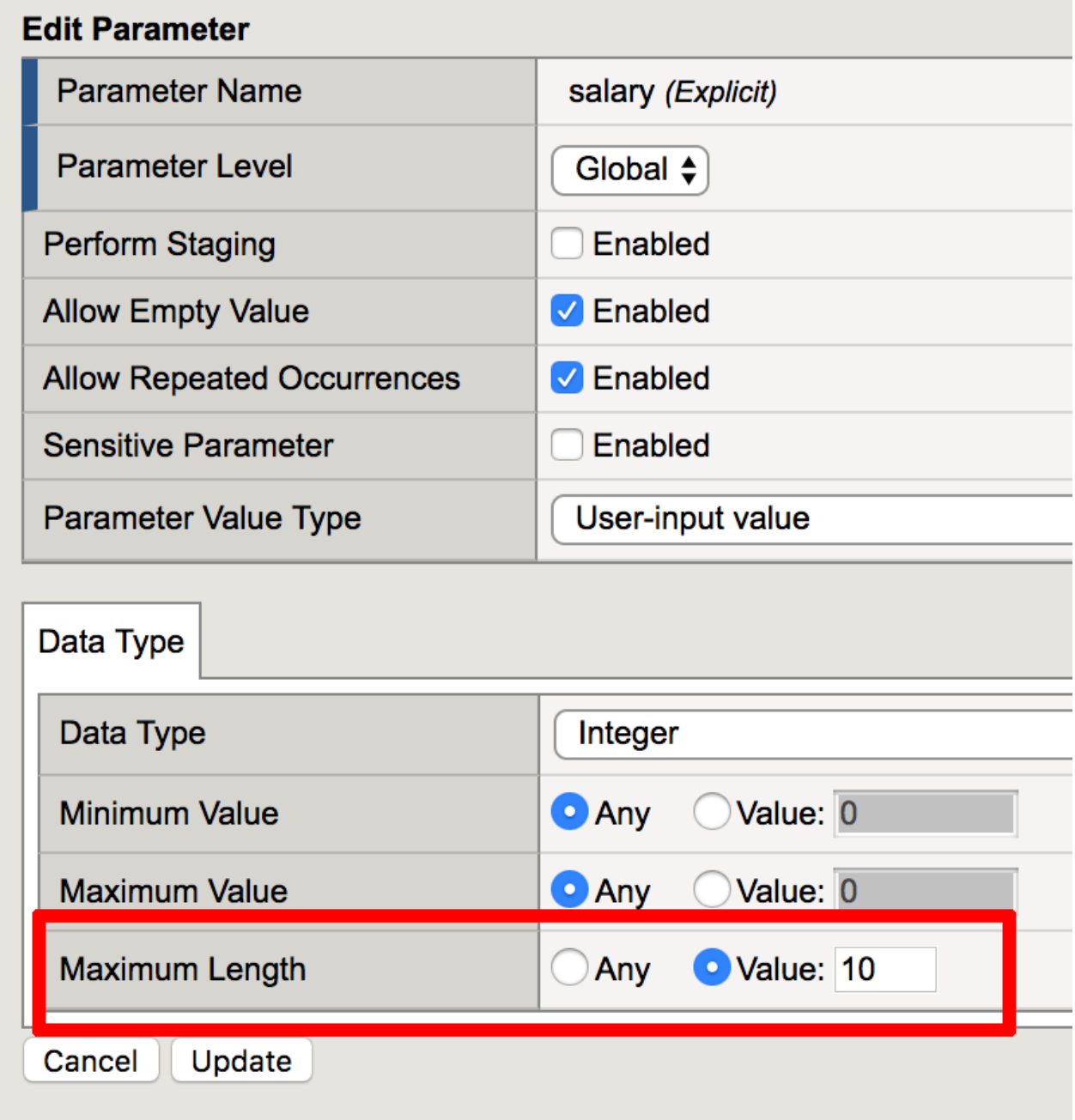
Check JSON profile configuration. Go to Security -> Application Security -> Content Profiles -> JSON Profiles —> prebuilt-API_LAB_JSON and check its settings.



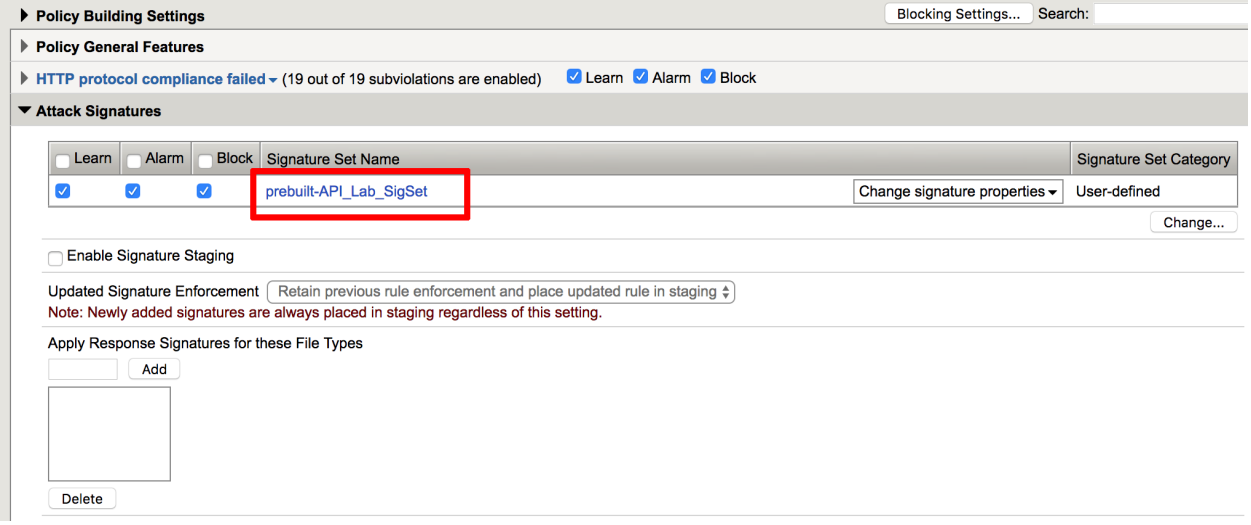
Check for allowed URLs in the security policy. Go to Security -> Application Security -> Allowed URLs -> Allowed HTTP URLs. Click on /**department*** and check Header-Based Content Profiles - Make sure prebuilt-API_LAB-JSON content profile is attached to this URL.



Check for allowed parameters in the security policy. Go to Security->Application Security->Parameters->Parameters List. Check length and other settings by clicking into each parameter. For example maximum length for parameter “salary” is 10.



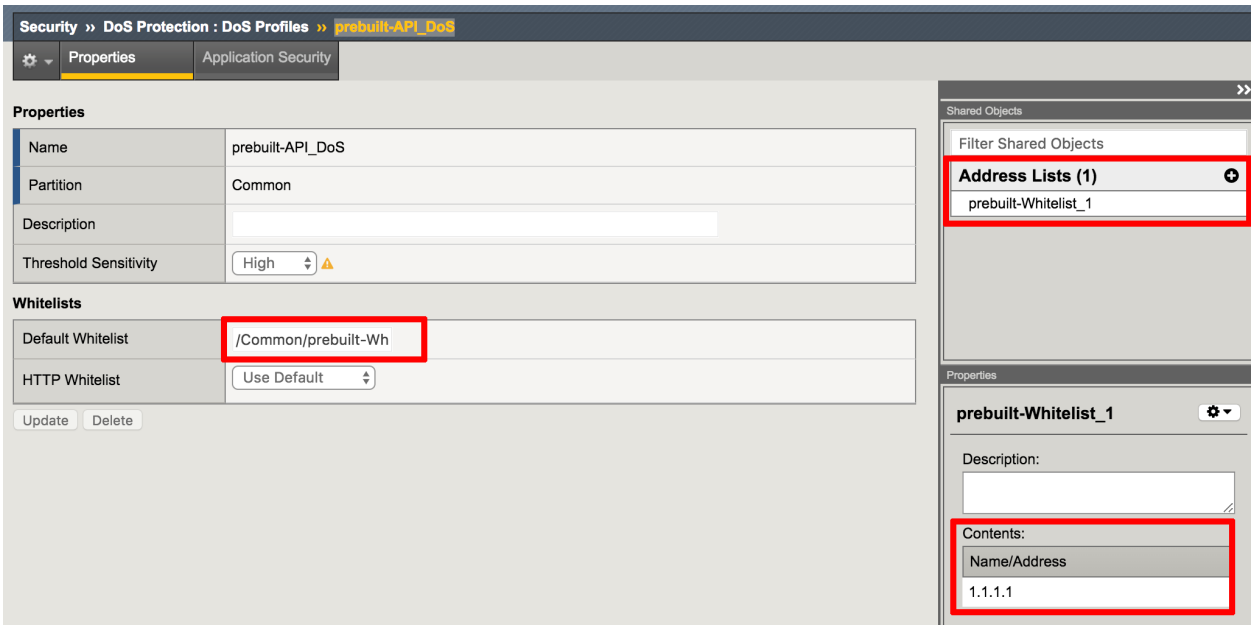
Check attack signatures configuration. Attack signature set is required to limit policy only to relevant signatures for the protected API. Go to Security -> Application Security -> Policy -> Policy Properties and click on Attack Signatures Configuration.



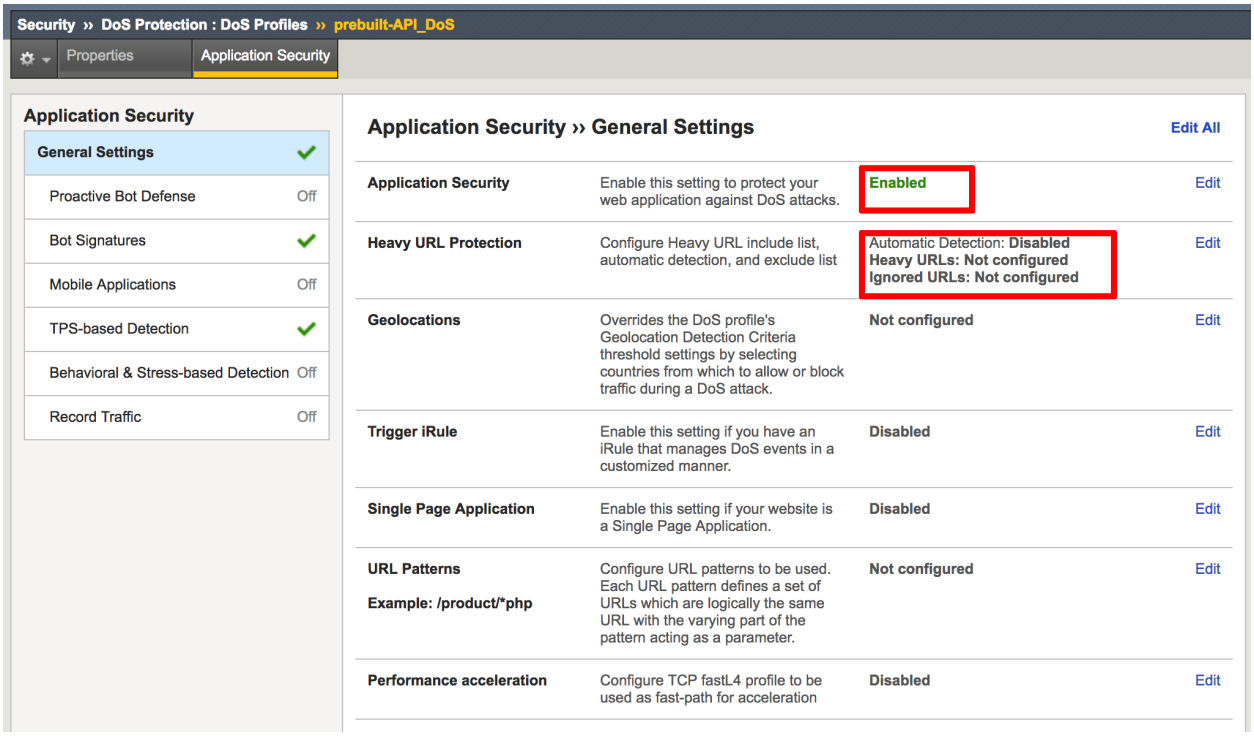
Click on prebuilt-API_Lab_SigSet, scroll down to “Signatures” section and examine the set of signatures.

3.2.3 API DoS Protection Profile

In this section you will check for configuration settings in DoS protection profile that will protect API interface against known bots and DoS attacks. Navigate to Security -> DoS Protection -> DoS Profiles and click on prebuilt-API_DoS profile. Check for the example of white list for known ip addresses.

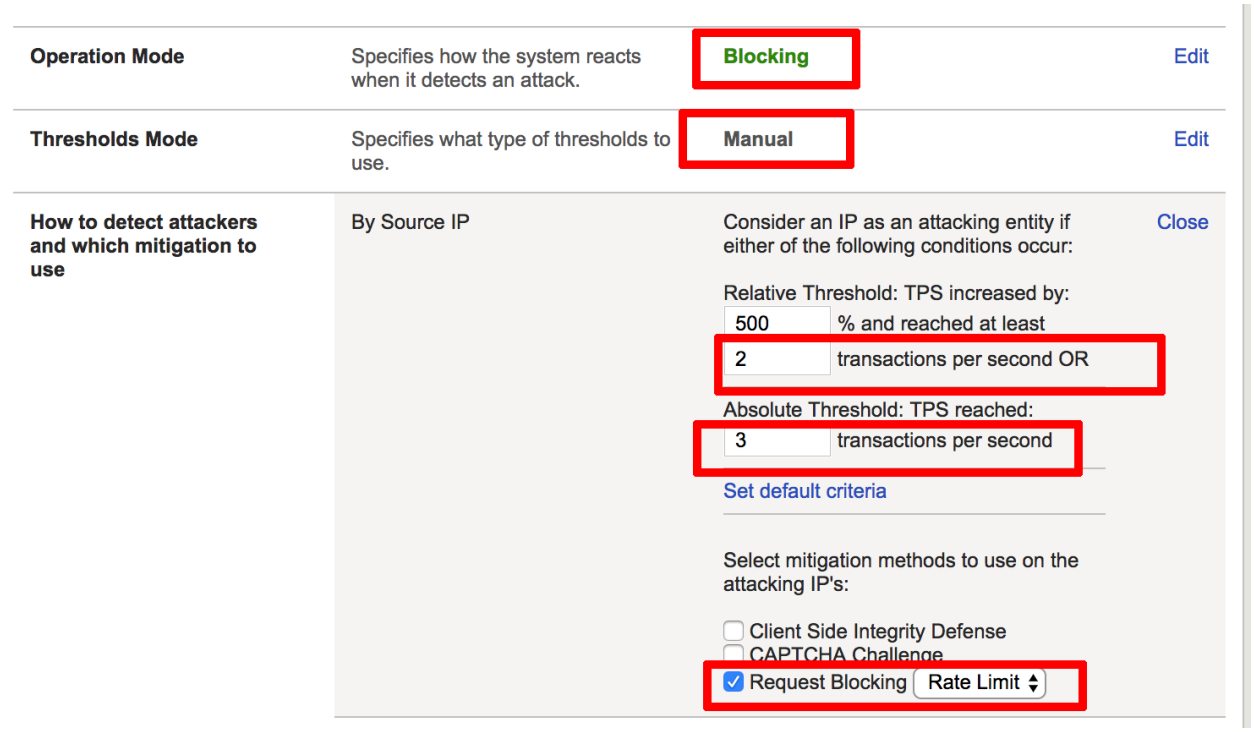


Click on **Application Security** tab and make sure its enabled and Heavy URL Protection is activated.



Go to **Bot Signatures** section, expand Bot Signature Categories and make sure Benign category is selected for blocking.

Go to **TPS-based Detection**, make sure it is in blocking mode and Thresholds Mode is set to Manual. Expand “How to detect attackers and which mitigation to use” section and check setting in subsection “By Source IP”.



Hint: More information on DoS Prevention available in the manual:

https://support.f5.com/kb/en-us/products/big-ip_asm/manuals/product/asm-implementations-12-1-0/1.html

F5 Networks, Inc. | f5.com

US Headquarters: 401 Elliott Ave W, Seattle, WA 98119 | 888-882-4447 // Americas: info@f5.com // Asia-Pacific: apacinfo@f5.com // Europe/Middle East/Africa: emeainfo@f5.com // Japan: f5j-info@f5.com