
IPRotate_Burp_Extension

Extension for Burp Suite which uses AWS API Gateway to change your IP on every request.

More info: [Bypassing IP Based Blocking Using AWS - Rhino Security Labs](#)

Description

This extension allows you to easily spin up API Gateways across multiple regions. All the Burp Suite traffic for the targeted host is then routed through the API Gateway endpoints which causes the IP to be different on each request. (There is a chance for recycling of IPs but this is pretty low and the more regions you use the less of a chance).

This is useful to bypass different kinds of IP blocking like bruteforce protection that blocks based on IP, API rate limiting based on IP or WAF blocking based on IP etc.

Usage

A version of this is available in the BApp store which you can install from there directly: <https://portswigger.net/bappstore>

With Python2 ENV set

1. Setup Jython in Burp Suite.
2. Ensure you have a set of AWS keys that have full access to the API Gateway service. This is available through the free tier of AWS.
3. Insert the credentials into the fields.
4. Insert the target domain you wish to target.
5. Select HTTPS if the domain is hosted over HTTPS.
6. Select all the regions you want to use, if you leave them all selected all valid regions will automatically be enabled.(The more you use the larger the IP pool will be)
7. Click "Enable".
8. Once you are done ensure you click disable to delete all the resources which were started.

If you want to check on the resources and endpoints that were started or any potential errors you can look at the output console in Burp.

Without Python2 ENV set (Advanced)

Use helper script for creating API GW in your AWS account. It requires [boto3](#) but it does not need to be setup for Burp, and you need to have valid AWS profile setup:

```
1 Usage: createapigws.py [OPTIONS]
2
3 Options:
4   --profile TEXT      AWS profile to use [default: pentest1]
5   --state-file TEXT   API GW state directory, script creates STATE_FILE
                        and
6                       STATE_FILE.json. [default: api_gateways.txt]
7   --create TEXT       specify target URL: https://example.com
8   --delete
9   --help              Show this message and exit.
```

1. Setup Jython in Burp Suite.
2. Ensure you have a set of AWS keys and profile setup that have full access to the API Gateway service. This is available through the free tier of AWS.
3. Insert state file path from `createapigws.py` to API GW File.
4. Insert the target domain you wish to target.
5. Insert the stage name to Stage name if required.
6. Select HTTPS if the domain is hosted over HTTPS.
7. Click "Enable".
8. Once you are done ensure you click disable.
9. To delete you API GWs use `createapigws.py` script.

The Burp UI

Access Key:

Secret Key:

API GW file:

Stage name:

Target host:

Target Protocol:

☐ HTTP

☒ HTTPS

Regions to launch API Gateways in:

<input checked="" type="checkbox"/> us-east-1	<input checked="" type="checkbox"/> us-west-1	<input checked="" type="checkbox"/> us-east-2
<input checked="" type="checkbox"/> us-west-2	<input checked="" type="checkbox"/> eu-central-1	<input checked="" type="checkbox"/> eu-west-1
<input checked="" type="checkbox"/> eu-west-2	<input checked="" type="checkbox"/> eu-west-3	<input checked="" type="checkbox"/> sa-east-1
<input checked="" type="checkbox"/> eu-north-1		

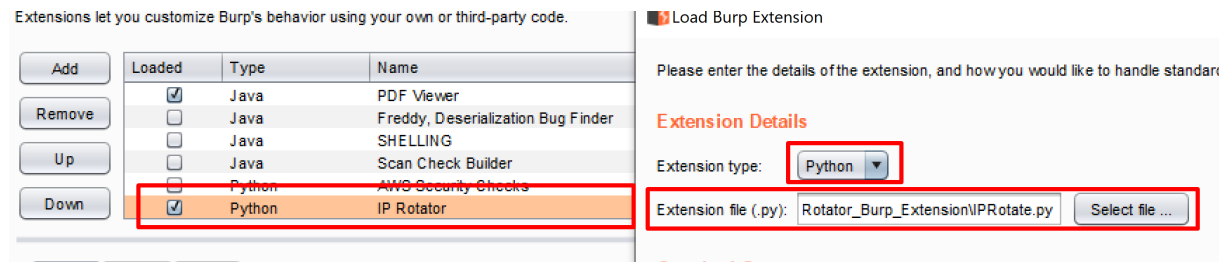
ENABLED

Example of how the requests look

```
Serving HTTP on 0.0.0.0 port 80 ...
10.69.69.3 - - [02/Aug/2019 08:48:17] "GET /test HTTP/1.1" 200 -
10.69.69.3 - - [02/Aug/2019 08:48:21] "GET /test HTTP/1.1" 200 -
10.69.69.3 - - [02/Aug/2019 08:48:22] "GET /test HTTP/1.1" 200 -
10.69.69.3 - - [02/Aug/2019 08:48:23] "GET /test HTTP/1.1" 200 -
10.69.69.3 - - [02/Aug/2019 08:48:23] "GET /test HTTP/1.1" 200 -
10.69.69.3 - - [02/Aug/2019 08:48:24] "GET /test HTTP/1.1" 200 -
108.128.161.229 - - [02/Aug/2019 08:59:23] "GET /test HTTP/1.1" 200 -
108.128.162.195 - - [02/Aug/2019 08:59:42] "GET /test HTTP/1.1" 200 -
3.123.14.136 - - [02/Aug/2019 08:59:47] "GET /test HTTP/1.1" 200 -
3.216.144.206 - - [02/Aug/2019 08:59:49] "GET /test HTTP/1.1" 200 -
3.15.35.122 - - [02/Aug/2019 08:59:55] "GET /test HTTP/1.1" 200 -
13.52.201.225 - - [02/Aug/2019 08:59:58] "GET /test HTTP/1.1" 200 -
34.223.68.76 - - [02/Aug/2019 09:00:10] "GET /test HTTP/1.1" 200 -
```

Setup

Make sure you have Jython installed and add IPRotate.py through the Burp Extension options.



Previous Research

After releasing this extension it was pointed out that there has been other research in this area using AWS API Gateway to hide an IP address. There is some awesome research and tools by @ustayready @ryHanson and @rmikehodes using this technique.

Be sure to check them out too:

- fireprox
- hideNsneak