Securing OAuth Token with HMAC Validation

Implementing security measures in order to prevent the possible attacks is a need in using enterprise software. Keyed-Hash Message Authentication Code (HMAC) validation is such measure which involved a cryptographic hash function and used to verify both the data integrity and authentication of a Message as with any Message Authentication code. In this tutorial you will use the HMAC to validate the OAuth tokens created in WSO2 API Manager and and WSO2 Identity Server.

- Preventing miss-use of OAuth Tokens
- WSO2 IS Extension - OAuth Token Generator Extension
- WSO2 API Manager extension - HMAC and timestamp verification handler

Preventing miss-use of OAuth Tokens

In API Manager, the main use case of HMAC is preventing miss-use of expired OAuth tokens or randomly generated OAuth tokens. Stolen or randomly generated tokens can be used to employ DOS/DDOS attacks effectively.

If an attacker uses random tokens to send API requests, API Manager will try to verify the token and it will hit through the critical path of verification. This is a costly transaction and it can cause high latencies and instability in API Manager clusters. Implementation of this particular solution is done using extensions developed for standard extension points of WSO2 API Manager and WSO2 Identity Server.

WSO2 IS Extension - OAuth Token Generator Extension

Engage the HMAC OAuth handler in order to do the Keyed-Hash Message Authentication Code (HMAC) validation by adding following into <IS_HOME>/repository/conf/identity/identity.xml

```
<IdentityOAuthTokenGenerator>com.sample.lahiru.wso2.hmac.oauth</IdentityOAuthTokenGenerator>
```

More information on developing OAuth token generator extensions here. Code for this particular solution can be found in oauth-hmac-extension GitHub repository.
This extension is responsible for enhancing the OAuth token with HMAC (Hash-based Message Authentication Code), so that above mentioned attacks will be less effective. Following two parts will be added to the token in addition to the default token created in WSO2 IS.

- HMAC
- Expiry timestamp

The format of the access token will be as follows thereafter. The token has 3 parts, delimited by ".".

Part I—original access token issued from WSO2 Identity Server

Part II—Hex value for token expiry time

Part III—HMAC calculation of ('Part I' + '.' + 'Part II')

**Access token format : <Part I>.<Part II>.<Part III>**

Example : ba13cf743c0b970ae6e8b6097364.000001e2ebabde.67830f288256eb80faa9dab85c3d29be7db1

### WSO2 API Manager extension - HMAC and timestamp verification handler

You can engage this handler by adding following entry before `#foreach($handler in $handlers)` line of `velocity_template.xml` file located in `<AM_HOME>/repository/resources/api_templates/` directory.

```xml
<handler class="com.sample.lahiru.wso2.hmac.handler.HMACTokenValidatorHandler"/>
```

Refer [Writing Custom Handlers](#) for understanding how to develop and engage WSO2 API handler extensions. Find the code for APIM handler in GitHub in [oauth-hmac-extension](#).

This custom handler verifies the HMAC of the token before it tries to authenticate using default authentication handler, which will be an expensive operation usually. It will also make sure the token is not expired. These verification will avoid any API calls to WSO2 API Manager, in case of the token is expired or HMAC is invalid.

HMAC validation handler calculates the HMAC using Part I and Part II (See Access token format), extracted from the token and validates by comparing that value with HMAC value included in the token (Part III).