

OpenID Connect Back-Channel Logout use case for Security Event Token (SET)



Michael B. Jones
IETF 97, Seoul
November 2016

OpenID Connect Back-Channel Logout



- http://openid.net/specs/openid-connect-backchannel-1_0.html
- Uses server-to-communication not using the browser
- Can be used by native applications, which have no active browser
- Sends a logout token, which is a Security Event Token (SET)
 - Signal from Identity Provider to Relying Party to perform logout

Logout Token Claims



- iss - Issuer Identifier
- sub - Subject Identifier
- aud - Audience(s)
- iat - Issued at time
- jti - Unique identifier for the token
- events – SET events array claim
 - First value is
<http://schemas.openid.net/event/backchannel-logout>
- sid - Session ID - String identifier for a Session (optional)

Example Logout Token Claims



- JWT Claims Set of a Logout Token:

```
{  
  "iss": "https://server.example.com",  
  "sub": "248289761001",  
  "aud": "s6BhdRkqt3",  
  "iat": 1471566154,  
  "jti": "bWJq",  
  "sid": "08a5019c-17e1-4977-8f42-65a12843ea02",  
  "events": [ "http://schemas.openid.net/event/backchannel-logout" ]  
}
```

Notes from this Use Case



- Does not use event-specific data payload
- Normal top-level OpenID Connect Issuer, Subject, & Session ID claims sufficient to identify target of event
- “events” claim value identifies the JWT as a logout token

- Does not use a special event transport
 - HTTP POST works great for this use case

Final Thoughts on SET



- SET is currently simple
 - Defines a required “events” claim and one other optional claim
 - A great match for this and other use cases
- Simplicity gives flexibility leading to adoption
 - Just like happened with JWTs