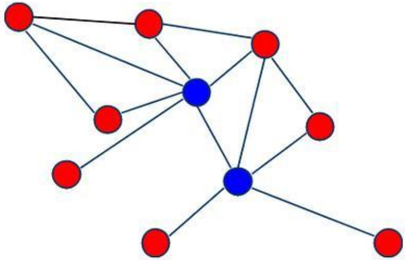
Insecuring your Data Using Federated Authentication

Clay Cooper, Identity and Access Management, Rochester Institute of Technology
(Not an expert on SAML or OpenID Connect/OIDC)

Abstract

Federated authentication is the gateway to zero-trust networking and access to the explosion of SaaS products that contain your company's data. How do you know that those products are correctly handling your single sign-on responses? We'll talk about the

- common SSO protocols and how they work
- how to test that they're configured securely
- and see some hypothetical ways that they've gone wrong in the past.



Protocols

Issues

Solutions

What is federated authentication?

aka Single Sign-On

A central service authenticates users and securely issues responses for consuming services to trust.

Why do we care about federated authentication?

- Only trusted parties handle credentials!
- Reduces password re-use by centralizing authentication
- Centralized location to enforce 2FA/MFA, login policies, risk assessment
- Just-in-time account creation reduces data stored by applications

What protocols enable federated authentication?

- Kerberos (1989 - MIT)
  - MIT Kerberos
  - Heimdal
  - Active Directory
- SAML (2002 - OASIS)
- CAS (2003 - Stanford)
- OpenID Connect (2014 - OpenID Foundation)

Other notable protocols:

- PKI/Smart cards
- NTLM
- CoSign (UM) 🪦

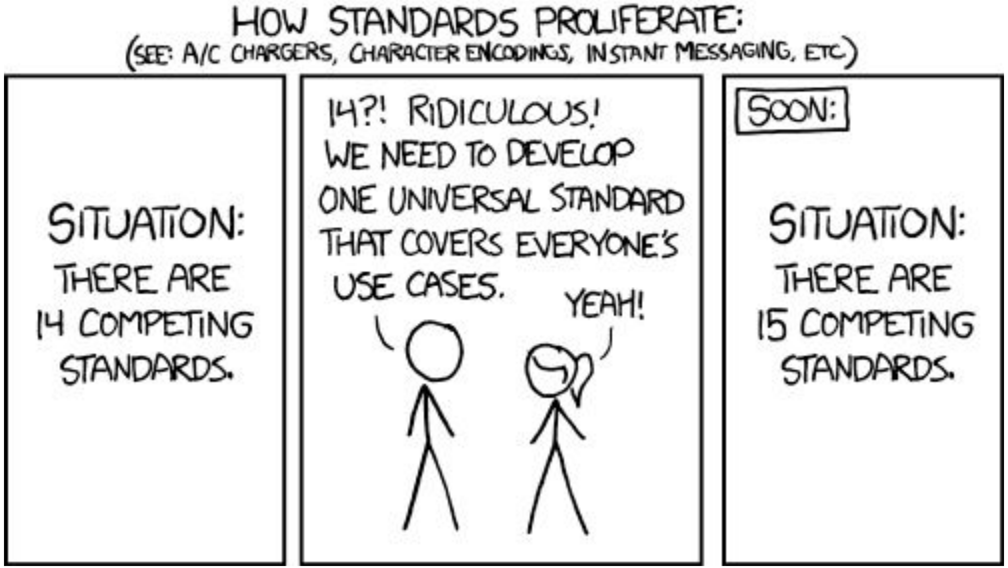
What protocols enable federated authentication?

- Kerberos (1989 - MIT)
    - MIT Kerberos
    - Heimdal
    - Active Directory
- SAML (2002 - OASIS)
- CAS (2003 - Stanford)
- OpenID Connect (2014 - OpenID Foundation)

Other notable protocols:

- PKI/Smart cards
- NTLM
- CoSign (UM) 🪦

What protocols enable federated authentication?


HOW STANDARDS PROLIFERATE:
(SEE: A/C CHARGERS, CHARACTER ENCODINGS, INSTANT MESSAGING, ETC)
SITUATION: THERE ARE 14 COMPETING STANDARDS.
14?! RIDICULOUS! WE NEED TO DEVELOP ONE UNIVERSAL STANDARD THAT COVERS EVERYONE'S USE CASES.
YEAH!
SOON:
SITUATION: THERE ARE 15 COMPETING STANDARDS.

https://xkcd.com/927/

SAML and OIDC

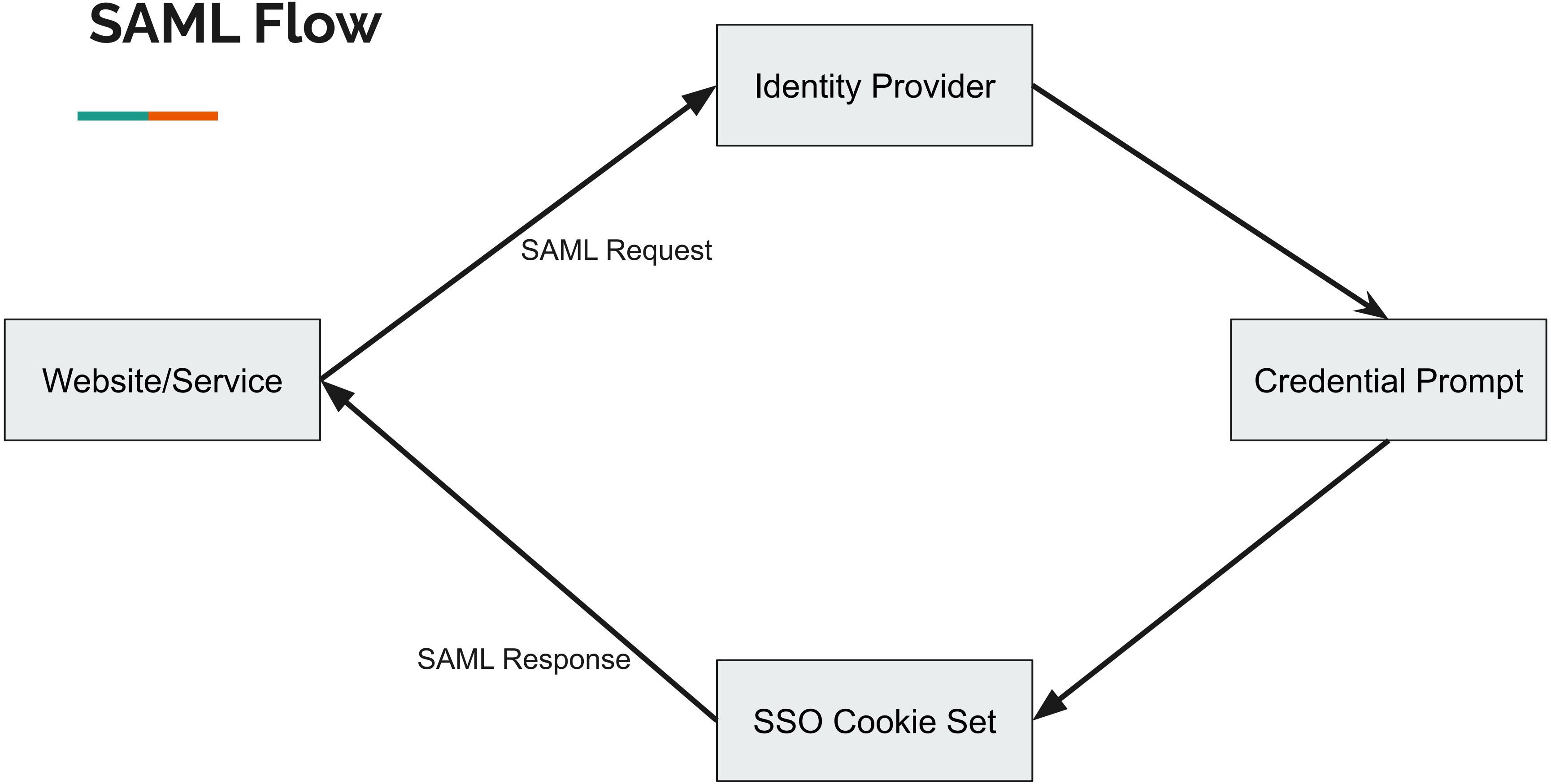
SAML

- XML-based
- Authentication only
- Public-private keys
- Trust built via out-of-band configuration
- Front-channel communication

OIDC

- JSON-based
- Identity layer on top of OAuth authorization protocol
- Public-private keys or basic authentication
- Relies on global PKI/server TLS
- Front-channel and back-channel

SAML Flow


Identity Provider
SAML Request
Website/Service
Credential Prompt
SAML Response
SSO Cookie Set

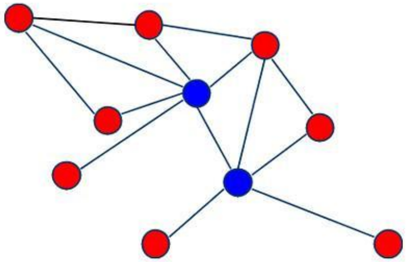
SAML Keys

Service Provider

- Private Key - signs requests
- Public Key - decrypts responses

Identity Provider

- Private Key - signs responses
- Public Key - validates responses



Protocols

Issues

Solutions

```
<?xml version="1.0"?>
<samlp:Response xmlns:samlp="urn:oasis:names:tc:SAML:2.0:protocol" xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion" ID="pfx2f6a5da4-e4fe-1726-0479-2dffd3ae82d4" Version="2.0" IssueInstant="2014-07-17T01:01:48Z"
Destination="http://sp.example.com/demo1/index.php?acs" InResponseTo="ONELOGIN_4fee3b046395c4e751011e97f8900b5273d56685">
  <saml:Issuer>http://idp.example.com/metadata.php</saml:Issuer>
  <ds:Signature xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
    <ds:SignedInfo>
      <ds:CanonicalizationMethod Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#"/>
      <ds:SignatureMethod Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-sha1"/>
      <ds:Reference URI="#pfx2f6a5da4-e4fe-1726-0479-2dffd3ae82d4">
        <ds:Transforms>
          <ds:Transform Algorithm="http://www.w3.org/2000/09/xmldsig#enveloped-signature"/>
          <ds:Transform Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#"/>
        </ds:Transforms>
        <ds:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
        <ds:DigestValue>ELIDED</ds:DigestValue>
      </ds:Reference>
    </ds:SignedInfo>
    <ds:SignatureValue>ELIDED</ds:SignatureValue>
    <ds:KeyInfo>
      <ds:X509Data>
        <ds:X509Certificate>ELIDED</ds:X509Certificate>
      </ds:X509Data>
    </ds:KeyInfo>
  </ds:Signature>
  <samlp:Status>
    <samlp:StatusCode Value="urn:oasis:names:tc:SAML:2.0:status:Success"/>
  </samlp:Status>
  <saml:Assertion xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xs="http://www.w3.org/2001/XMLSchema" ID="_d71a3a8e9fcc45c9e9d248ef7049393fc8f04e5f75" Version="2.0" IssueInstant="2014-07-17T01:01:48Z">
    <saml:Issuer>http://idp.example.com/metadata.php</saml:Issuer>
    <saml:Subject>
      <saml:NameID SPNameQualifier="http://sp.example.com/demo1/metadata.php" Format="urn:oasis:names:tc:SAML:2.0:nameid-format:transient">_ce3d2948b4cf20146dee0a0b3dd6f69b6cf86f62d7</saml:NameID>
      <saml:SubjectConfirmation Method="urn:oasis:names:tc:SAML:2.0:cm:bearer">
        <saml:SubjectConfirmationData NotOnOrAfter="2024-01-18T06:21:48Z" Recipient="http://sp.example.com/demo1/index.php?acs" InResponseTo="ONELOGIN_4fee3b046395c4e751011e97f8900b5273d56685"/>
      </saml:SubjectConfirmation>
    </saml:Subject>
    <saml:Conditions NotBefore="2014-07-17T01:01:18Z" NotOnOrAfter="2024-01-18T06:21:48Z">
      <saml:AudienceRestriction>
        <saml:Audience>http://sp.example.com/demo1/metadata.php</saml:Audience>
      </saml:AudienceRestriction>
    </saml:Conditions>
    <saml:AuthnStatement AuthnInstant="2014-07-17T01:01:48Z" SessionNotOnOrAfter="2024-07-17T09:01:48Z" SessionIndex="_be9967abd904ddcae3c0eb4189adbe3f71e327cf93">
      <saml:AuthnContext>
        <saml:AuthnContextClassRef>urn:oasis:names:tc:SAML:2.0:ac:classes:Password</saml:AuthnContextClassRef>
      </saml:AuthnContext>
    </saml:AuthnStatement>
    <saml:AttributeStatement>
      <saml:Attribute Name="uid" NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:basic">
        <saml:AttributeValue xsi:type="xs:string">test</saml:AttributeValue>
      </saml:Attribute>
      <saml:Attribute Name="mail" NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:basic">
        <saml:AttributeValue xsi:type="xs:string">test@example.com</saml:AttributeValue>
      </saml:Attribute>
      <saml:Attribute Name="eduPersonAffiliation" NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:basic">
        <saml:AttributeValue xsi:type="xs:string">users</saml:AttributeValue>
        <saml:AttributeValue xsi:type="xs:string">examplerole1</saml:AttributeValue>
      </saml:Attribute>
    </saml:AttributeStatement>
  </saml:Assertion>
</samlp:Response>
```

Signature

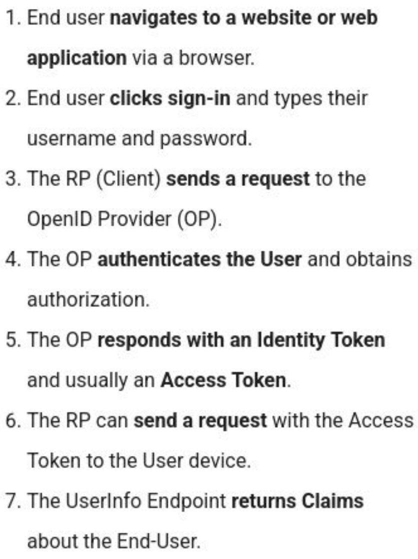
Assertion

Unique ID

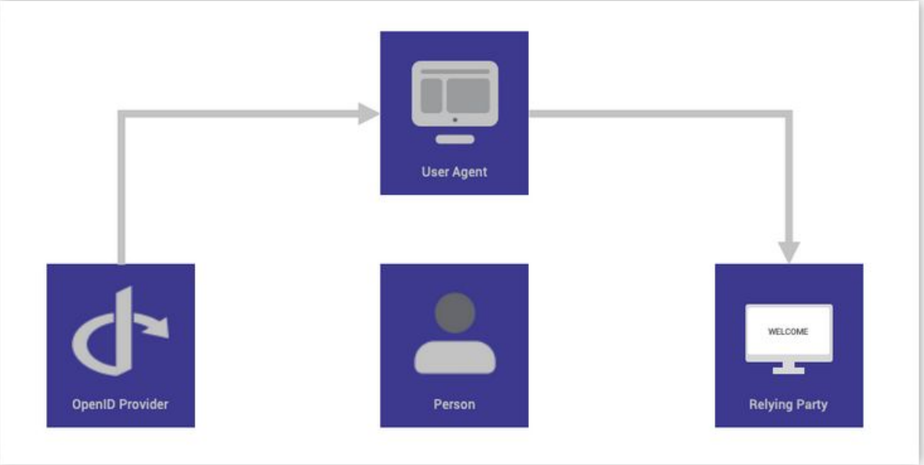
Exploits, overlooked issues… maybe developer gaps?

- Response signature not checked - Elevate access by modifying response to admin user’s info
- Response not encrypted - Leak PII
- Insecure SSO or session cookie
- Message replay or expiration not checked
- Spoofing headers when app should use env vars
- Disclosure of secrets (private key [SAML] or account password [OIDC])

OpenID Connect



1. End user **navigates to a website or web application** via a browser.
2. End user **clicks sign-in** and types their username and password.
3. The RP (Client) **sends a request** to the OpenID Provider (OP).
4. The OP **authenticates the User** and obtains authorization.
5. The OP **responds with an Identity Token** and usually an **Access Token**.
6. The RP can **send a request** with the Access Token to the User device.
7. The UserInfo Endpoint **returns Claims** about the End-User.

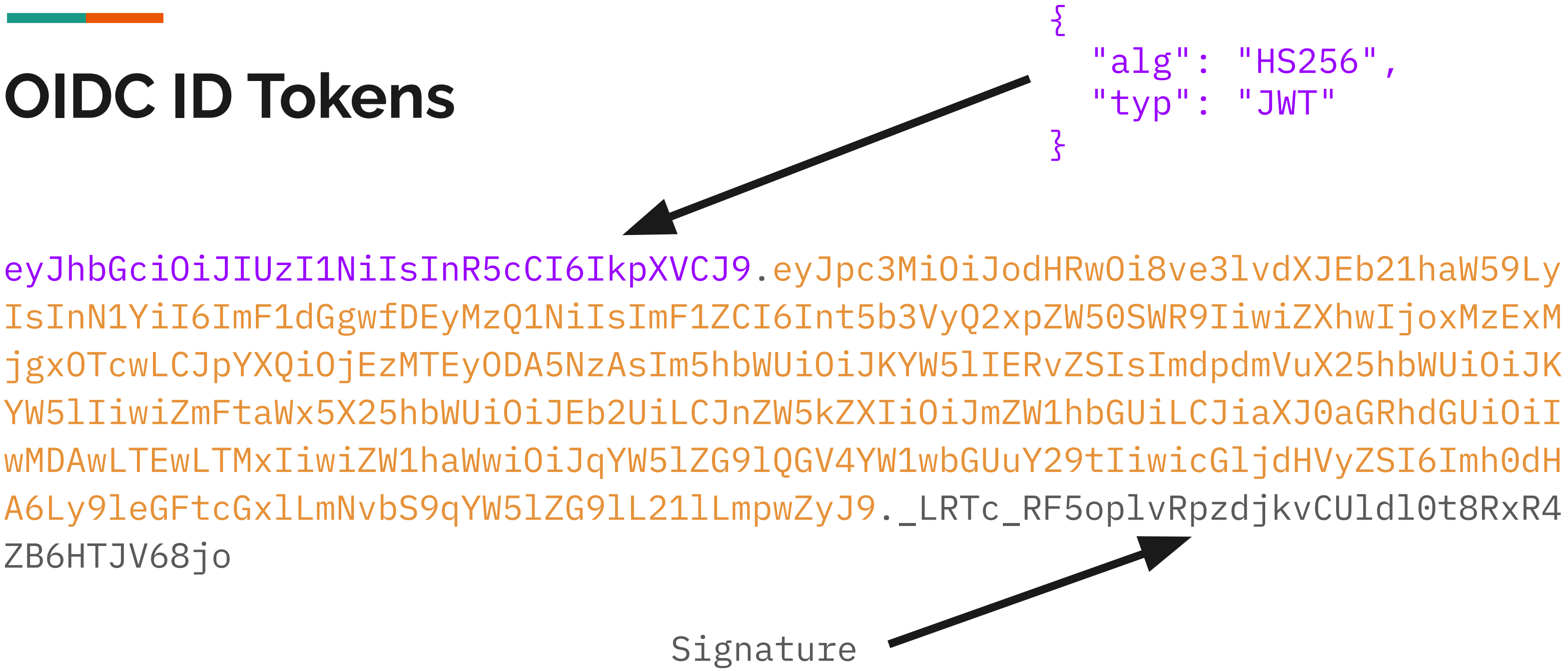


https://openid.net/developers/how-connect-works/

OIDC ID Tokens

```
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpc3MiOiJodHRwOi8ve3lvdXJEb21haW59Ly
IsInN1YiI6ImF1dGgwfDEyMzQ1NiIsImF1ZCI6Int5b3VyQ2xpZW50SWR9IiwiZXhwIjoxMzExM
jgxOTcwLCJpYXQiOjEzMTEyODA5NzAsIm5hbWUiOiJKYW5lIERvZSIsImdpdmVuX25hbWUiOiJK
YW5lIiwiZmFtaWx5X25hbWUiOiJEb2UiLCJnZW5kZXIiOiJmZW1hbGUiLCJiaXJ0aGRhdGUiOiI
wMDAwLTEwLTMxIiwiZW1haWwiOiJqYW5lZG9lQGV4YW1wbGUuY29tIiwicGljdHVyZSI6Imh0dH
A6Ly9leGFtcGxlLmNvbS9qYW5lZG9lL21lLmpwZyJ9._LRTc_RF5oplvRpzdjkvCUldl0t8RxR4
ZB6HTJV68jo
```

OIDC ID Tokens



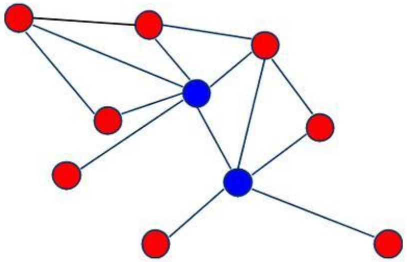
```
{
  "alg": "HS256",
  "typ": "JWT"
}
```

```
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpc3MiOiJodHRwOi8ve3lvdXJEb21haW59Ly
IsInN1YiI6ImF1dGgwfDEyMzQ1NiIsImF1ZCI6Int5b3VyQ2xpZW50SWR9IiwiZXhwIjoxMzExM
jgxOTcwLCJpYXQiOjEzMTEyODA5NzAsIm5hbWUiOiJKYW5lIERvZSIsImdpdmVuX25hbWUiOiJK
YW5lIiwiZmFtaWx5X25hbWUiOiJEb2UiLCJnZW5kZXIiOiJmZW1hbGUiLCJiaXJ0aGRhdGUiOiI
wMDAwLTEwLTMxIiwiZW1haWwiOiJqYW5lZG9lQGV4YW1wbGUuY29tIiwicGljdHVyZSI6Imh0dH
A6Ly9leGFtcGxlLmNvbS9qYW5lZG9lL21lLmpwZyJ9._LRTc_RF5oplvRpzdjkvCUldl0t8RxR4
ZB6HTJV68jo
```

Signature

OIDC ID Tokens

```
{
  "iss": "https://example.com/oidc/",
  "sub": "123456",
  "aud": "https://api.example.com/",
  "exp": 1311281970,
  "iat": 1311280970,
  "name": "Jane Doe",
  "given_name": "Jane",
  "family_name": "Doe",
  "gender": "female",
  "birthdate": "0000-10-31",
  "email": "janedoe@example.com",
  "picture": "http://example.com/janedoe/me.jpg"
}
```



Protocols

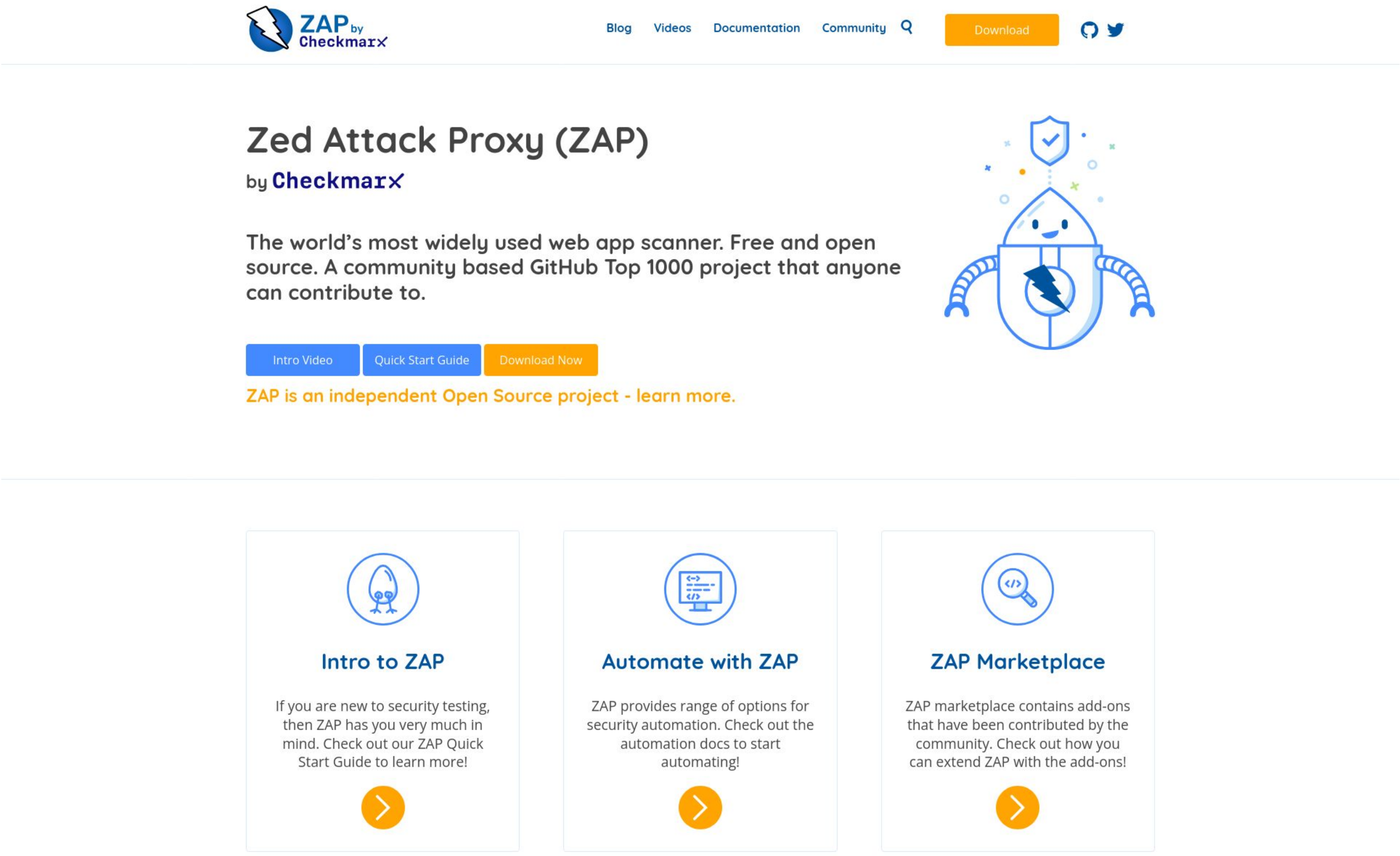
Issues

Solutions

Solutions

- Maintain libraries
- Ensure “secure” cookie flag
- Use modern, secure algorithms
- Protect SAML private key and OIDC password
-

- SAML: Inspect messages
- SAML: Tamper with messages



Zed Attack Proxy (ZAP)

by Checkmarx

The world's most widely used web app scanner. Free and open source. A community based GitHub Top 1000 project that anyone can contribute to.

ZAP is an independent Open Source project - learn more.

Intro to ZAP

If you are new to security testing, then ZAP has you very much in mind. Check out our ZAP Quick Start Guide to learn more!

Automate with ZAP

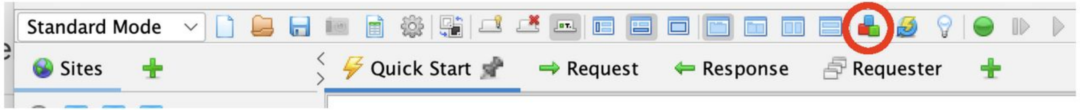
ZAP provides range of options for security automation. Check out the automation docs to start automating!

ZAP Marketplace

ZAP marketplace contains add-ons that have been contributed by the community. Check out how you can extend ZAP with the add-ons!

ZAP Marketplace

ZAP Marketplace contains ZAP add-ons which have been written by the ZAP team and the community. The add-ons help to extend the functionalities of ZAP. If you are using the latest version of ZAP then you can browse and download add-ons from within ZAP by clicking on this button in the toolbar:



You can also import the add-ons that you have downloaded manually via the “File / Load Add-on File…” menu option in the ZAP desktop.

Name	ID	Version	Status	Author	Last Updated
saml					
SAML Support Detect, Show, Edit, Fuzz SAML requests	saml	10	alpha	ZAP Dev Team	2022-10-28

Going Further

https://cheatsheetseries.owasp.org/cheatsheets/SAML_Security_Cheat_Sheet.html

https://duo.com/blog/the-beer-drinkers-guide-to-saml

https://docs.oasis-open.org/security/saml/v2.0/saml-core-2.0-os.pdf

https://openid.net/

Final Thoughts

- Each protocol has its place
- Federated authentication is better than password sprawl
- Security is layers of compensating controls
- Federated authentication enables better adoption of stronger credentials, 2FA, MFA, etc.