

Crumbling the Cookie:

Fixing a Weak Link in Authentication on the Web



Zack Voase

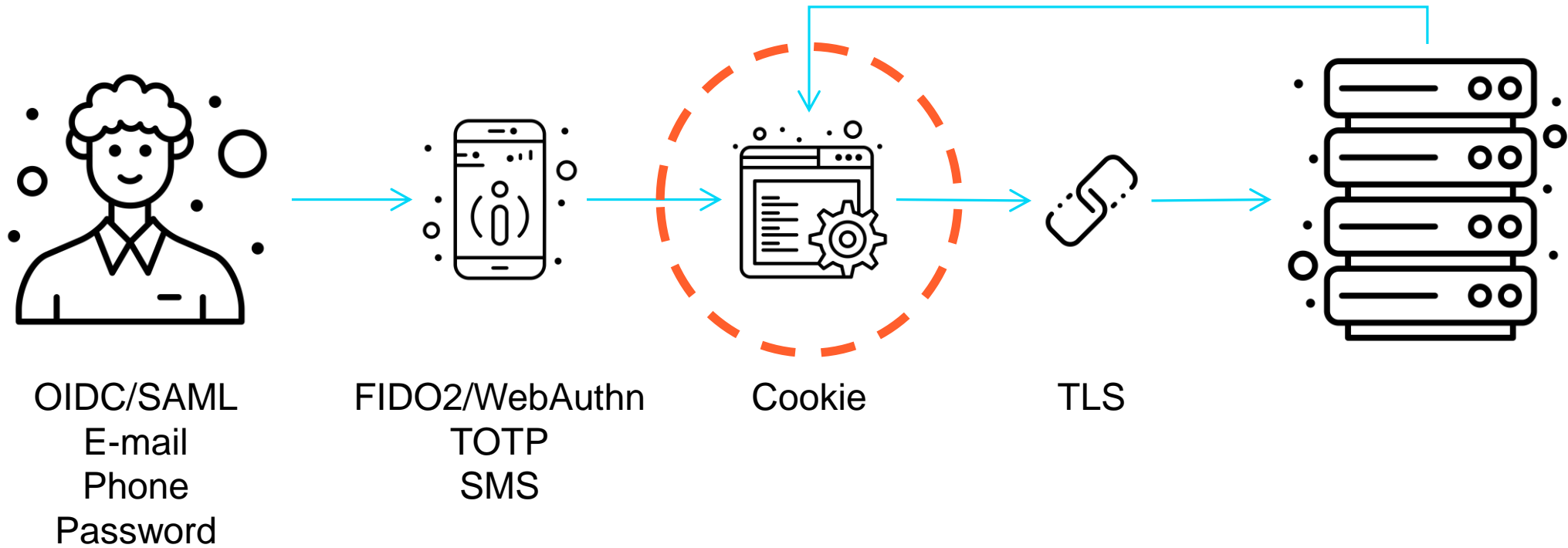
Senior Security SWE
Netflix

Agenda

- What's in a cookie?
- What's wrong with cookies?
- What would a better solution look like?
- A rough sketch of an improvement
- Closing thoughts



The Rusty Chain of Authentication

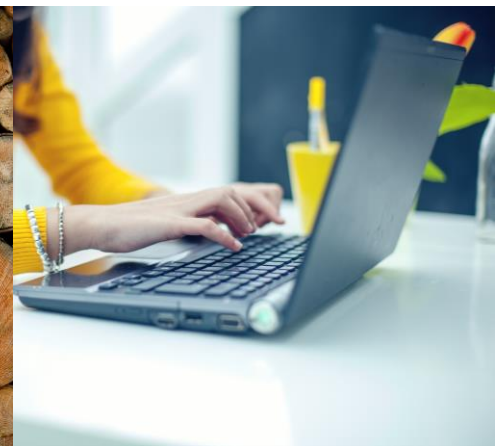
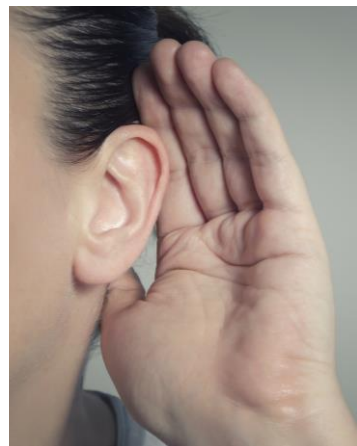


The Ingredients of a Cookie

- *Bearer Token* (Session ID or JWT)
- Expiration time
- Domain/path constraints
- Flags for security:
 - SameSite
 - Secure
 - HttpOnly
- `__Host-` and `__Secure-` prefixes



Threats, Threats, Everywhere





The Quadrant of Desires®



Security

- MITM-resistant
- Tamper/theft-evident
- No insecure configuration
- Can use modern hardware



Privacy

- User control over sessions
- Sessions without identification
- Lower risk of edge caching



Developer Experience

- Can be handled at the LB
 - High performance



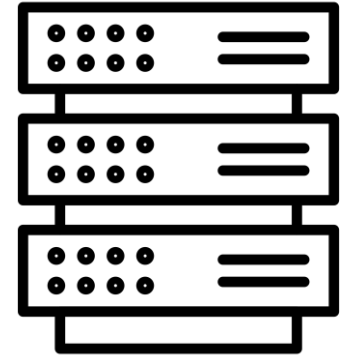
User Experience

- Integration with browser
- OS lock/unlock/attention-aware

Prior Work

- Mozilla Persona/BrowserID (2011–2016)
- TLS Token Binding
- OAuth DPoP (Demonstrated Proof-of-Possession)
- Cakes
- Macaroons
- HTTP State Tokens


Sketching A New Protocol: WebSession

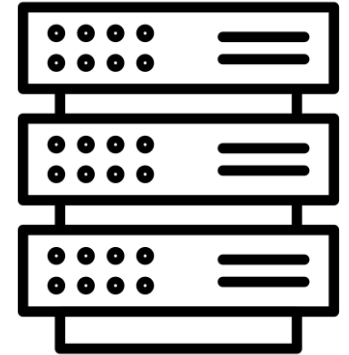


Sketching A New Protocol: WebSession



200 OK
WWW-Authenticate:
WebSession <S_{pub}> [<Options>]

 S_{pub}, S_{priv}




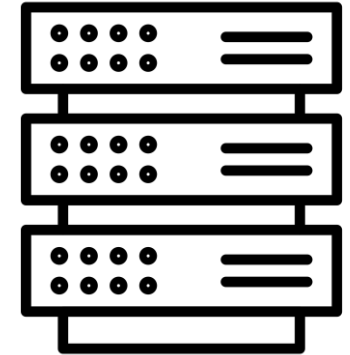
Sketching A New Protocol: WebSession




Start a new session?

Yes No

 S_{pub} , S_{priv}



Sketching A New Protocol: WebSession

 C_{pub} , C_{priv}




$secret = DH(C_{priv}, S_{pub})$

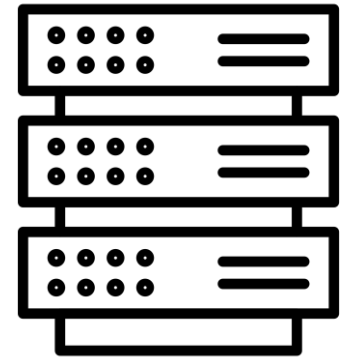
$nonce = random()$

$body = C_{pub} . S_{pub} . origin . nonce$


$signature = HMAC(secret, body)$

$token = signature . body$

 S_{pub} , S_{priv}




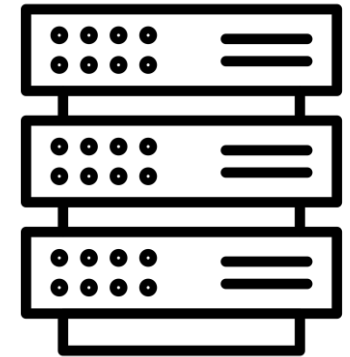
Sketching A New Protocol: WebSession

 C_{pub} , C_{priv}




GET /
Authorization:
WebSession <token>


 S_{pub} , S_{priv}

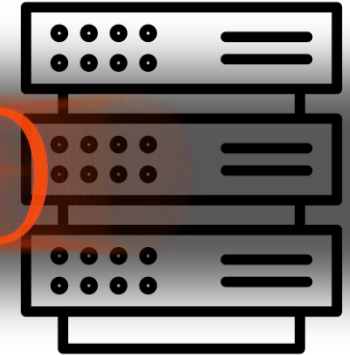


Sketching A New Protocol: WebSession

 C_{pub} , C_{priv}



 S_{pub} , S_{priv}



SESSION ESTABLISHED

```
signature, body = token.split()
 $S_{pub}$ ,  $C_{pub}$ , origin, nonce = body.split()
 $S_{priv}$  = lookup( $S_{pub}$ )
secret = DH( $C_{pub}$ ,  $S_{priv}$ )
check_for_nonce_reuse(nonce)
signature == HMAC(secret, body)
```

WebSession Options

- WebAuthn integration
 - Server can explicitly query for WebAuthn discoverable credentials
 - S_{pub} can be used as a WebAuthn challenge
 - Secure session + identity establishment in 3 round-trips and no JS
- Request log out on screen lock/power off/inactivity/etc.
- Domain and path scoping
- First-Party Sets

Revisiting the Quadrant



Security

- MITM-resistant
- Tamper/theft-evident
- No insecure configuration
- Can use modern hardware



Privacy

- User control over sessions
- Sessions without identification
- Lower risk of edge caching



Developer Experience

- Can be handled at the LB
 - High performance



User Experience

- Integration with browser
- OS lock/unlock/attention-aware

Summary and Open Questions

- Cookies are bad, right now we're stuck with them
- A better future is possible
- But it's going to take work
- Are the incentives there?



THANK YOU!