HTTP Unprompted Authentication

draft-ietf-httpbis-unprompted-auth

IETF 117 – San Francisco – 2023-07-26

David Schinazi – dschinazi.ietf@gmail.com
David Oliver – david@guardianproject.info
Jonathan Hoyland – jonathan.hoyland@gmail.com

Quick Summary, Motivation, History

Client authenticates to server

Using asymmetric cryptography

Server hides the fact that it serves authenticated resources

Adopted by HTTPBIS in February, discussed in Yokohama in March

Changed almost everything since then based on input from Yokohama

Rough Shape of Solution (this part didn't change)

Use TLS Key exporter to generate nonce

Sign or HMAC the nonce

Doesn't leak any information

Can't be replayed on a separate connection

This is now an Authentication Scheme

Remove Unprompted-Authentication header entirely

Introduce new Signature auth scheme

Used with Authorization or Proxy-Authorization

(Dropped HMAC feature entirely)

We now use the TLS exporter context

```
Signature Algorithm (16),          Algorithm
Key ID Length (i),                 Key ID
Key ID (..),
Scheme Length (i),                 Origin
Scheme (..),
Host Length (i),
Host (..),
Port (16),
Realm Length (i),                  Realm (optional)
Realm (..),
```

Send part of the key exporter output in header

Avoids SEEMS-LEGIT attacks where in some signature schemes there exist signatures that are valid for multiple inputs

Generate 48 bytes of key exporter output:

```
Signature Input (256),
Verification (128),
```

Send verification in header

We now contextualize the signature

Reusing keys between protocols is forbidden but someone might still do it

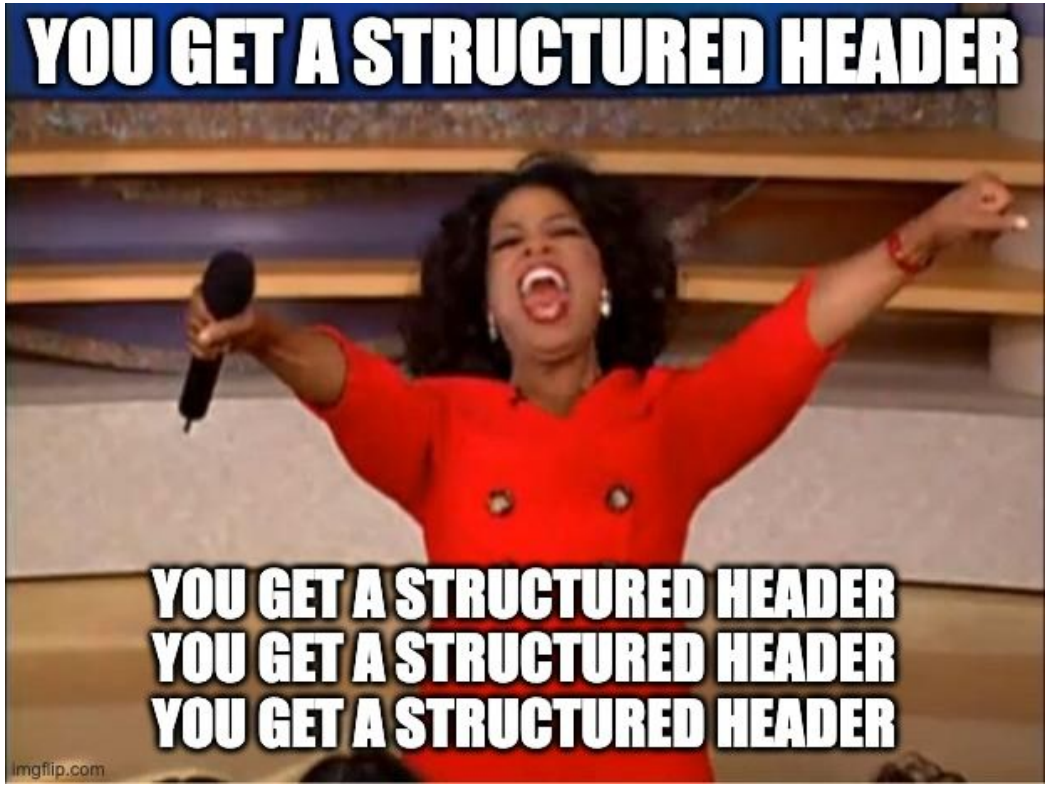
So we copied what TLS 1.3 does

- A string that consists of octet 32 (0x20) repeated 64 times
- The context string "HTTP Signature Authentication"
- A single 0 byte which serves as a separator
- Signature Input extracted from the key exporter output

Putting it all together

```
Authorization: Signature \
  k=":YmFzZW1lbnQ=:", \
  s=2055, \
  v=":dmVyaWZpY2F0aW9uXzE2Qg==:", \
  p=":SW5zZXJ0IHNpZ25hdHVyZSBvZiBub25jZSBoZXJlIHdo \
    aWNoIHRha2VzIDUxMiBiaXRzIGZvciBFZDI1NTE5IQ==:"
```

#2581: Structured Headers



#2581: Structured Headers

We want to send bytes

```
Example-ByteSequence: :V2UgPDMgU3RydWN0dXJlZCBGaWVsZHM=:
```

Except auth parameters are not structured headers

```
auth-param = token BWS "=" BWS ( token / quoted-string )
```

We can always skip padding (=) but parameters don't allow unquoted colons

So we added quotes

```
v=":dmVyaWZpY2F0aW9uXzE2Qg==:"
```

#2581: Structured Headers

So, do we want to keep or leave them?
We have 3 of these byte parameters and one integer

Proposal 1: quoted structured header

```
v=":dmVyaWZpY2F0aW9uXzE2Qg==:"
```

Proposal 2: base64url without padding

```
v=dmVyaWZpY2F0aW9uXzE2Qg
```

#2599: Add public key to context

Potential key confusion attacks when the server is checking the signature using the wrong public key

Simplest solution: add the public key to the TLS key exporter

Should solve the issue without increasing the amount of bytes sent on the wire

Is that correct?

HTTP Unprompted Authentication

draft-ietf-httpbis-unprompted-auth

IETF 117 – San Francisco – 2023-07-26

David Schinazi – dschinazi.ietf@gmail.com
David Oliver – david@guardianproject.info
Jonathan Hoyland – jonathan.hoyland@gmail.com