Access token phishing

John Bradley, Torsten Lodderstedt

OAuth Security Workshop
July 13&14 2017, ETH Zurich
What’s the setup?

Client

RS X

dynamic binding via Standard API (e.g. e-Mail, Banking, eSignature)

AS

Obtain Access Token

Client
What if ...
… RS X is a bad guy?

AS

Obtain Access Token

Client

RS X

RS 1

legitimate RS
What can we do?
What if the client would know upfront which places it is safe to send access tokens to?

```
{
  ...
  "resource_servers": ["email.example.com", "storage.example.com", "video.example.com"]
  ...
}
```

puts the burden of security checks to clients
Audience Restriction

AS

unknown RS

Obtain Access Token for https://rsx.evil.com*

RS X
https://rsx.evil.com

Client

RS 1
https://rs1.legit.com

* e.g. using https://tools.ietf.org/html/draft-campbell-oauth-resource-indicators

Audience does not match
Proof of Possession

AS

Obtain Access Token for Client

Client

Signature (rsx, client)

Client Key X, ...

key material

RS X

https://rsx.evil.com

Signature (rsx, client)

Client Key X, ...

RS 1

https://rs1.legit.com

Signature does not match

Client Key X, ...

Signature (rsx, client)
Proof Posession (Existing Proposals)

- Transport
  - Token Binding - draft-ietf-oauth-token-binding
  - MTLS - draft-ietf-oauth-mtls

- Application
  - Signed Request - draft-ietf-oauth-signed-http-request
  - J-POP - draft-sakimura-oauth-jpop
What do you think?