Token Binding Protocol I-D Changes Since IETF 95

Andrei Popov, Microsoft Corp.

TokenBindingKeyParameters

Per feedback, the initial set of identifiers for the Token Binding key parameters has been moved from TBNEGO to TBPROTO, without change:

```
enum {
  rsa2048_pkcs1.5(0), rsa2048_pss(1), ecdsap256(2), (255)
} TokenBindingKeyParameters;
```

The corresponding IANA considerations moved to TBPROTO as well.

referred_token_binding

A more detailed explanation of referred Token Bindings has been added in section 3:

“Token Binding over HTTP [I-D.ietf-tokbind-https] describes a use case for referred_token_binding where Token Bindings are established between multiple communicating parties: User Agent, Identity Provider and Relying Party. User Agent sends referred_token_binding to the Identity Provider in order to prove possession of the Token Binding key it uses with the Relying Party. The Identity Provider can then bind the token it is supplying (for presentation to the Relying Party) to the Token Binding ID contained in the referred_token_binding.”

referred_token_binding

The construction of referred Token Bindings clarified in section 4.1:

- Set TokenBinding.tokenbinding_type to referred_token_binding.
- Set TokenBinding.tokenbindingid to the Token Binding ID used with the Relying Party.
- Set TokenBinding.signature to the result of signing the EKM value of the TLS connection to the Identity Provider, using the Token Binding key established with the Relying Party and the signature algorithm indicated by the associated key parameters. Note that these key parameters may differ from the key parameters negotiated with the Identity Provider.

ecdsap256 Signature and Key Formats

Further clarified ecdsap256 signature and key formats:

“When an ecdsap256 key is used, TokenBinding.signature contains a pair of **32-byte** integers, R followed by S, generated using Curve P-256 as defined in [ANSI.X9-62.2005] and [FIPS.186-4.2013]. R and S are encoded in big-endian format, **preserving any leading zero bytes**. ECPoint.point contains the X coordinate followed by the Y coordinate. The X and Y coordinates are unsigned **32-byte** integers encoded in big-endian format, **preserving any leading zero bytes**. Future specifications may define Token Binding keys using other elliptic curves with their corresponding signature and point formats.”

TokenBindingID

Added a recommendation that the Token Binding ID be presented to the application as an opaque byte sequence:

“Token Binding protocol implementations SHOULD make Token Binding IDs available to the application as opaque byte sequences. E.g. server applications will use Token Binding IDs when generating and verifying bound tokens.”

Clients and Servers MAY Create Bound Tokens

Per feedback, clarified that both clients and servers MAY create bound tokens:

“For example, HTTPS servers employing Token Binding for securing their HTTP cookies will bind the cookies. In the case of a server-initiated challenge-response protocol employing Token Binding and TLS, the client can, for example, incorporate the Token Binding ID within the signed object it returns, thus binding the object.”

Links And Contact Information

- TLS Extension for Token Binding Negotiation: https://datatracker.ietf.org/doc/draft-ietf-tokbind-negotiation/
- The Token Binding Protocol Version 1.0: https://datatracker.ietf.org/doc/draft-ietf-tokbind-protocol/
- Token Binding over HTTP: https://datatracker.ietf.org/doc/draft-ietf-tokbind-https/
- GitHub: https://github.com/TokenBinding/Internet-Drafts

- Dirk Balfanz balfanz@google.com
- Andrei Popov andreipo@microsoft.com
- Jeff Hodges Jeff.Hodges@paypal.com

The Token Binding Protocol Message Format

```
struct {
    ExtensionType extension_type;
    opaque extension_data<0..2^16-1>;
} Extension;
struct {
    TokenBindingType tokenbinding_type;
    TokenBindingID tokenbindingid;
    opaque signature<0..2^16-1>;// Signature over the exported keying material value
    Extension extensions<0..2^16-1>;
} TokenBinding;
struct {
    TokenBinding tokenbindings<0..2^16-1>;
} TokenBindingMessage;
```

Token Binding ID Format

```
struct {
    TokenBindingKeyParameters key_parameters;
    select (key_parameters) {
        case rsa2048_pkcs1.5:
        case rsa2048_pss:
            RSAPublicKey rsapubkey;
        case ecdsap256:
            ECPoint point;
    }
} TokenBindingID;
```

- Provided_token_binding is used to establish a Token Binding when connecting to a server.
- Referred_token_binding is used when requesting tokens to be presented to a different server.