Strong Authentication in Web Application
“State of the Art 2011”

Sylvain Maret / Digital Security Expert / OpenID Switzerland

@smaret
Version 0.9a
Abstract
Strong Authentication: State of the Art 2011

- Risk Based Authentication
- Biometry - Match on Card
- OTP for Smartphones
- OTP SMS
- PKI
- SuisseID
- Mobile-OTP
- OATH (HOTP, TOTP, OCRA)
- Open Source approach

How to integrate Strong Authentication in Web Application?

- OpenID, SAML, Identity Federation for Strong Authentication
- API, SDK, Agents, Web Services, Modules
- PAM, Radius, JAAS
- Reverse Proxy (WAF) and WebSSO
- PKI / SSL client authentication
- PHP example with Multi-OTP PHP class
- AppSec (Threat Modeling - OWASP)
RSA FAILED ?
Who am I?

- Security Expert
  - 17 years of experience in ICT Security
  - Principal Consultant at MARET Consulting
  - Expert at Engineer School of Yverdon & Geneva University
  - Swiss French Area delegate at OpenID Switzerland
  - Co-founder Geneva Application Security Forum
  - OWASP Member
  - Author of the blog: [la Citadelle Electronique](http://ch.linkedin.com/in/smaret) or @smaret
  - [http://www.slideshare.net/smaret](http://www.slideshare.net/smaret)

- Chosen field
  - AppSec & Digital Identity Security
Protection of digital identities: a topical issue…
«Digital identity is the cornerstone of trust»

Definition of strong authentication

Strong Authentication on Wikipedia
Strong Authentication
A new paradigm?
Which Strong Authentication technology?
Legacy Token / OTP / PKI / SuisseID ? / Open Source Solution ?
<table>
<thead>
<tr>
<th></th>
<th>OTP</th>
<th>PKI (HW)</th>
<th>Biometry</th>
</tr>
</thead>
<tbody>
<tr>
<td>Strong authentication</td>
<td>✅</td>
<td>✅</td>
<td></td>
</tr>
<tr>
<td>Encryption</td>
<td>✅</td>
<td>✅</td>
<td></td>
</tr>
<tr>
<td>Digital signature</td>
<td>✅</td>
<td>✅</td>
<td></td>
</tr>
<tr>
<td>Non repudiation</td>
<td>✅</td>
<td>✅</td>
<td></td>
</tr>
<tr>
<td>Strong link with the user</td>
<td>✅</td>
<td></td>
<td>✅</td>
</tr>
</tbody>
</table>
Strong Authentication with PKI
PKI: Digital Certificate

Software Certificate (PKCS#12; PFX)

Hardware Token (Crypto PKI)
Strong Authentication
SSL/TLS Mutual Authentication: how does it work?

Validation Authority

CRL or OCSP Request

Valid
Invalid
Unknown

SSL / TLS Mutual Authentication
Demo #1: Software Certificate Auth using an IDP OpenID

http://www.clavid.com/
Strong Authentication with Biometry (Match on Card technology)

- A reader
  - Biometry
  - SmartCard

- A card with chip
  - Technology MOC
  - Crypto Processor
    - PC/SC
    - PKCS#11
    - Digital certificate X509
Strong Authentication

With

(O)ne (T)ime (P)assword
(O)ne (T)ime (P)assword

- OTP Time Based
  - Like SecurID
- OTP Event Based
- OTP Challenge Response Based

- Others:
  - OTP via SMS
  - OTP via email
  - Biometry and OTP
  - Bingo Card
  - Etc.
ie = OTP(K, T) = Truncate(HMAC-SHA-1(K, T))
ie = OTP(K,C) = \text{Truncate}(\text{HMAC-SHA-1}(K,C))
Crypto-101 / OTP Challenge Response Based

K = Secret Key / Seed

nonce

HASH Function

OTP Challenge

CLIENT
(PROVER)

Verifier sends challenge to prover
Challenge = Q

Prover Computes Response
R = OCRA(K, {C | Q | [P | S | T]})
Prover sends Response = R

SERVER
(VERIFIER)

Verifier Validates Response
If Response is valid, Server sends OK
If Response is not, Server sends NOK

ie:
Others OTP technologies…

OTP Via SMS

“Flicker code” Generator Software that converts already encrypted data into optical screen animation

By Elcard
Demo #2: Protect WordPress (OTP Via SMS)
How to Store my Secret Key?

A Token!
OTP Token: Software vs Hardware?
Software OTP for Smartphone

Hack RSA SecurID : l'histoire n'est pas terminée !

Beaucoup de personnes dans la communauté de la sécurité informatique se demandent pourquoi tant de silence de la part de RSA SecurID. La phrase qui ressort le plus est : « RSA Silent About Compromise For 7 Days – Assume SecurID Is Broken ». Effectivement, le doute n'est pas bon en sécurité des systèmes d'information. Mais que fait donc RSA ?

Selon moi, comme mentionné dans mon 1er billet, cette histoire est un évènement marquant de la sécurité informatique.

ICTTF - International Cyber Threat Task Force

RSA SecurID Hack - My Opinion

Posted by Paul C Dwyer March 26 - Filed in Authentication - #APT #rea #securid - 1,678 views

Okay so what's all this about the RSA SecurID Hack?

Since the news, I have received a large number of calls and emails for my opinion on this matter. So here it is, hope it helps and as always it's my personal opinion and not that of my employer.

So let's look at some background and key facts. The RSA SecurID technology is used for authentication.

Mostly used for corporate and government VPN's it is the de facto way of authenticating

L'art de fortifier ne consiste pas dans des règles et des systèmes mais uniquement dans le bon sens et
Where are the seed?
Crypto-101 / Time Based OTP

S = Serial Number

K = Secret Key / Seed

T = UTC Time

HASH Function
RSA SecurID

ie = OTP(K, T, S) = Truncate(HASH(K, T, S))
Seed distribution? Still a good model?

Editor / Vendor
Secret Key are generated on promise
### RSA SecurID Token Calculator

#### System Time

<table>
<thead>
<tr>
<th>Serial Number</th>
<th>Key</th>
<th>Delta Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>3C</td>
<td>1</td>
<td>76a379...</td>
</tr>
</tbody>
</table>

#### Local Time

<table>
<thead>
<tr>
<th>Time Hash</th>
<th>Local Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>601258</td>
<td>2011/05/12 - 15:55</td>
</tr>
<tr>
<td>554294</td>
<td>2011/05/12 - 15:56</td>
</tr>
<tr>
<td>821442</td>
<td>2011/05/12 - 15:57</td>
</tr>
<tr>
<td>583947</td>
<td>2011/05/12 - 15:58</td>
</tr>
<tr>
<td>697354</td>
<td>2011/05/12 - 15:59</td>
</tr>
<tr>
<td>129263</td>
<td>2011/05/12 - 16:00</td>
</tr>
<tr>
<td>757723</td>
<td>2011/05/12 - 16:01</td>
</tr>
<tr>
<td>742894</td>
<td>2011/05/12 - 16:02</td>
</tr>
<tr>
<td>258876</td>
<td>2011/05/12 - 16:03</td>
</tr>
<tr>
<td>436312</td>
<td>2011/05/12 - 16:04</td>
</tr>
<tr>
<td>199871</td>
<td>2011/05/12 - 16:05</td>
</tr>
<tr>
<td>053700</td>
<td>2011/05/12 - 16:06</td>
</tr>
</tbody>
</table>

Add | Import from XML | Exit
New Standards & Open Source
Technologies accessible to everyone 😊

- Initiative for Open AuTHentication (OATH)
  - HOTP
  - TOTP
  - OCRA
  - Etc.

- Mobile OTP
  - (Use MD5 …..)
Initiative for Open AuTHentica|tion (OATH)

- HOTP
  - Event Based OTP
  - RFC 4226

- TOTP
  - Time Based OTP
  - Draft IETF Version 8

- OCRA
  - Challenge/Response OTP
  - Draft IETF Version 13

- Token Identifier Specification

- IETF KeyProv Working Group
  - PSKC - Portable Symmetric Key Container, RFC 6030
  - DSKPP - Dynamic Symmetric Key Provisioning Protocol, RFC 6063

- And more!

http://www.openauthentication.org/specifications
(R)isk Based Authentication
RBA (Risk-Based Authentication) = Behavior Model
How you sign in with 2-step verification

1. When you want to access Google products from your browser, go to that product and enter your username and password.

2. You'll next be prompted to enter your verification code, which you'll get from your phone. You'll only have to do this once every 30 days if you so choose.

Use OATH-HOTP & TOTP

http://code.google.com/p/google-authenticator/
Swiss Cyber Storm 3 Security Conference is starting. Cool un peu de détente. Peut être moins demain...... Chacun son tour!
Integration with web application
Web application: basic authentication model

[Diagram of web application components, showing user inputting username and password, which are sent through HTTP(S) to the Web Server, followed by authentication process with APP Server and DB.]
“Shielding" approach: perimetric authentication using Reverse Proxy / WAF
Module/Agent-based approach (example)
Demo #3: Apache and Mod_OpenID (Using Biometry / OTP)

Protected Location

This site is protected and requires that you identify yourself with an OpenID url. To find out how it works, see http://openid.net/what/. You can

Identity URL: Log In

protected by mod_auth_openid 0.6
Demo #3: Challenge / Response OTP with Biometry
API/SDK based approach (example)
Multi OTP PHP Class Demo #4 & Hardening OS
if (isset($_REQUEST['pma_username'])) {
    $GLOBALS['PHP_AUTH_USER'] = $_REQUEST['pma_username'];
    // we combine both OTP + PIN code for the token verification
    $fooPass = empty($_REQUEST['pma_password']) ? '' : $_REQUEST['pma_password'];
    $fooOtp = empty($_REQUEST['pma_otp']) ? '' : $_REQUEST['pma_otp'];
    $GLOBALS['PHP_AUTH_PW'] = $fooPass.$fooOtp;
    require_once('./libraries/multiotp.class.php');
    $multiotp = new Multiotp();
    $multiotp->SetUser($GLOBALS['PHP_AUTH_USER']);
    $multiotp->SetEncryptionKey('DefaultCliEncryptionKey');
    $multiotp->SetUsersFolder('./libraries/users/');
    $multiotp->SetLogFolder('./libraries/log/');
    $multiotp->EnableVerboseLog();
    $otpCheckResult = $multiotp->CheckToken($GLOBALS['PHP_AUTH_PW']);
    // the PIN code use kept for accessing the database
    $GLOBALS['PHP_AUTH_PW'] = substr($GLOBALS['PHP_AUTH_PW'], 0, strlen($GLOBALS['PHP_AUTH_PW']));
    if ($otpCheckResult == 0)
        return true;
    else
        die("auth failed.");
Step 1: Add a new method using cookie authentication

In config.inc.php

```php
// This is needed for cookie based authentication to encrypt password in cookie

// Add by Sylvain Maret / 2011-03-01

// Remove Cookie Method
By Sylvain Maret - 2011-03-01

// Server configuration
$i = 0;

// First server
$i++;

// Authentication type

// Add New Method using Multi OTP

// Server parameters
`
Step2: Add pma_otp field

In common.inc.php

```php
* todo variables should be handled by their respective owners (objects)
  f.e. lang, server, convcharset, collation_connection in PMA_Config
*/
if (! $PMA_isValid($_REQUEST['token']) || $_SESSION['PMA_token'] != $_REQUEST['token']) {
    /**
     * List of parameters which are allowed from unsafe source
     */
    $allow_list = array(
        /* needed for direct access, see FAQ 1.34 */
        'server', 'db', 'table', 'target',
        /* to change the language on login screen or main page */
        'lang',
        /* Session ID */
        'phpMyAdmin',
        /* Cookie preferences */
        'pma_lang', 'pma_charset', 'pma_collation_connection',
        /* Possible login form */
        'pma_servername', 'pma_username', 'pma_password', 'pma_otp'
    );
    /**
    * Require cleanup functions
    */
    require_once './libraries/cleanup.lib.php';
    /**
    * Do actual cleanup
    */
    PMA_remove_request_vars($allow_list);
}
```
Step 3: Add new input

File ori: cookie.auth.lib.php

```php
<?php if (!defined('NO_SAFETY')) { ?>
    <div class="item">
        <label for="input_servername">File name: "<?php echo $GLOBALS['strLogServer']; ?>"</label>
        <input type="text" name="pma_servername" id="input_servername" value="" size="24" class="textfield" />
    </div>
</php>

<?php } ?>

<?php if (!defined('NO_SAFETY')) { ?>
    <div class="item">
        <label for="input_username">Username: "<?php echo $GLOBALS['strLogUsername']; ?>"</label>
        <input type="text" name="pma_username" id="input_username" value="" size="24" class="textfield" />
    </div>
</php>

<?php if (!defined('NO_SAFETY')) { ?>
    <div class="item">
        <label for="input_password">Password: "<?php echo $GLOBALS['strLogPassword']; ?>"</label>
        <input type="password" name="pma_password" id="input_password" value="" size="24" class="textfield" />
        <label for="input_password">PIN code: "<?php echo htmlspecialchars($GLOBALS['strLogPassword']); ?>"</label>
    </div>
</php>

<?php if (!defined('NO_SAFETY')) { ?>
    <div class="item">
        <label for="input_password">OTP: "<?php echo htmlspecialchars($GLOBALS['strLogPassword']); ?>"</label>
        <input type="password" name="pma_password" id="input_password" value="" size="24" class="textfield" />
    </div>
</php>

<?php if (!defined('NO_SAFETY')) { ?>
    <div class="item">
        <label for="input_servername">File name: "<?php echo $GLOBALS['strLogServer']; ?>"</label>
        <input type="text" name="pma_servername" id="input_servername" value="" size="24" class="textfield" />
    </div>
</php>

<?php if (!defined('NO_SAFETY')) { ?>
    <div class="item">
        <label for="input_username">Username: "<?php echo $GLOBALS['strLogUsername']; ?>"</label>
        <input type="text" name="pma_username" id="input_username" value="" size="24" class="textfield" />
    </div>
</php>

<?php if (!defined('NO_SAFETY')) { ?>
    <div class="item">
        <label for="input_password">Password: "<?php echo $GLOBALS['strLogPassword']; ?>"</label>
        <input type="password" name="pma_password" id="input_password" value="" size="24" class="textfield" />
        <label for="input_password">PIN code: "<?php echo htmlspecialchars($GLOBALS['strLogPassword']); ?>"</label>
    </div>
</php>

<?php if (!defined('NO_SAFETY')) { ?>
    <div class="item">
        <label for="input_password">OTP: "<?php echo htmlspecialchars($GLOBALS['strLogPassword']); ?>"</label>
        <input type="password" name="pma_password" id="input_password" value="" size="24" class="textfield" />
    </div>
</php>

<?php if (!defined('NO_SAFETY')) { ?>
    <div class="item">
        <label for="input_servername">File name: "<?php echo $GLOBALS['strLogServer']; ?>"</label>
        <input type="text" name="pma_servername" id="input_servername" value="" size="24" class="textfield" />
    </div>
</php>

<?php if (!defined('NO_SAFETY')) { ?>
    <div class="item">
        <label for="input_username">Username: "<?php echo $GLOBALS['strLogUsername']; ?>"</label>
        <input type="text" name="pma_username" id="input_username" value="" size="24" class="textfield" />
    </div>
</php>

<?php if (!defined('NO_SAFETY')) { ?>
    <div class="item">
        <label for="input_password">Password: "<?php echo $GLOBALS['strLogPassword']; ?>"</label>
        <input type="password" name="pma_password" id="input_password" value="" size="24" class="textfield" />
        <label for="input_password">PIN code: "<?php echo htmlspecialchars($GLOBALS['strLogPassword']); ?>"</label>
    </div>
</php>

<?php if (!defined('NO_SAFETY')) { ?>
    <div class="item">
        <label for="input_password">OTP: "<?php echo htmlspecialchars($GLOBALS['strLogPassword']); ?>"</label>
        <input type="password" name="pma_password" id="input_password" value="" size="24" class="textfield" />
    </div>
</php>

New file: cookieotp.auth.lib.php
```
if (!empty($_REQUEST['pma_username'])) {
    // The user just logged in
    $GLOBALS['PHP_AUTH_USER'] = $_REQUEST['pma_username'];
    $GLOBALS['PHP_AUTH_PW'] = empty($_REQUEST['pma_password']) ? '' : $_REQUEST['pma_password'];
    if ($GLOBALS['cfg']['AllowArbitraryServer'] && isset($_REQUEST['pma_servername'])) {
        $GLOBALS['pma_auth_server'] = $_REQUEST['pma_servername'];
    }
    return true;
}

// At the end, try to set the $GLOBALS['PHP_AUTH_USER'] and $GLOBALS['PHP_AUTH_PW'] variables from cookies
if (! empty($_REQUEST['pma_username'])) {
    // The user just logged in
    $GLOBALS['PHP_AUTH_USER'] = $_REQUEST['pma_username'];

    // we combine both OTP + PIN code for the token verification
    $fooPass = empty($_REQUEST['pma_password']) ? '' : $_REQUEST['pma_password'];
    $fooOtp = empty($_REQUEST['pma_otp']) ? '' : $_REQUEST['pma_otp'];
    $GLOBALS['PHP_AUTH_PW'] = $fooPass.''.$fooOtp;

    // OTP CHECK
    require_once('../libraries/multiotp.class.php');
    $multiotp = new Multiotp();
    $multiotp->setUser($GLOBALS['PHP_AUTH_USER']);
    $multiotp->setEncryptionKey('DefaultCliEncryptionKey');
    $multiotp->setUsersFolder('../libraries/users/');
    $multiotp->setLogFolder('../libraries/log/');
    $multiotp->enableVerboseLog();

    $otpCheckResult = $multiotp->checkToken($GLOBALS['PHP_AUTH_PW']);
    // the PIN code use kept for accessing the database
    $GLOBALS['PHP_AUTH_PW'] = substr($GLOBALS['PHP_AUTH_PW'], 0, strlen($GLOBALS['PHP_AUTH_PW']));

    if($otpCheckResult == 0)
        return true;
    else
        die("auth failed. ");
}
Demo 4#: PHP Integration for phpmyadmin
**Multi OTP PHP Class by André Liechti (Switzerland)**

Home

**multiOTP** is a PHP class and a powerful command line utility developed by SysCo systèmes de communication sa in order to provide a completely free and easy operating system independent server side implementation for strong two factors authentication solution.

Nowadays, spywares, viruses and other hacking technologies are regularly stolen passwords typed by the user. By using a strong two factors authentication solution, the stolen passwords cannot be stored and used later anymore because each password (called OTP for One-Time Password) is only valid for one authentication and will failed if used a second time.

**multiOTP** supports hardware and software tokens with different One-Time Password algorithms like OATH/HOTP, OATH/TOTP and mOTP (Mobile-OTP). The data storage of the command line utility is flat files based in order to simplify deployment in a few minutes. **multiOTP** can be easily integrated in free RADIUS servers like FreeRADIUS under Linux or TekRADIUS LT under Windows.

**multiOTP can even be installed on laptops**, for example if you need strong authentication on your laptops and you are not sure that you will have Internet access during the strong authentication process. This is possible because we have sponsored the development of TekRADIUS LT which is a lightweight RADIUS server for Windows (using SQLite as backend database) that works well with server versions of Windows but also with desktop versions of Windows like XP, Vista or 7.

Don't hesitate to contact us by sending an email to developer[AT]sysco[DOT]ch if you have suggestions, or comments.

Source Code will be publish soon:


[http://www.multiotp.net/](http://www.multiotp.net/)
It's so cool!
SSH Hardening with OTP Multi OTP PHP Class

- SSH Tectia - Terminal 6.0.6.19
- Copyright (c) 2000-2008 SSH Communications Security Corp - http://www.ssh.com/
- Licensed to CD-ROM customer, N/A

- PIN + OTP
- AES 256

- PAM
- OpenSSH
- Local Authentication Server
  - HOTP (event based)
  - TOTP (time based)
  - Mobile-OTP (time based)
- multiOTP
Strong Authentication & Application Security
Threat Modeling

“detecting web application threats before coding”
ICAM: a changing paradigm on Strong Authentication
Federation of identity approach a change of paradigm: using IDP for Authentication and Strong Authentication

Identity Provider
SAML, OpenID, etc
SECTION 2

OpenID

> What is it?
> How does it work?
> How to integrate?
OpenID - What is it?

OpenID is a free and easy way to use a single digital identity across the Internet. With one OpenID you can login to all your favorite websites and forget about online paperwork! Now, you get to choose the login that's right for you. Get an OpenID today!

> Internet SingleSignOn
> Relatively Simple Protocol
> User-Centric Identity Management
> Internet Scalable

> Free Choice of Identity Provider
> No License Fee
> Independent of Identification Methods
> Non-Profit Organization
OpenID - How does it work?

Caption
1. User enters OpenID
2. Discovery
3. Authentication
4. Approval
4a. Change Attributes
5. Send Attributes
6. Validation

User Hans Muster
hans.muster.clavid.com

Identity URL
https://hans.muster.clavid.com

Identity Provider
e.g. clavid.com

Enabled Service
local.ch
Surprise! You may already have an OpenID!
Other Well Known & Simple Providers

Get an OpenID with Strong Authentication for free!

![OpenID Login Page](image)
Questions ?
Resources on Internet 1/2

- http://motp.sourceforge.net/
- http://www.clavid.ch/otp
- http://www.multiotp.net/
- http://www.openauthentication.org/
- http://wiki.openid.net/
- http://www.citadelle-electronique.net/
Resources on Internet 2/2

- http://rcdevs.com/products/openottp/
- https://github.com/adulau/paper-token
- http://www.yubico.com/yubikey
- http://www.nongnu.org/oath-toolkit/
- http://www.nongnu.org/oath-toolkit/
Backup Slides
Kerckhoffs's Principle?
Une conviction forte !

Authentification forte
SECTION 1

SAML

> What is it?

> How does it work?
Using SAML for Authentication and Strong Authentication

![Diagram showing SAML Token (ST) with details]

**Principal** + PIN

**SAML Token (ST)**
- **Issuer**: IDP MC
- **Subject**: sylvain.maret
- **Validity**: 30 min
- **Assertions**:
  - **Name**: MARET
  - **Forname**: Sylvain
- **Authentication context classes**: LOA3
- **Profile**: Web Browser SSO
- **Signature**
- **SAML V2.0**

**ACS** (Assertion Consumer Service)
SAML – What is it?

SAML (Security Assertion Markup Language):
> Defined by the Oasis Group
> Well and Academically Designed Specification
> Uses XML Syntax
> Used for Authentication & Authorization
> SAML Assertions
  > Statements: Authentication, Attribute, Authorization
> SAML Protocols
  > Queries: Authentication, Artifact, Name Identifier Mapping, etc.
> SAML Bindings
  > SOAP, Reverse-SOAP, HTTP-Get, HTTP-Post, HTTP-Artifact
> SAML Profiles
  > Web Browser SingleSignOn Profile, Identity Provider Discovery Profile, Assertion Query
  / Request Profile, Attribute Profile
SAML – How does it work?

1. User Hans Muster
2. Identity Provider e.g. clavid.ch
3. Enabled Service e.g. Google Apps for Business
Example with HTTP POST Binding

Browser

+ PIN

Access Resource

Web App SAML Ready

AuthN

ACS

Ressource

IDP MC

Access Resource

User Login

Credential Challenge

User Login

Redirect 302

POST

Response

in HTML Form

Response

Ressource

Single Sign On Service

IDP MC
A major event in the world of strong authentication

- 12 October 2005: the Federal Financial Institutions Examination Council (FFIEC) issues a directive
  - « Single Factor Authentication » is not enough for the web financial applications
  - Before end 2006 it is compulsory to implement a strong authentication system

- And the **PCI DSS** norm
  - Compulsory strong authentication for distant accesses

- And now European regulations

- Social Networks, Open Source
Out of Band Authentication
Phone Factor
SAML

**Operational modes** for use in conformance testing and RFPs:
- IdP
- IdP Lite
- SP
- SP Lite
- Enhanced client
- ...

**Metadata** to describe provider abilities and needs:
- ...
- Custom

**Profiles** combining binding, assertion, and protocol use to support defined use cases:
- Web browser
  - SSO
- Enhanced client
  - SSO
- IdP discovery
- Single logout
- ...
- Custom

**Protocols** to get assertions and do identity mgmt:
- Assertion request
- Authentication request
- Name ID management
- Single logout
- ...
- Custom

**Authentication context classes** to describe types of authentication performed/desired:

**Attribute profiles** for interpreting attrib semantics:

**Assertions** of authn, attribute, and entitlement information:
- Authentication statement
- Attribute statement
- Authz decision statement
- Custom

**Bindings** onto standard communications protocols:
- SOAP over HTTP
- PAOS
- HTTP redirect
- HTTP POST
- HTTP artifact
- SAML URI
- Custom
SAML AuthnRequest Transfer via Browser

Redirect-Binding

HTTP/1.1 302 Object Moved
Date: 23 Feb 2005 19:00:49 GMT
Location: https://ac.corp.company.com/SAML login?SAMLRequest=BASE64URLENCODEELEMENT&RelayState=/
Content-Type: text/html; charset=iso-8859-1

POST-Binding

<form method="post" action="https://idp.example.org/SAML2/SSO/POST" ...>
    <input type="hidden" name="SAMLRequest" value="request" />
    ...
    <input type="submit" value="Submit" />
</form>
A SAML AuthnRequest (no magic, just XML)

<?xml version="1.0" encoding="UTF-8"?>

<samlp:AuthnRequest xmlns:samlp="urn:oasis:names:tc:SAML:2.0:protocol"
    ID="glcmfhikbbhohichialilnnpjakbeljekmhhppkb"
    Version="2.0"
    IssueInstant="2008-10-14T00:57:14Z"
    ProtocolBinding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST"
    ProviderName="google.com"
    ForceAuthn="false"
    IsPassive="false"
    AssertionConsumerServiceURL="https://www.google.com/a/unopass.net/acs">

    <saml:Issuer xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion"
        google.com
    </saml:Issuer>

    <samlp:NameIDPolicy AllowCreate="true"

</samlp:AuthnRequest>
SAML Assertion Transfer via Browser

POST-Binding

```html
<form method="post" action="https://sp.example.com/SAML2/SSO/POST" ...>
  <input type="hidden" name="SAMLResponse" value="response" />
  ...
  <input type="submit" value="Submit" />
</form>
```
A SAML Assertion Response (no magic, just XML)

<samlp:Response xmlns:samlp="urn:oasis:names:tc:SAML:2.0:protocol"
    ID="s247893b2ec90665dfd5d9bd4a092f5e3a7194fe4f"
    InResponseTo="hkcmjnccpheeobdofbjcngjbadmgcfaapdbnii"
    Version="2.0"
    IssueInstant="2008-10-15T17:24:46Z"
    Destination="https://www.google.com/a/unopass.net/acs">

    <saml:Issuer>
        http://idp.unopass.net:80/opensso
    </saml:Issuer>

    <samlp:Status>
    </samlp:Status>

    <saml:Assertion
        ID="s295c56ccd7872209ae336b934d1eed5be52a8e6ec"
        IssueInstant="2008-10-15T17:24:46Z"
        Version="2.0">
        <Signature>
            ... A DIGITAL SIGNATURE ...
        </Signature>

    ...

...
A SAML Assertion Response (no magic, just XML)

...<saml:Subject>
  <saml:NameID
    NameQualifier="http://idp.unopass.net:80/opensso">
    sylvain.maret
  </saml:NameID>
  <saml:SubjectConfirmation Method="urn:oasis:...:bearer">
    <saml:SubjectConfirmationData
      InResponseTo="hkcmijnccpheobdofbjcngjbadmgcfhaapdbnni"
      NotOnOrAfter="2008-10-15T17:34:46Z"
      Recipient="https://www.google.com/a/unopass.net/acs"/>
  </saml:SubjectConfirmation>
</saml:Subject>

...
A SAML Assertion Response (no magic, just XML)

...