KAFE

서비스 제공자의 구축

- Korean Access Federation (KAFE) -

한국과학기술정보연구원

KAFE

목 차

KAFE

SAML 서비스 제공자 시스템의 구축

2015. 09. 09. - 초안

제 1 장 설치 환경

본 설치매뉴얼은 simpleSAMLphp 1.13 버전을 이용해 Ubuntu 또는 CentOS 환경에서 SAML 2.0 SP(Service Provider)를 구축하는 방법을 기술한다. SP 의 구축을 위해 다음과 같은 요구조건이 충족되어야 한다.

- LAMP 스택의 설치: Apache, MySQL, PHP 5.3 이상

- 공인인증서(SSL)의 설치

제 2 장 simpleSAMLphp 의 설치

2.1 절 simpleSAMLphp

simpleSAMLphp 는 UNINETT 에서 개발한 SAML v2.0 소프트웨어이다. IdP(Identity Provider) 또는 SP(Service Provider)로 설치가능하며 최신 버전은 1.13.2 이다(2015 년 9 월). 이하 IdP 또는 SP 는 SAML 2.0 IdP 또는 SAML 2.0 SP 를 의미한다. simplesamlphp.org 에서 추가적인 정보를 얻을 수 있다.

2.2 절 simpleSAMLphp 의 설치 환경

simpleSAMLphp 의 설치를 위한 서버 환경은 다음과 같다. 특별한 언급이 없는 한 서비스 제공자용 서버는 Ubuntu 14.04 LTS(64 비트)를 이용한다.

- php, MySQL, httpd 가 설치

- IPv6 disable 권장

- selinux disable

- Linux 방화벽(iptables) 80/443(http/https) 포트 개방

- 시간 동기화를 위한 NTP 설정(NTP 서버: time.kriss.re.kr)

2.3 절 simpleSAMLphp 의 설치

simpleSAMLphp 의 구동을 위해 요구되는 소프트웨어 패키지를 설치한다. simpleSAMLphp 의 설치 경로는 /var/simplesamlphp 로 가정한다.

```
~# clear
```

```
~# sudo apt-get install php-date openssl php5-mcrypt
// 인증 소스로 LDAP를 이용하는 경우
~# sudo apt-get install php5-ldap


//simplesamlphp 다운로드
~#sudo wget https://simplesamlphp.org/res/downloads/simplesamlphp-1.13.2.tar.gz


//압축해제 및 설치
~# sudo cp ./simplesamlphp-1.13.2.tar.gz     /var/
~# sudo cd   /var
~# sudo tar zxvf ./simplesamlphp-1.13.2.tar.gz
~# sudo mv ./simplesamlphp-1.13.2 ./simplesamlphp
```

※ CentOS 6.7 (php 5.5.30)에서 mcrypt 설치방법

```
//기존 php 5.5.30을 모두 지웠다고 가정한다.
~# rpm -Uvh https://mirror.webtatic.com/yum/el6/latest.rpm
~# yum install php56w php56w-opcache php56w-mcrypt php56w-xml php56w-mysql
~# service httpd restart
```

2.4 절 Apache 서버 설정

아래 설정 방법은 HTTP(80 포트)에 대한 환경설정을 보여준다. HTTPS(443)에 대한 Apache 환경설정 방법은 생략한다.
※ ID 제공자 서버는 반드시 공인인증서를 설치하고 HTTPS(443)을 이용해야 한다.

```
~# sudo cd /etc/apache2/sites-available
~# sudo   nano   000-default.conf
// <VirtualHost *:80>을 찾아 아래와 같이  수정
<VirtualHost *:80>
        DocumentRoot    /var/www/html/
        Alias      /simplesaml        /var/simplesamlphp/www


~# sudo nano /etc/apache2/apache2.conf
// 다음과 같은 항목을 추가
```

```
<Directory /var/simplesamlphp/>
        Options Indexes FollowSymLinks
        AllowOverride None
        Require all granted
</Directory>

~# sudo service apache2 restart
```

2.5 절 simpleSAMLphp 의 구동환경 설정

아래와 같이 simpleSAMLphp 를 환경설정한다. 'secretsalt'는 다음 명령을 이용해 추출할 수 있다.

tr -c -d '0123456789abcdefghijklmnopqrstuvwxyz' </dev/urandom | dd bs=32 count=1 2>/dev/null;echo

```
~# sudo cd   /var/simplesamlphp/config
~# sudo nano config.php
// 아래와 같이 환경설정 함
'baseurlpath' => 'simplesaml/',
'certdir' => 'cert/'
'loggingdir'        => 'log/',
'datadir' => 'data/',

// 다음 사항은 꼭 수정해야 함
'auth.adminpassword'      => '[관리용 패스워드 입력]',
'secretsalt' => '[secret salt 입력]',
'technicalcontact_name' => '[관리자 이름]',
'technicalcontact_email' => '[관리자 이메일]',
'language.default' => 'en',
'timezone' => 'Asia/Seoul',
```

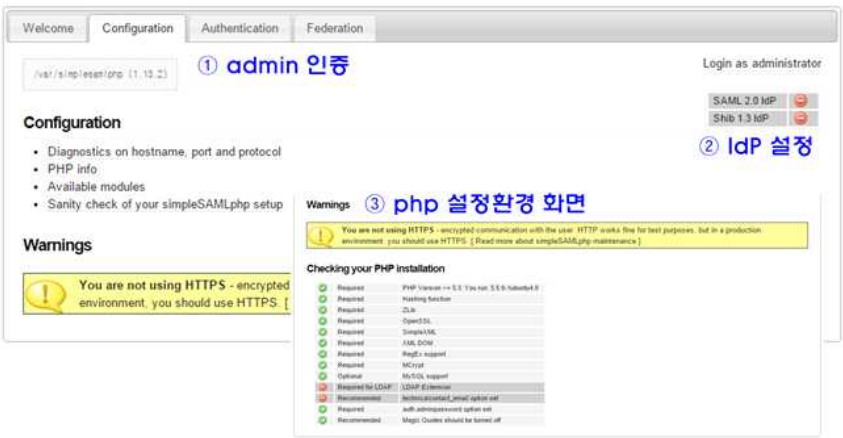
simpleSAMLphp 에서 제공하는 특정 모듈을 활성화하고 싶다면 [모듈명] 디렉토리에서 enable 파일을 생성한다. 다음 예는 LDAP 모듈을 활성화하기 위한 방법이다.

```
~# sudo   cd    /var/simplesamlphp/modules
~# sudo   cd   ./ldap
~# sudo   touch    enable
```

환경 설정이 완료되었다면 웹 브라우저를 이용해 http://[서버주소]/simplesaml 을 접속한다. 정상적으로 설치되었다면 아래와 같은 화면이 나타난다.

2.6 절 설정된 환경의 검증

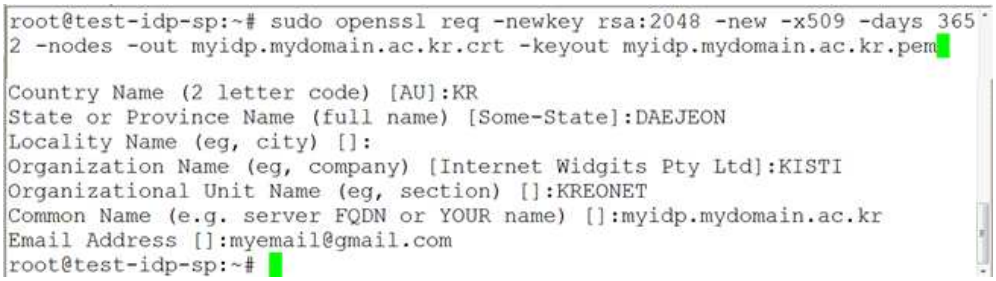
Authentication 탭에서 admin 으로 로그인 한다. 현재 IdP 가 활성화되지 않은 상태이므로 ②와 같이 적색원이 보여야 한다. admin 으로 로그인한 후 ③과 같이 표시된다면 정상적으로 설치된 상태이다. 관리자 이메일과 LDAP Extension 을 설치했다면 모두 녹색원으로 표시되어야 한다.



제 3 장 SAML 서비스 제공자의 설치

3.1 절 SSL 자가 인증서의 생성

아래 [myidp.mydomain.ac.kr]은 SP 구축자의 기관 환경에 맞게 적절히 변경해야 한다. Common Name (e.g., server FQDN or YOUR name)은 SP 용 서버의 IP 주소 또는 도메인명으로 설정해야 한다.



자가 인증서를 생성한 후 환경설정을 계속한다.

```
~# sudo   cd    /var/simplesamlphp
~# sudo   mkdir cert
// 생성된 .cert 파일과 .pem 파일은 /var/simplesamlphp/cert 디렉토리로 이동
```

SP 의 메타데이터를 수정해 자가 인증서를 등록한다.

```
~# sudo   cd    /var/simplesamlphp/config
~# sudo   nano  authsources.php

// 'default-sp' => array에 아래와 같이 수정해 인증서를 등록한다.
'privatekey'        => '/var/simplesamlphp/cert/[인증서이름].pem',
'certificate'       => '/var/simplesamlphp/cert/[인증서이름].crt',
```

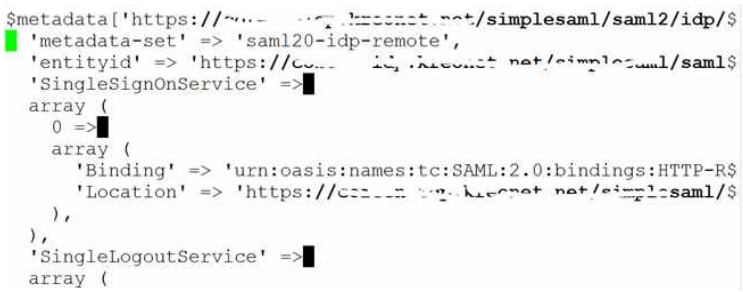
3.2 절 메타데이터 설정 및 IdP 메타데이터의 등록

SP 의 entityID 를 설정한다. entityID 는 http(s)://domain_name/sp/saml_software 의 형식에 따른다.

```
~# sudo   cd    /var/simplesamlphp/config
~# sudo   nano  authsources.php

// 'default-sp' => array에 아래와 같이 entityID 값을 변경한다.
'entityID'          => 'https://myidp.mydomain.ac.kr/sp/simplesamlphp',
```

SP 와 IdP 를 ID 연계하기 위해서, IdP 와 SP 간 각각의 메타데이터를 교차 등록해야 한다. SP 에 IdP 의 메타데이터를 등록하는 방법은 다음과 같다. IdP 의 메타데이터 파일을 확보하고 있다고 가정한다.

```
~# sudo   cd    /var/simplesamlphp/metadata

~# sudo   nano   saml20-idp-remote.php


// 아래 그림(예시)처럼 IdP의 메타데이터를 추가한다.
```



※ simpleSAMLphp 기반의 IdP 를 보유하고 있다고 가정했을 때, http://[IdP 서버주소]/simplesaml 에 접근해 Federation 텝을 클릭하면 IdP 의 메타데이터 정보를 확인할 수 있다.

위 그림처럼 SP 에 IdP 의 메타데이터를 등록하고 등록된 메타데이터에 이름 'name'을 추가한다. 등록된 IdP 의 메타데이터에 'name'이 이미 등록되어 있는 경우에는 'name' 등록을 생략한다. 'name'의 내용은 사용자가 SP 에서 IdP 를 선택할 때(IdP discovery), 목록의 형태로 보여진다.

```
~# sudo   cd    /var/simplesamlphp/metadata

~# sudo   nano   saml20-idp-remote.php


// 아래 그림(예시)처럼 IdP의 메타데이터를 추가한다.

$metadata['https://[IdP 주소]/idp/simplesamlphp'] = array(

  'name' => array(

            'en' => '[IdP의 이름]',

  ),
```

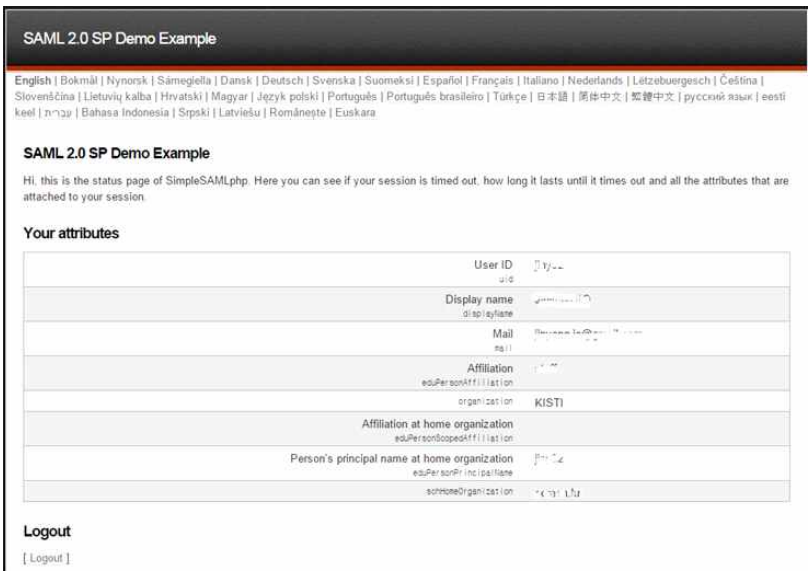
SP 의 메타데이터도 IdP 에 등록되어야 SP-IdP 간 SAML 통신이 가능하다. 아래 그림처럼 http://[SP 서버주소]/simplesaml 에 접근해 Federation 탭을 클릭하면 SP 의 메타데이터 정보를 확인할 수 있다.

SP 의 메타데이터 정보를 복사해서 IdP 에 메타데이터 등록을 요청해야 한다. IdP 의 /var/simplesamlphp/metadata/saml20-sp-remote.php 파일에 SP 의 메타데이터를 등록할 수 있다. simpleSAMLphp 는 평문(flat format) 형태의 메타데이터와 XML 형태의 메타데이터를 함께 제공한다. ID 연계되는 상대 IdP 또는 SP 에 자신의 메타데이터를 등록할 때는 평문 형태의 메타데이터를 이용한다.

SP 에 IdP 의 메타데이터가 등록되었다면 아래 그림과 같이 http://[SP 의 주소]/simplesaml 에 접속해서 Authentication → Test authentication sources 의 'default-sp'를 클릭한다.

아래 그림과 같이 Select your identity provider 에서 saml20-idp-remote.php 의 'name'으로 등록한 이름을 클릭한다. 본 예시에서는 'name'을 'Coreen IdP -guest users'로 설정했다.

IdP 에 등록된 사용자 ID 와 비밀번호를 이용해 로그인하면 아래 그림과 같이 IdP 가 제공하는 사용자 속성정보를 확인할 수 있다.



제 4 장 메타데이터의 설정

4.1 절 oid to name 변환

SAML 2.0 는 Attribute 이름의 oid 표기를 권장한다. ID 제공자가 oid 형태(e.g.,urn:oid:2.5.4.4)의 속성이름을 제공하고 서비스 제공자가 friendly name(e.g., sn) 을 이용해 사용자를 인가한다면 oid 를 friendly name 으로 변경해야 한다. KAFE 는 oid 표기법을 권장하므로 SP 에서 다음과 같이 oid2name 변환을 한다.

```
~# cd     /var/simplesamlphp/config
~# nano   config.php


// 'authproc.sp' => array( 가 포함된 라인을 찾아 다음과 같이 추가한다.
50 => array(
        'class' => 'core:AttributeMap',
        'oid2name',
        // 서비스 제공자에서 사용자 인가를 위해 friendly name 'o' 대신에 friendly name 'organizationName'
           을 이용한다면 다음 라인을 추가한다.
        'o'   => 'organizationName',
),
```

제 5 장 웹 응용과 SAML SP 의 연동

지금부터는 웹 응용과 SAML SP 를 연동하는 방법에 대해서 기술한다.

5.1 절 연동 시 고려 사항

웹 응용은 simpleSAMLphp 가 제공하는 API(Application Programming Interface)를 이용해 사용자를 인가(Authorization)해야 한다. KAFE(Korean Access Federation)에서는 아래 표와 같은 속성들의 이용을 권장하고 있다.

제공 속성	설명	개인정보 가능성
uid	사용자 ID(시험 서비스 기간에 한시적 적용)	O
eduPersonTargetedID	서비스 제공자 별 암호화된 사용자 고유번호	
sn	성	O
givenName	이름	O
displayName	사용자의 화면표시 이름	O
mail	사용자 이메일 주소	O
eduPersonAffilation	사용자의 기관내 직무정보	
organizationName	사용자의 소속기관명	
schacHomeOrganization	사용자 소속기관의 최상위 도메인 이름	
eduPersonPrincipalName	도메인 내 사용자 ID 정보	O
eduPersonScopedAffilation	도메인 내 사용자 직무 정보	

IdP 가 제공하는 속성 정보는 위 표에 명시된 속성들 보다 확장될 수 있다. 웹 응용이 SAML SP 연동 시 고려해야 할 사항은 다음과 같다.

- 사용자를 구분하는 방법; SP 는 사용자를 구분할 수 있는(또는 사용자 충돌을 피할 수 있는) 방법을 준비해야 한다. 다수의 IdP 에 동일한 사용자 식별자값(예; uid 등)이 존재할 가능성이 있다. SP 는 다수의 사용자 속성 또는 메타데이터를 통해 얻은 값들을 이용해 사용자를 구분할 수 있어야 한다.

5.2 절 사용자 인증 및 인가 관련 웹 응용 코드

simpleSAMLphp 가 /var/simplesamlphp/에 설치되어 있고 SP 로 동작한다고 가정한다. 또한 웹 응용의 root 디렉토리가 /var/www/html 이며 php 구동을 위한 Apache 환경설정이 완료되어 있다고 가정한다. IdP 에 등록된 계정은 student/student1234, faculty/faculty1234 이다.

다음은 웹 응용의 skeleton code 이다. 로그인한 사용자의 속성정보 및 사용자 인증을 수행한 IdP 의 정보를 배열의 형태로 얻게 된다.

```
~# cd /var/www/html
~# nano index.php

// 다음과 같이 추가
<?php
        include_once('/var/simplesamlphp/lib/_autoload.php');
        $as = new SimpleSAML_Auth_Simple('default-sp');
        $as->requireAuth();

        $attributes = $as->getAttributes();
        print_r($attributes);

        $idp = $as->getAuthData('saml:sp:IdP');
        print_r($idp);
?>
```

SP 가 IdP 와 연계되어 있고 각각의 메타데이터가 교차 등록되었을 경우, http://[SP 의 주소]/로 접속하면 아래와 같이 ID 제공자를 선택하는 화면이 나타난다.

사용자가 로그인에 성공할 경우 아래와 같이 사용자 속성이름과 속성값들이 배열형태로 리턴 된다.

```
// 로그인에 성공한 후 결과값 출력 예시
Array( [uid] => Array( [0] => student ) [displayName] => Array( [0] => my name) ..... )
https://[IdP의 주소]/idp/simplesamlphp
```

다음은 권한부여(또는 인가)를 위한 예제 코드이다. ID 제공자가 전달한 속성 정보 중 eduPersonAffilation 이 'faculty', organizationName 이 'DGIST', eduPersonPrincipalName 이 'faculty@coreen.kr'일 경우에 관리자 권한을 갖는 예제이다. 사용자 인가에 사용할 속성은 서비스 상황에 맞게 선택할 수 있다.

```
~# cd /var/www/html
~# nano index.php
```

```
// 다음과 같이 변경
<?php
        include_once('/var/simplesamlphp/lib/_autoload.php');
        $as = new SimpleSAML_Auth_Simple('default-sp');
        $as->requireAuth();

        $attributes = $as->getAttributes();
        $idp = $as->getAuthData('saml:sp:IdP');

        $uid = $attributes['uid'][0];
        $displayName = $attributes['displayName'][0];
        $mail = $attributes['mail'][0];
        $eduPersonAffilation = $attributes['eduPersonAffiliation'][0];
        $organizationName = $attributes['organizationName'][0];
        $schacHomeOrganization = $attributes['schacHomeOrganization'][0];
        $eduPersonPrincipalName = $attributes['eduPersonPrincipalName'][0];
        $eduPersonTargetedID = $attributers['eduPersonTargetedID'][0];

        //authorization
        if ($eduPersonAffilation === 'faculty' && $organizationName === 'DGIST' &&
$eduPersonPrincipalName === 'faculty@coreen.kr'){
                $isAdmin = 1;
        }else{
                $isAdmin = 0;
        }

        if($isAdmin){
                echo "Welcome Prof. ".$displayName."!!<br>";
                echo "You are allowed to access IT resource 1, 2, and 3.";
        }else{
                echo "Welcome Student ".$displayName."!!<br>";
```

```
                echo "You are allowed to access IT resource 1 only.";
        }
?>
```

5.3 절 로그인 및 로그아웃

다음은 SURFnet 에서 제공하는 simpleSAMLphp 용 SP 의 예제 코드이다. ID 연계를 위한 SAML 메시지 중개시스템인 OpenConext 와 연동해 보다 강력한 사용자 인증을 수행하기 위한 코드이다. SURFnet 의 ID 연계구조는 Hub&spoke 이지만 KREONET 의 KAFE 는 Full mesh 구조를 갖기 때문에 메시지 중개시스템을 이용하지 않는다. 아래 코드의 LOA(Level Of Assurance) 관련 부분은 KAFE 에서 사용되지 않는다.

```
//source code provided by SURFnet
//https://wiki.surfnet.nl/display/SUAAS/Configuring+a+simpleSAMLphp+SP+for+step-up+authentication


<?php
// Include SimpleSAMLphp. Assume this script is placed in the <simplesaml>/www dir.
require_once('../lib/_autoload.php');


// Name of session variable for storing the min required LOA(Level of Assurance) for a login
define( 'SSP_SESSION_MIN_LOA', 'RequestedMinLOA' );


// Build return URL. This is where ask simplesamlPHP to direct the browser to after login or logout
// Point to this script, but without any request parameters so we won't trigger an login again (and again,
and again, and ...)
$returnURL = ($_SERVER['HTTPS'] == 'on') ? 'https://' : 'http://';
$returnURL .= $_SERVER['HTTP_HOST'];
$returnURL .= $_SERVER['SCRIPT_NAME'];


// Map integer level of assurance level to identifier used by the gateway
$gLOAmap = array(
    1 => 'http://suaas.example.com/assurance/loa1',
    2 => 'http://suaas.example.com/assurance/loa2',
    3 => 'http://suaas.example.com/assurance/loa3',
);
```

```
try {
    // Init SP instance
    // Assumes you have setup a SP named "default-sp" in <simplesaml>/config/authsources.php
    // See: https://simplesamlphp.org/docs/stable/simplesamlphp-sp
    $as = new SimpleSAML_Auth_Simple('default-sp');       // Init SP instance

    /** @var $session SimpleSAML_Session */
    $session = SimpleSAML_Session::getInstance();

    // Process login action. Assumes the login function of your SP uses ...?action=login
    if (isset($_REQUEST['action']) && $_REQUEST['action'] == 'login' ) {
        // We use the SSP session to keep track of the LOA we want.
        // Unset any existing RequiredAuthnContextClassRef
        $session->deleteData('string', SSP_SESSION_MIN_LOA);

        // login
        $requiredLOA = 2; // The LOA we want.

        // Store the requested LOA in the session so we can verify it later
        $session->setData('string', SSP_SESSION_MIN_LOA, $requiredLOA);

        $as->login( array(
            'ReturnTo' => $returnURL,
            'ForceAuthn' => false,
            'saml:AuthnContextClassRef' => $gLOAmap[$requiredLOA]   // Specify LOA
        ) );
        exit;     // Never reached. Added for clarity
    }
    // Process logout action
    if( isset($_REQUEST['action']) && $_REQUEST['action'] == 'logout' ) {
        $as->logout( array (
```

```
                'ReturnTo' => $returnURL,
            ) );    // Process logout


            exit;     // Never reached. Added for clarity
        }


        // Display HTML page
        echo <<<head
        <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
        <html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
            <head>
                <meta http-equiv="Content-type" content="text/html;charset=UTF-8" />
                <style type="text/css">
                    table,th,td {border: 1px solid black;}
                    th,td {padding 1px}
                </style>
                <title>simpleSAMLphp Demo</title>
            </head>
            <body>
                <h1>SimpleSAMLphp LOA Demo</h1>
head;


        // Show some info when authenticated
        if ( $as->isAuthenticated() ) {
            $attributes = $as->getAttributes();
            $requestedLoA = $session->getData('string', SSP_SESSION_MIN_LOA);       // What we requested
during login
            $authState = $session->getAuthState();
            $authnConext = $authState['saml:sp:AuthnContext'];
            $nameID = $session->getNameID();
            $authnInstant = gmdate('r', $authState['AuthnInstant'] );
```

```
$expire = gmdate('r', $authState['Expire'] );

echo "<h2>You are logged in</h2>";

echo "<h3>SimpleSAMLphp Session</h3>";
echo "<p>SimpleSAMLphp session start: <b>{$authnInstant}</b></br />";
echo "SimpleSAMLphp session expire: <b>{$expire}</b></p>";

echo "<h3>LOA</h3>";
echo "<p>Received authnConext: <b>{$authnConext}</b></p>";

// Map LoA identifier back to integer LoA level
$actualLoA = array_search($authnConext, $gLOAmap);
if (false ==! $actualLoA)
    echo "<p>Actual LoA is: <b>{$actualLoA}</b></p>";
else
    $actualLoA = -1;

if (NULL !== $requestedLoA) {
    echo "<p>Requested LoA was: <b>{$requestedLoA}</b></p>";
    if ($actualLoA >= $requestedLoA)
        echo '<p><b>You were authenticated at or above the minimally required LoA</b></p>';
    else
        echo '<p><b>You were NOT authenticated at the required LoA</b></p>';
}

echo <<<html
<h3>NameID</h3>
<table>
    <tr><th>Value</th><td>{$nameID['Value']}</td></tr>
    <tr><th>Format</th><td>{$nameID['Format']}</td></tr>
</table>
```

```
html;

        echo <<<html
        <h3>SAML Attributes</h3>
        <table>
            <tr><th>Attribute</th><th>Value(s)</th></tr>
html;
        foreach ($attributes as $attrName => $attrVal) {
            echo "<tr><td>{$attrName}</td><td>";
            if (is_array($attrVal))
                echo implode('<br />', $attrVal);
            else
                echo $attrVal;
            echo "</td>";
        }
        echo <<<html
        </table>

        <h3>Logout</h3>
        <p>
            <form name="logout" action="{$returnURL}" method="get">
                <input type="hidden" name="action" value="logout"/>
                <input type="submit" value="Logout" />
            </form>
        </p>
html;
    } else {
        echo <<<html
            <h2>Your are not logged in</h2>
html;
    }
```

```
    echo <<<html
            <h3>Login (again)</h3>
            <p>
                <form name="login" action="{$returnURL}" method="get">
                    <input type="hidden" name="action" value="login"/>
                    <input type="submit" value="Login" />
                </form>
            </p>
html;
    echo <<<html
        </body>
    </html>
html;
}
catch (Exception $e)
{
    echo $e->getFile().':'.$e->getLine().' : '.$e->getMessage();
}
```

제 6 장 보안 및 개인정보

6.1 절 페이지 접근제어

simplesamlphp 를 설치하면 일반사용자가 웹 브라우저를 통해 IdP 또는 SP 의 metadata, phpinfo 등 보안정보에 접근할 가능성이 있다. simplesamlphp 를 설치한 후 해당 정보들을 은닉하기 위해 config.php 파일을 수정해야 한다. default themes 의 userloginpass.php 파일을 수정해 사용자 인터페이스를 변경할 수 있다.

```
~# clear
~# cd /var/simplesamlphp/config/config.php
//아래와 같이 protectindexpage와 protectmetadata 값을 true로 변경한다.
 'admin.protectindexpage' => true,
 'admin.protectmetadata' => true,
```

SSP 의 관리자페이지 노출 취약점을 해결하기 위해 apache 설정을 변경해 줘야 한다. CentOS 6.5 기준으로 SP 가 SSL 이 적용되어 있을 때 다음과 같이 설정한다.

```
~# clear
~# cd /etc/httpd/conf.d
~# nano ssl.conf
//</VirtualHost> 앞에 다음과 같이 추가한다. 설치환경에 맞게 적절히 수정되어야 한다.
<Location /simplesaml/module.php/core/loginuserpass.php>
    Order Deny,Allow
    Deny from all
    # 192.168.0.*만 접속 가능
    Allow from 192.168.0.0/24
</Location>
```

6.2 절 Privacy Policy

SP 의 메타데이터에 'privacypolicy'가 설정되어 있으면 consent 에서 해당 privacypolicy 를 링크한다. Consent 화면에는 privacypolicy 의 URL 이 %SPENTITYID%로 변경되어 표시된다.
※ 서비스 제공자는 ID 제공자에게 privacypolicy URL 정보를 전달하고, ID 제공자가 privacypolicy 정보를 설정해야 한다.

```
~# clear
~# nano /var/simplesamlphp/metadata/saml20-sp-remote.php
//특정 SP의 metadata 내에
```

```
'privacypolicy' => 'URL',   // 예; 'privacypolicy' => 'https://yourdomain.com/privacypolicy',
// 이 설정되어 있으면 consent 시 해당 URL이 화면 출력됨
```

6.3 절 SSP의 보안 강화 사항

showerrors 항목을 false로 해서 오류가 발생했을 때 노출되지 않아야 할 오류정보(stacktrace는 시스템 정보를 노출함)가 사이트에 노출되는 것을 방지한다. 또한, admin 비밀번호를 설정해 SSP의 정보가 노출되지 않도록 한다.

```
~# nano /var/simplesamlphp/config/config.php
//5.1과 동일
 'admin.protectindexpage' => true,
//추가 또는 수정
 'showerrors' => false,
```

쿠키 보안을 위해, 평문 연결(plain text connection, Non-TLS)일 때 쿠키 정보가 전송되는 것을 막고 자바스크립트가 쿠키에 접근하는 것을 막아야 한다. TLS(https connection)를 반드시 이용해야 한다. 쿠키 보안 설정을 하지 않으면 Cross Site Scripting 공격에 취약할 수 있다.

```
~# nano /var/simplesamlphp/config/config.php
// 다음과 같이 수정
 'session.cookie.secure' => true,
 'session.phpsession.httponly' => true,
```

SSP가 'redirect'를 할 도메인 이름을 설정한다. 다음과 같이 empty array로 설정하면 SSP가 자동으로 신뢰하는 도메인으로만 redirect 한다.

```
~# nano /var/simplesamlphp/config/config.php
// 다음과 같이 수정
 'trusted.url.domains' => array(),
```

SHA1(보안취약) 대신 SHA-256으로 이용한다. ID 제공자는 saml20-idp-hosted.php를, SP 제공자는 authsources.php를 수정한다. SSP 1.12 이상의 버전에서는 config 파일에 반영되어 있으므로 주석표시만 제거한다.

```
// ID 제공자일 경우에 해당
~# nano /var/simplesamlphp/metadata/saml20-idp-hosted.php
// 다음과 같이 주석제거
'signature.algorithm' => 'http://www.w3.org/2001/04/xmldsig-more#rsa-sha256',
```

[최근 갱신: 2015-11-6 -draft v0.14]