SAML-ing WordPress/Php Application with SimpleSamlphp

Objective

- Install/Setup Apache/Mysql/PHP
- Install/setup WordPress
- Install/setup simplesamlphp
- Install/setup WordPress simplesamlphp plugin
- Provide hybrid protection mechanism for certain pages with WordPress

Below are the details on above bullets.

Install/setup Apache/MySql/PHP

Steps:

1. Setup OS : This setup assumes the OS used is CentOS (6.x). I used CentOS 6.3 VM Image provided by thoughtpolice, http://www.thoughtpolice.co.uk/vmware/
2. Setup Apache
    a. Install Apache
        i. `sudo yum install httpd`
    b. Start Apache Service
        i. `sudo service httpd start`
    c. Make it auto start
        i. `sudo chkconfig mysqld on`
3. Setup Mysql
    a. Install Mysql
        i. `sudo yum install mysql-server`
    b. Start Mysql Service
        i. `sudo service mysqld start`
    c. Make it auto start
        i. `sudo chkconfig mysqld on`
    d. Set mysql root password
        - `run "sudo /usr/bin/mysql_secure_installation" this to reset root password.`
        - `CentOS automates all the process of setting up mysql.`
          `When asked series of questions, you can provide 'yes' as default answer.`

        i. **`The set of questions asked are`**

`Remove anonymous users? [Y/n] y`
`Disallow root login remotely? [Y/n] y`
`Remove test database and access to it? [Y/n] y`
`Reload privilege tables now? [Y/n] y`

4. Setup PHP modules
    a. Install PHP
        i. `sudo yum install php php-mysql`
    b. Check PHP modules
        i. `yum search php-`
    c. Find more info on Modules
        i. `yum info <name of the module>`
    d. Install specific module that you want, For the purpose of this setup we do not need add any extra modules right now.
        i. `sudo yum install name of the module`
5. Check PHP is working fine
    a. Create a test php file
        i. `sudo vi /var/www/html/test.php`
        ii. `Add "<?php phpinfo(); ?>"`

iii. `save it`

b. Restart httpd
   i. `sudo service httpd restart`

c. Test the page, it should be available on http://<your ipaddress>/test.php and should include all the info about PHP setup on the apache
d. You can get your ip address by issuing this command
   i. ifconfig or ifconfig eth0 | grep inet | awk '{ print $2 }'

6. Open port 80 for http connection, by default this is not opened in VM issued
   a. Add iptables entry for port 80
      i. `iptables -I INPUT 5 -i eth0 -p tcp --dport 80 -m state --state NEW,ESTABLISHED -j ACCEPT`

   b. Check Iptables
      i. `iptables --line -vnL`

   c. Save changes
      i. service iptables save
      ii. when asked for save , say "OK"
   d. restart iptables services
      i. `service iptables restart`

7. Add SSL support on apache
   a. Install mod_ssl
      i. `yum install mod_ssl`

   b. create a new directory
      i. `mkdir /etc/httpd/ssl`

   c. create self signed certificate
      i. `openssl req -x509 -nodes -days 730 -newkey rsa:2048 -keyout /etc/httpd/ssl/apache.key -out /etc/httpd/ssl/apache.crt`

         This command asks you to provide various element value for certificate.
   d. setup certificate on apache
      i. Open SSL conf file; vi /etc/httpd/conf.d/ssl.conf
      ii. Add these

```
ServerName example.com:443

SSLEngine on
SSLCertificateFile /etc/httpd/ssl/apache.crt
SSLCertificateKeyFile /etc/httpd/ssl/apache.key
```

   e. Restart apache
      i. `/etc/init.d/httpd restart`

8. Open port 443 for SSL
   a. Follow same steps a for opening port 80
9. ---

At this point you should have cleanly setup Apache, Mysql and PHP.

Install/Setup WordPress

Following are the steps followed to install/setup WordPress after successful install/setup of OS/Apache/Mysql/PHP/SSL

1. Download WordPress from https://wordpress.org/download/. The download used in this instance is 4.0
   a. extract the gz wordpress file and place it under /var/www/html/wp directory. When you extract wordpress-4.0.tar.gz file it put all the file under wordpress folder. Copy all the files/folders within wordpress to /var/www/html/wp directory. Or you can simple place the gz file on /var/www/html then extract then rename "wordpress" to wp. The name could remain wordpress but I choose to have it as wp.
2. Setup Wordpress MySql Database. You can use mysql command line for this. The basic idea here is to create wp database, wp user which has all the access to the DB, as show below

WordPress DB

```
$ mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 5340 to server version: 3.23.54

Type 'help;' or '\h' for help. Type '\c' to clear the buffer.

mysql> CREATE DATABASE databasename;
Query OK, 1 row affected (0.00 sec)

mysql> GRANT ALL PRIVILEGES ON databasename.* TO "wp"@"localhost"
    -> IDENTIFIED BY "wp";
Query OK, 0 rows affected (0.00 sec)

mysql> FLUSH PRIVILEGES;
Query OK, 0 rows affected (0.01 sec)
mysql> EXIT
Bye
$
```

3. Setup wp-config.php. Edit this file found withing /var/www/html/wp directory to setup
    a. DB_NAME
    b. DB_USER
    c. DB_PASSWORD
    d. DB_HOST
4. Run WordPress install script. At this point the setup script will setup your database objects and configurations required for the application.
5. --

Install/setup simplesamlphp

1. Download SimpleSAMLPhp from https://simplesamlphp.org/download. The version used here is 1.13.0
2. Install it per, https://simplesamlphp.org/docs/stable/simplesamlphp-install. During the installation simplesamlphp was placed in non-default directory on /opt/simplesamlphp so for this **13 Installing simpleSAMLphp in alternative locations** was followed. During this process the content of www directory was moved from /opt/simplesamplphp to /var/www/html/simplesaml. The directory structure is shown as below;

```
[root@centoswp html]# cd simplesaml/
[root@centoswp simplesaml]# ls -al
total 64
drwxr-xr-x. 9 root root 4096 Oct 13 23:14 .
drwxr-xr-x. 4 root root 4096 Oct 16 23:04 ..
drwxr-xr-x. 2 root root 4096 Sep 25 08:25 admin
drwxr-xr-x. 2 root root 4096 Sep 25 08:25 auth
-rw-r--r--. 1 root root 3444 Sep 25 08:25 authmemcookie.php
-rw-r--r--. 1 root root 3008 Sep 25 08:25 errorreport.php
drwxr-xr-x. 2 root root 4096 Sep 25 08:25 example-simple
-rw-r--r--. 1 root root 3020 Oct 13 23:14 _include.php
-rw-r--r--. 1 root root  144 Sep 25 08:25 index.php
-rw-r--r--. 1 root root  529 Sep 25 08:25 logout.php
-rw-r--r--. 1 root root 5374 Sep 25 08:25 module.php
drwxr-xr-x. 5 root root 4096 Sep 25 08:25 resources
drwxr-xr-x. 4 root root 4096 Sep 25 08:25 saml2
drwxr-xr-x. 4 root root 4096 Sep 25 08:25 shib13
drwxr-xr-x. 3 root root 4096 Sep 25 08:25 wsfed
```

   Some of the configuration with respect to this are;
    a. On apache conf, the alias created is ; **Alias /simplesaml /var/www/html/simplesaml**
    b. On _include.php within /var/www/html/simplesaml directory, require_once was passd with '/opt/simplesamlphp/lib/_autoload.php' variable and $configdir points to '/opt/simplesamlphp/config'. Hence;
    require_once('/opt/simplesamlphp/lib/_autoload.php');
    $configdir = '/opt/simplesamlphp/config';
3. Setup SimpleSAMLPhp as a Service Provider (SP).

a. Follow the instruction as per, https://simplesamlphp.org/docs/stable/simplesamlphp-sp. The Documentation is pretty good, at least the version that was used.
b. Since IDP in use was not Feide OPenId and is Shibboleth IDP, https://<ipaddress or servername>/simplesaml/**admin/metadata-converter.php** was used to create PHP version of the IDP
c. The PHP version of the IDP was then placed on **metadata/saml20-idp-remote.php**
d. SP Metadata was obtained using **https://<ipaddress>/simplesaml/module.php/saml/sp/metadata.php/default-sp** and was provided to IDP
e. Ask IDP admin to add the provided SP Metadata and ask them to add identified per user (like UID, email....)

4. Test SP / IDP integration
   a. Go to https://<ipaddress/servername>/simplesaml/module.php/core/frontpage_auth.php and click the "Test configured authentication sources" link.
   b. Choose default-sp
   c. This should bring IDP login page
   d. Once login it should present with a simplesamlphp page which displays attributes and their values from the IDP connected
5. If you see (as shown below) idp released attributes/values that means your SimpleSAMLPhp is working fine

SAML 2.0 SP Demo Example

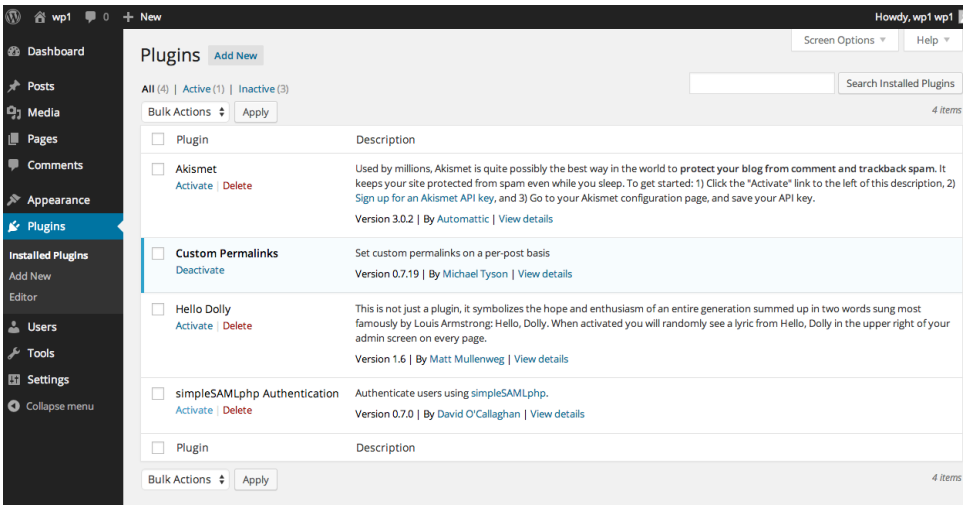
Hi, this is the status page of SimpleSAMLphp. Here you can see if your session is timed out, how long it lasts until it times out and all the attributes that are attached to your session.

Your attributes

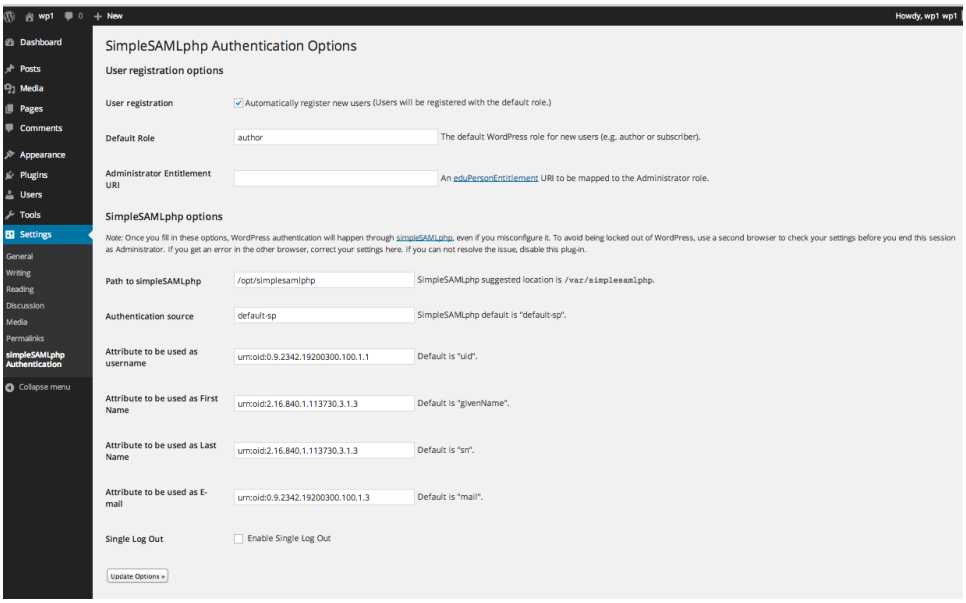
urn:oid:2.5.4.4	Rayamajhi
urn:oid:2.5.4.42	Susil
urn:oid:0.9.2342.19200300.100.1.1	SF234877
urn:oid:2.16.840.1.113730.3.1.3	022348775
urn:oid:0.9.2342.19200300.100.1.3	susil.rayamajhi@ucsf.edu

Install/setup WordPress simplesamlphp plugin

1. This section is relevant if you are trying to protect all of WP content
2. Install simplesamlphp-authentication plugin from https://wordpress.org/plugins/simplesamlphp-authentication/ . The version used here is 0.7.0
3. Unzip and place the plugin wihtin **/var/www/html/wp/wp-content/plugins** directory
4. Activate the plugin from WordPress from the page show below;



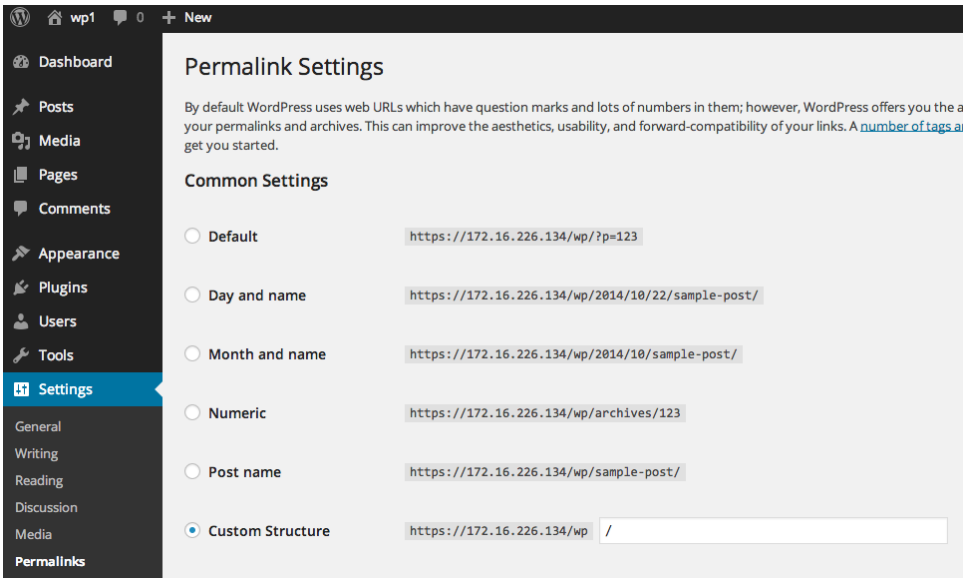
5. Go to Plugin Option Page, shown below;



6. Provide Setup Information and save the setting. In the above setting ,
    a. User Registration is Checked, which inserts user record into WordPress Tables
    b. The default role assigned with user is inserted in "author"
    c. The Attribute names are urn name for uid/givenname/sn/mail
    d. Default sp is "default-sp". This is what is being conf'd to point to IDP for SSO
    e. Path to simpleSAMLPhp is /opt/simplesamlphp since this is where the software is installed/extracted.
7. At this point all your wordpress login required pages will be SSO'd

Provide hybrid protection mechanism for certain pages with WordPress

1. Use this page if you need to protect just certain page within WordPress
2. Setup WordPress such that its' permlink is in the format of https:/servername/blogname**/pagename,** (as show below)



3. Once setup put a interseption to check against the page and invoke SSO login page using simplesamlphp protection code

```
 if (strstr($pageName,"q2")) {
require_once('/opt/simplesamlphp/lib/_autoload.php');
$as = new SimpleSAML_Auth_Simple('default-sp');
$as->requireAuth();
```

```
$attributes = $as->getAttributes();
//print_r($attributes);
}
```

}

Here the url https://servername/wp/q2 , with the page q2 is protected

4. Test the page by launching the url above in a new browser session. This should invoke SSO page for authentication.

Alternate: The other option will be to write a simple WordPress Plugin to do just as mentioned above where you can add a hidden field on each protected filed then check that field value for every call and if it is protected then invoke the simplesamlphp code to take user to SSO page.

For simplicity this test setup's services and files/directories are owned by root.

Enjoy!
Susil