SP Hands-on Session

Installing and Configuring a Shibboleth 2 Service Provider

SWITCH

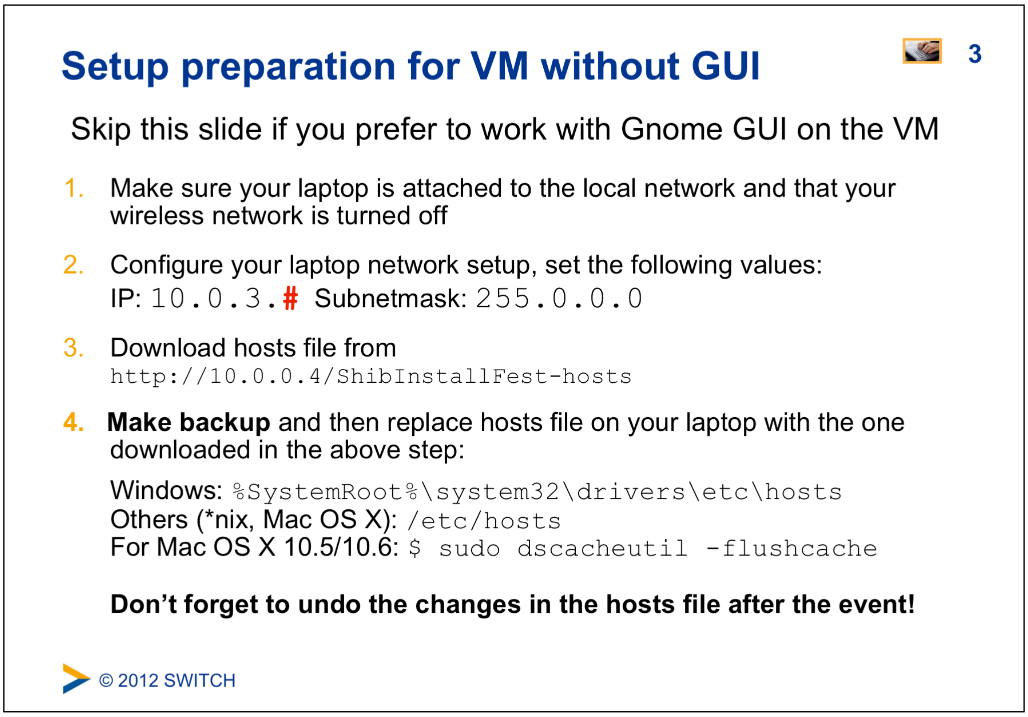
Serving Swiss Universities

Notes:

Credits and General Information

- Slides were originally created by Scott Cantor, Internet 2 Developer of the Shibboleth Service Provider
- Focus lies on a general overview
- Course material will be published online
- If you see this ⌨ on a slide, hands-on work is required
- URLs at bottom right point to pages with more details
- On slides with ⌘ Detour separate presentations focus on special topic

Notes:



Setup preparation for VM without GUI

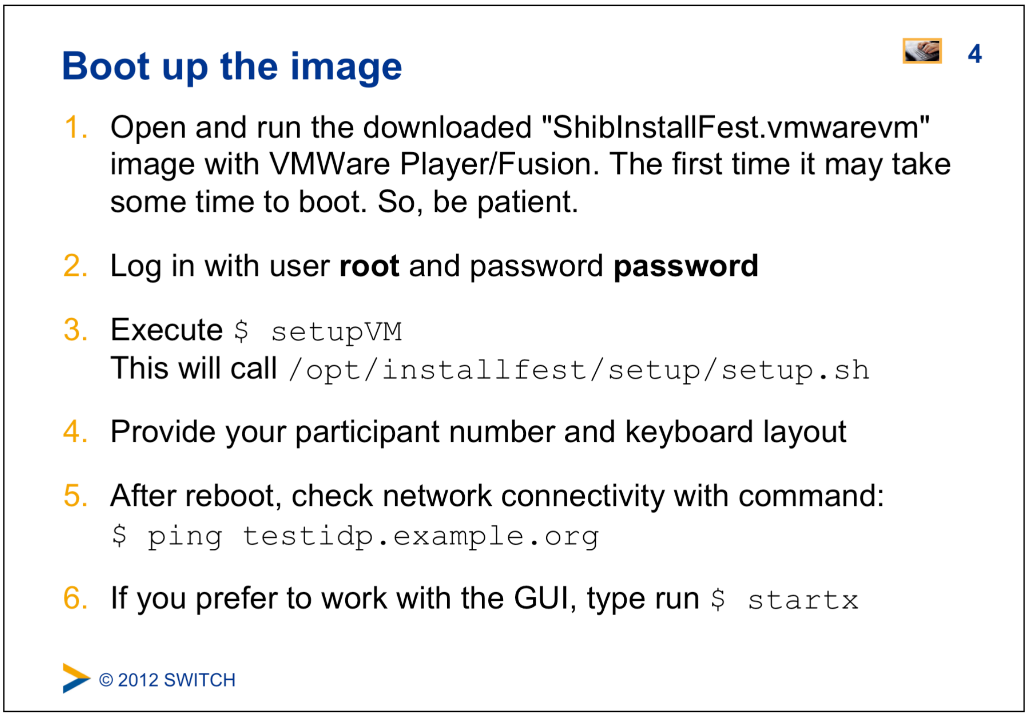
Skip this slide if you prefer to work with Gnome GUI on the VM

1. Make sure your laptop is attached to the local network and that your wireless network is turned off
2. Configure your laptop network setup, set the following values:
   IP: `10.0.3.#` Subnetmask: `255.0.0.0`
3. Download hosts file from
   `http://10.0.0.4/ShibInstallFest-hosts`
4. **Make backup** and then replace hosts file on your laptop with the one downloaded in the above step:

   Windows: `%SystemRoot%\system32\drivers\etc\hosts`
   Others (*nix, Mac OS X): `/etc/hosts`
   For Mac OS X 10.5/10.6: `$ sudo dscacheutil -flushcache`

   Don't forget to undo the changes in the hosts file after the event!

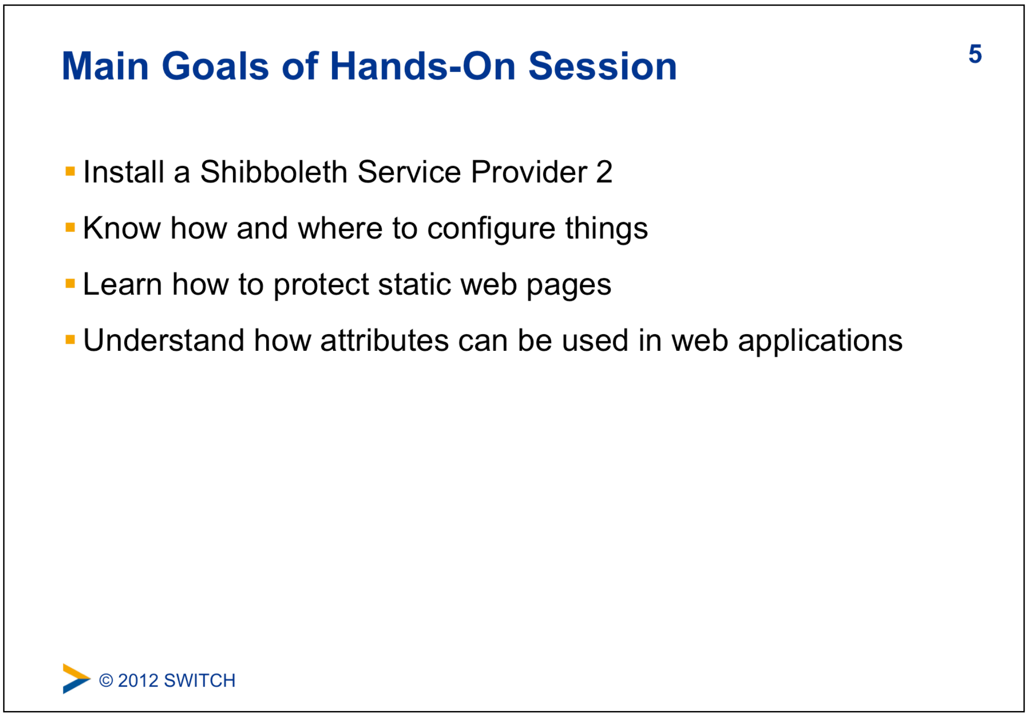
Notes:



Boot up the image

1. Open and run the downloaded "ShibInstallFest.vmwarevm" image with VMWare Player/Fusion. The first time it may take some time to boot. So, be patient.
2. Log in with user **root** and password **password**
3. Execute `$ setupVM`
   This will call `/opt/installfest/setup/setup.sh`
4. Provide your participant number and keyboard layout
5. After reboot, check network connectivity with command:
   `$ ping testidp.example.org`
6. If you prefer to work with the GUI, type run `$ startx`

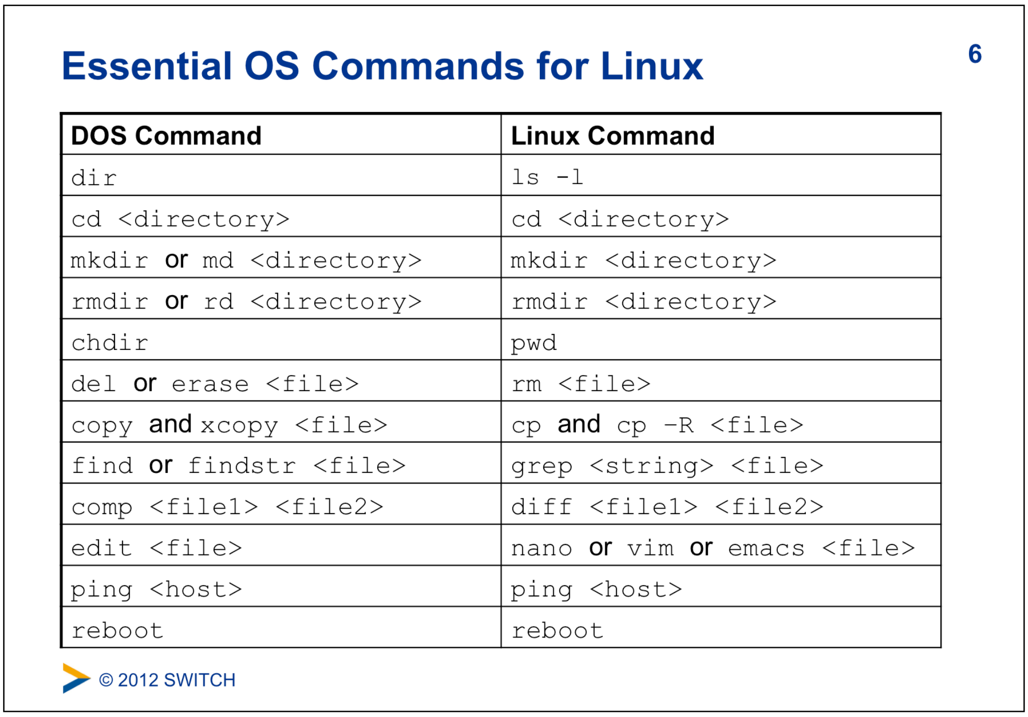
Notes:



Main Goals of Hands-On Session

- Install a Shibboleth Service Provider 2
- Know how and where to configure things
- Learn how to protect static web pages
- Understand how attributes can be used in web applications

Notes:



Essential OS Commands for Linux

DOS Command	Linux Command
`dir`	`ls -l`
`cd <directory>`	`cd <directory>`
`mkdir` or `md <directory>`	`mkdir <directory>`
`rmdir` or `rd <directory>`	`rmdir <directory>`
`chdir`	`pwd`
`del` or `erase <file>`	`rm <file>`
`copy` and `xcopy <file>`	`cp` and `cp -R <file>`
`find` or `findstr <file>`	`grep <string> <file>`
`comp <file1> <file2>`	`diff <file1> <file2>`
`edit <file>`	`nano` or `vim` or `emacs <file>`
`ping <host>`	`ping <host>`
`reboot`	`reboot`

Notes:

File Editing Commands for Terminal Editor

Editor	nano	vim	emacs
Open file	`$ nano <file>`	`$ vim <file>`	`$ emacs <file>`
Save file	<ctrl>-o	<esc>, :w	<ctrl>-x, <ctrl>-s
Save and exit	<ctrl>-x	<esc>, :wq	<ctrl>-x, <ctrl>-c, y
Search string	<ctrl>-w, **string**	<esc>, /**string**	<ctrl>-s, **string**
Go to line number	<ctrl>--, **number**	<esc>, **number**, <shift>-G	<esc>, **number**, <shift>-G

“nano” is recommended for Linux beginners without GUI
Alternative for GUI users: Gnome “Text Editor” on desktop

Notes:

Tips and Tricks for Hands-On Session

- Don’t enable the wireless network during the workshop!
  This could break your connectivity with other workshop hosts!
- Lines starting with `$` are commands to be executed
- Character `\` is line break symbol,
  which allows to break a line when typed
- Watch out for invalid XML/configuration errors

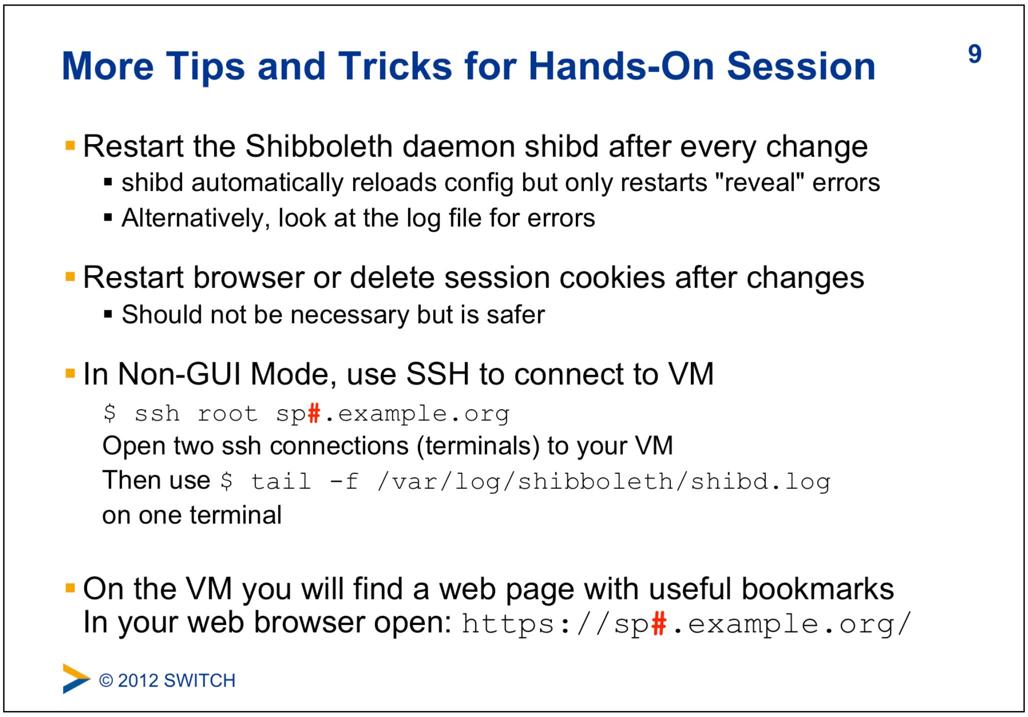
```
$ shibd -tc /etc/shibboleth/shibboleth2.xml
```

  - Reports errors regarding well-formedness and schema validity

```
$ xmlwf /path/some-XML-File.xml
```

  - Reports errors and line/column number if XML is not well-formed
  - E.g. `shibboleth2.xml:`**`261:2`**`: mismatched tag`

Notes:



More Tips and Tricks for Hands-On Session

- Restart the Shibboleth daemon shibd after every change
  - shibd automatically reloads config but only restarts "reveal" errors
  - Alternatively, look at the log file for errors
- Restart browser or delete session cookies after changes
  - Should not be necessary but is safer
- In Non-GUI Mode, use SSH to connect to VM

  ```
  $ ssh root sp#.example.org
  ```

  Open two ssh connections (terminals) to your VM

  Then use `$ tail -f /var/log/shibboleth/shibd.log`

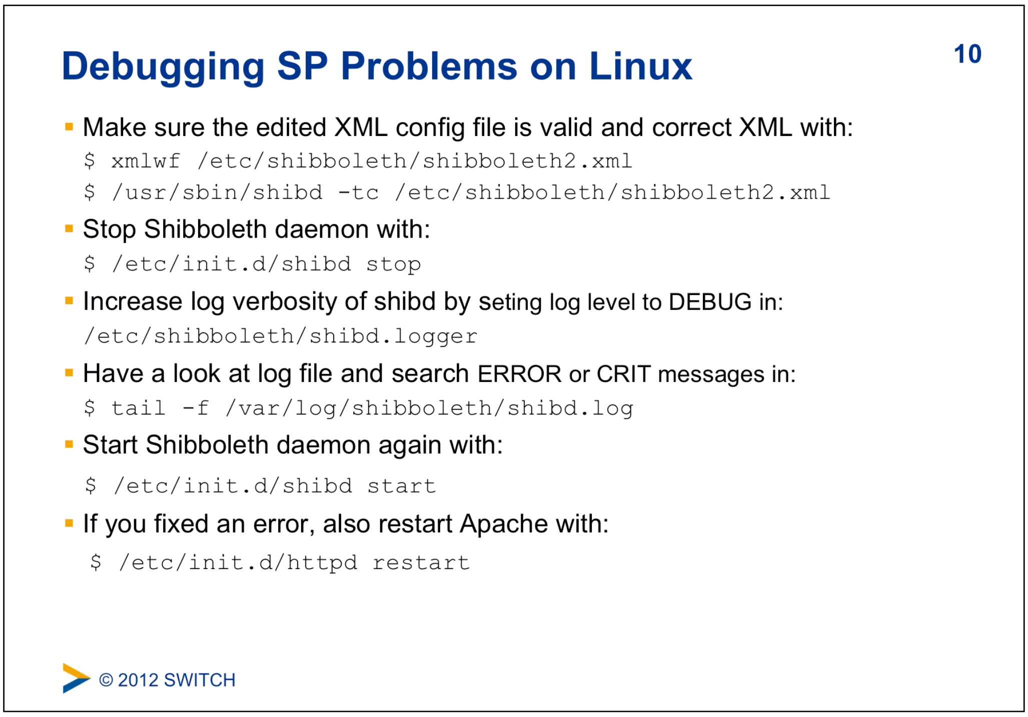
  on one terminal
- On the VM you will find a web page with useful bookmarks
  In your web browser open: `https://sp#.example.org/`

Notes: ______________________________

____

____

____



Debugging SP Problems on Linux

- Make sure the edited XML config file is valid and correct XML with:

  ```
  $ xmlwf /etc/shibboleth/shibboleth2.xml
  $ /usr/sbin/shibd -tc /etc/shibboleth/shibboleth2.xml
  ```
- Stop Shibboleth daemon with:

  ```
  $ /etc/init.d/shibd stop
  ```
- Increase log verbosity of shibd by seting log level to DEBUG in:

  `/etc/shibboleth/shibd.logger`
- Have a look at log file and search ERROR or CRIT messages in:

  ```
  $ tail -f /var/log/shibboleth/shibd.log
  ```
- Start Shibboleth daemon again with:

  ```
  $ /etc/init.d/shibd start
  ```
- If you fixed an error, also restart Apache with:

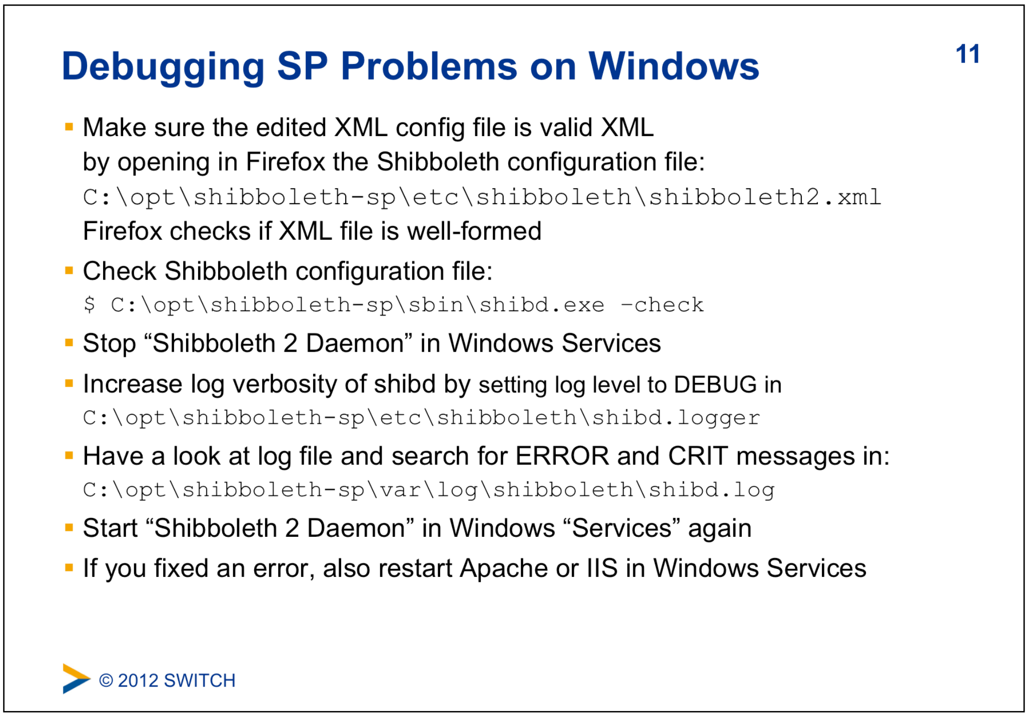
  ```
  $ /etc/init.d/httpd restart
  ```

Notes: ______________________________

____

____

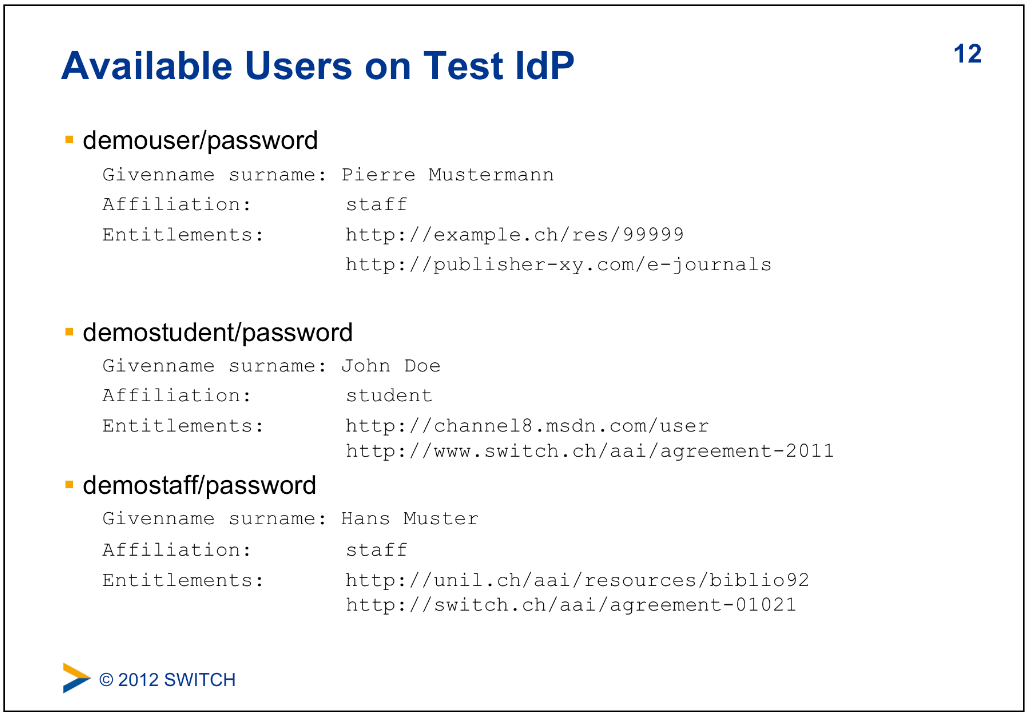
____



Debugging SP Problems on Windows

- Make sure the edited XML config file is valid XML by opening in Firefox the Shibboleth configuration file:
  ```
  C:\opt\shibboleth-sp\etc\shibboleth\shibboleth2.xml
  ```
  Firefox checks if XML file is well-formed
- Check Shibboleth configuration file:
  ```
  $ C:\opt\shibboleth-sp\sbin\shibd.exe –check
  ```
- Stop “Shibboleth 2 Daemon” in Windows Services
- Increase log verbosity of shibd by setting log level to DEBUG in
  ```
  C:\opt\shibboleth-sp\etc\shibboleth\shibd.logger
  ```
- Have a look at log file and search for ERROR and CRIT messages in:
  ```
  C:\opt\shibboleth-sp\var\log\shibboleth\shibd.log
  ```
- Start “Shibboleth 2 Daemon” in Windows “Services” again
- If you fixed an error, also restart Apache or IIS in Windows Services

Notes:



Available Users on Test IdP

- demouser/password
  ```
  Givenname surname: Pierre Mustermann
  Affiliation:        staff
  Entitlements:       http://example.ch/res/99999
                      http://publisher-xy.com/e-journals
  ```
- demostudent/password
  ```
  Givenname surname: John Doe
  Affiliation:        student
  Entitlements:       http://channel8.msdn.com/user
                      http://www.switch.ch/aai/agreement-2011
  ```
- demostaff/password
  ```
  Givenname surname: Hans Muster
  Affiliation:        staff
  Entitlements:       http://unil.ch/aai/resources/biblio92
                      http://switch.ch/aai/agreement-01021
  ```

Notes:

SP Overview and Installation

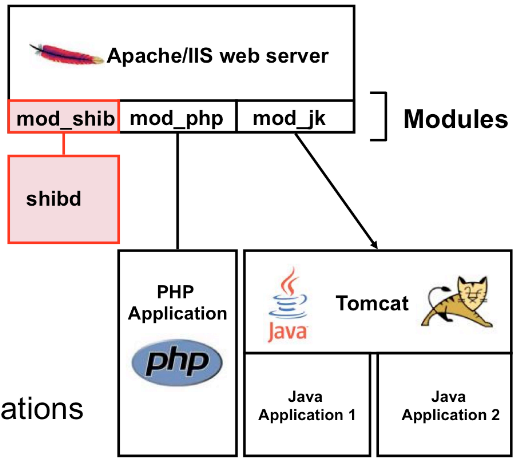
Goals:

1. Terminology and SP Overview
2. Installation and Directory Structure
3. Generating Key and Certificate
4. Quick Sanity Check
5. Picking an entityID

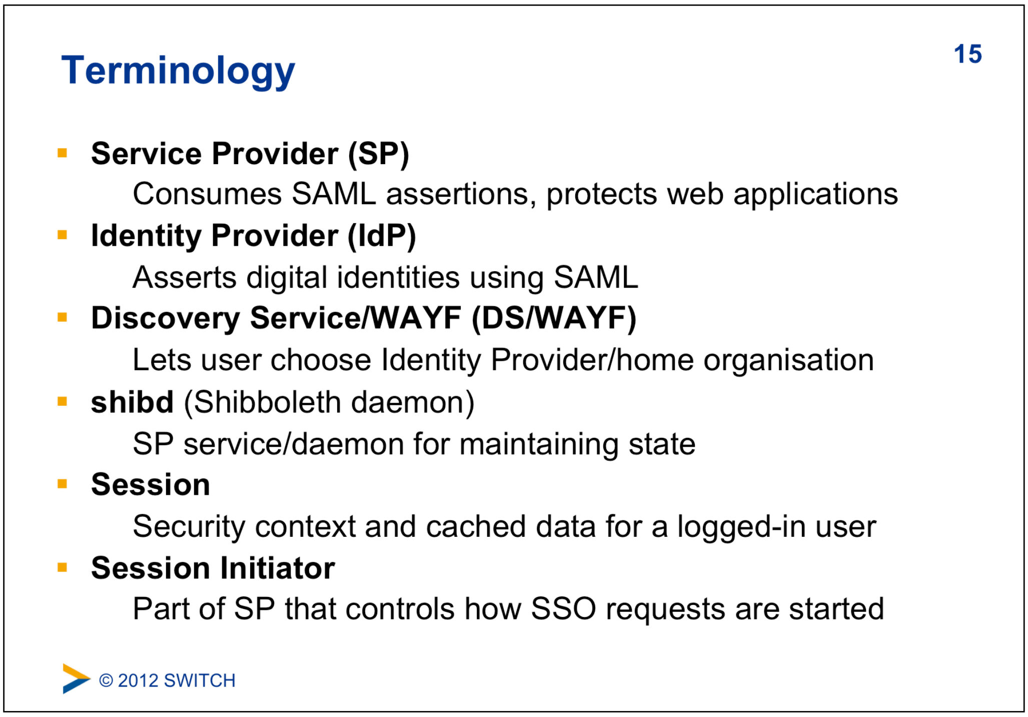
Notes:

SP: Daemon & mod_shib

- Runs on: Linux, Solaris, Windows, Mac OS X, FreeBSD, …
- Protects web applications
- shibd processes attributes
- Can authorize users with
  - Apache directives
  - Shibboleth XML Access rules
- Provides attributes to applications



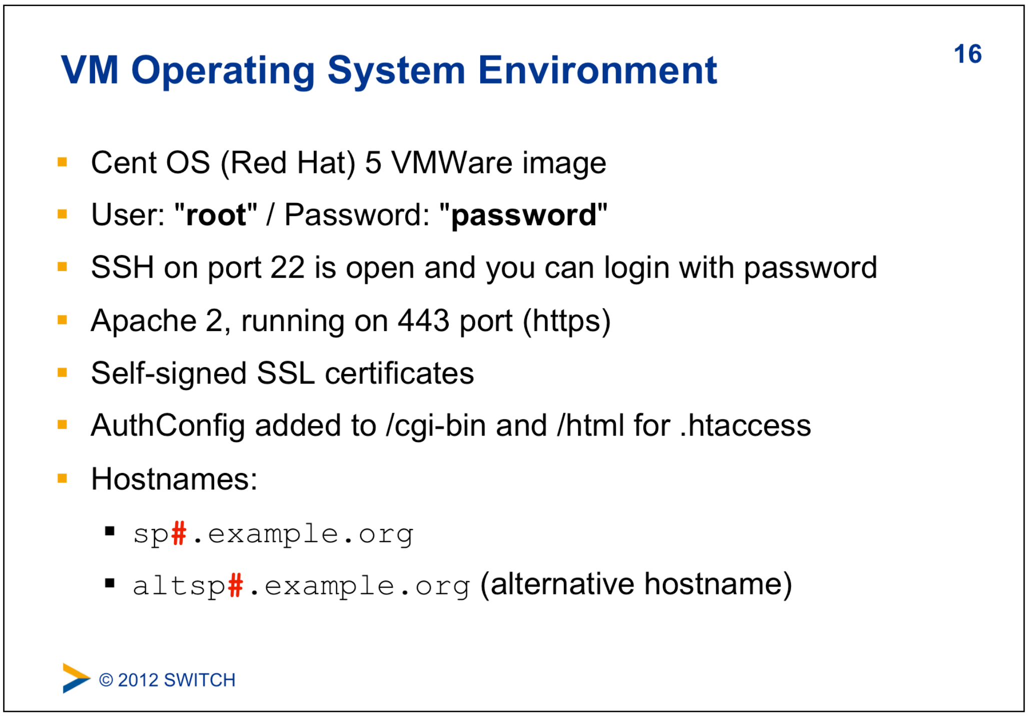
Notes:



Terminology

- **Service Provider (SP)**
  Consumes SAML assertions, protects web applications
- **Identity Provider (IdP)**
  Asserts digital identities using SAML
- **Discovery Service/WAYF (DS/WAYF)**
  Lets user choose Identity Provider/home organisation
- **shibd** (Shibboleth daemon)
  SP service/daemon for maintaining state
- **Session**
  Security context and cached data for a logged-in user
- **Session Initiator**
  Part of SP that controls how SSO requests are started

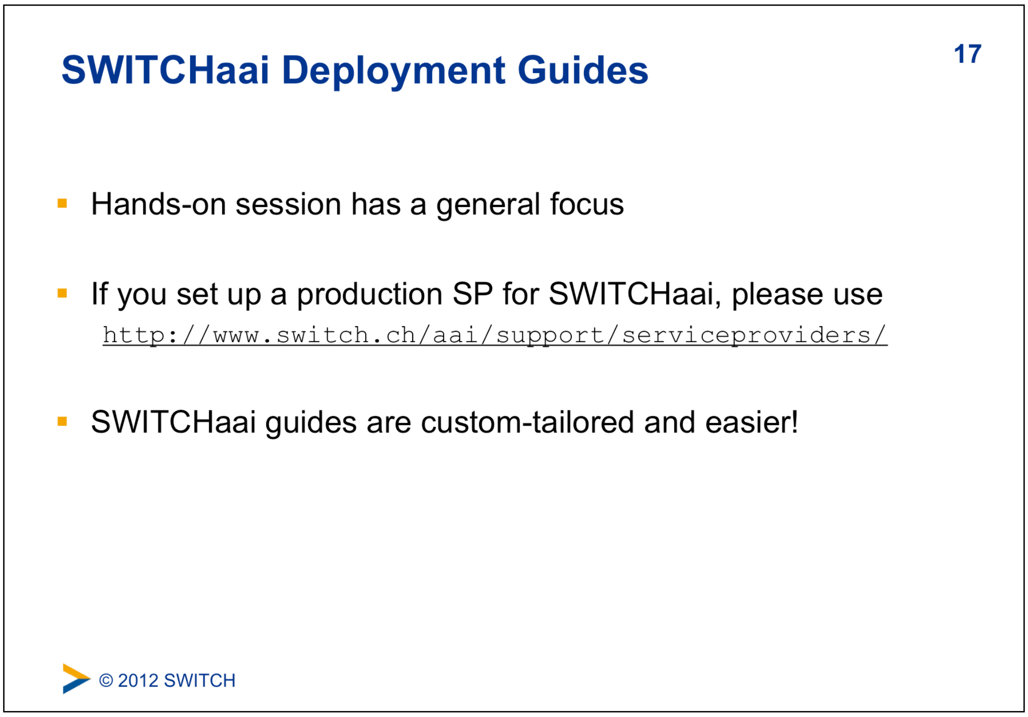
Notes:



VM Operating System Environment

- Cent OS (Red Hat) 5 VMWare image
- User: "**root**" / Password: "**password**"
- SSH on port 22 is open and you can login with password
- Apache 2, running on 443 port (https)
- Self-signed SSL certificates
- AuthConfig added to /cgi-bin and /html for .htaccess
- Hostnames:
  - `sp#.example.org`
  - `altsp#.example.org` (alternative hostname)

Notes:

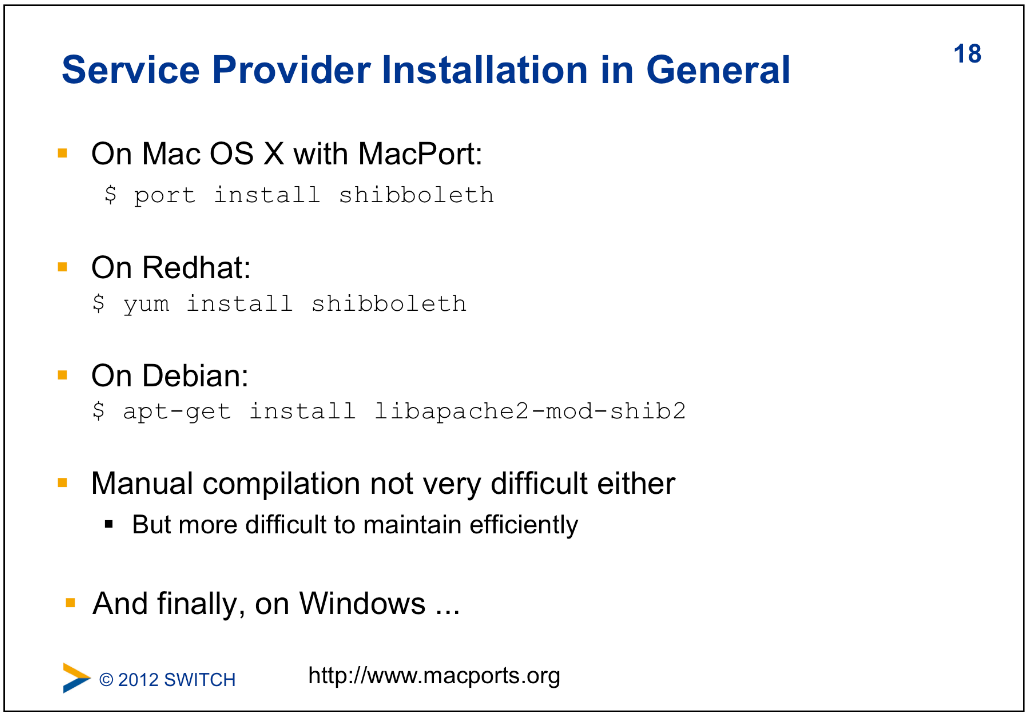


SWITCHaai Deployment Guides

17

- Hands-on session has a general focus
- If you set up a production SP for SWITCHaai, please use
  `http://www.switch.ch/aai/support/serviceproviders/`
- SWITCHaai guides are custom-tailored and easier!

Notes:



Service Provider Installation in General

18

- On Mac OS X with MacPort:
  ```
  $ port install shibboleth
  ```
- On Redhat:
  ```
  $ yum install shibboleth
  ```
- On Debian:
  ```
  $ apt-get install libapache2-mod-shib2
  ```
- Manual compilation not very difficult either
  - But more difficult to maintain efficiently
- And finally, on Windows ...

http://www.macports.org

Notes:

Service Provider Installation on Windows

- Windows installation requires more clicks but still is easy
- Shibboleth is generally installed in C:\opt\shibboleth-sp
  - Path to binary: C:\opt\shibboleth-sp\sbin\shibd.exe
- Directory structures within shibboleth-sp is like in Unix/Linux
  - etc\shibboleth\
  - var\log\shibboleth\
  - bin\
  - sbin\
  - sbin\

Notes:

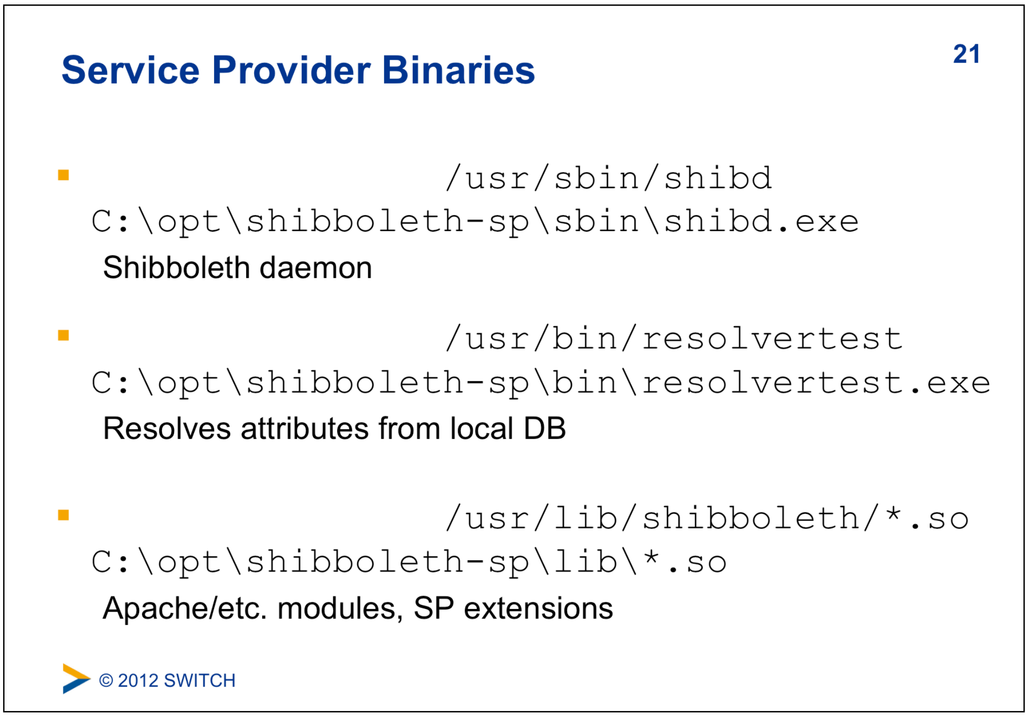
Service Provider Installation on VM Image

- Installation on your VM
- RPM-based:

```
$ rpm -ivh /opt/installfest/distro/RPMS/*.rpm
```

- Special files copied during shibboleth installation:
  - `apache22.conf` copied to `/etc/httpd/conf.d/shib.conf`
  - `shibd` init script copied to `/etc/init.d/shibd`

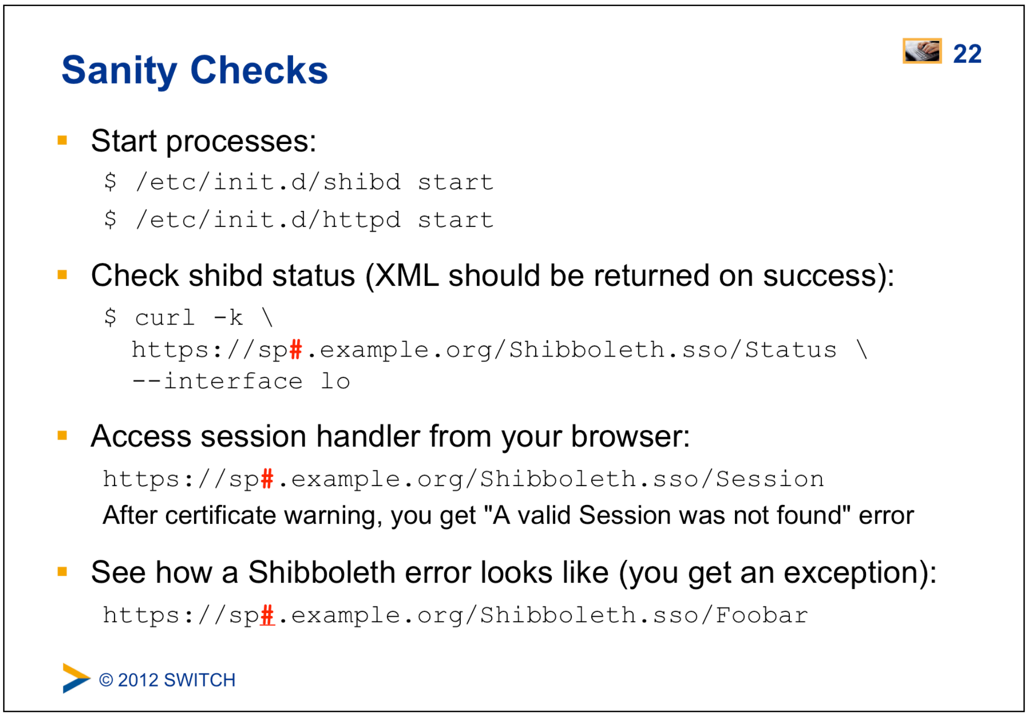
Notes:



Service Provider Binaries

- `/usr/sbin/shibd`
  `C:\opt\shibboleth-sp\sbin\shibd.exe`
  Shibboleth daemon
- `/usr/bin/resolvertest`
  `C:\opt\shibboleth-sp\bin\resolvertest.exe`
  Resolves attributes from local DB
- `/usr/lib/shibboleth/*.so`
  `C:\opt\shibboleth-sp\lib\*.so`
  Apache/etc. modules, SP extensions

Notes: ____________________



Sanity Checks

- Start processes:
  ```
  $ /etc/init.d/shibd start
  $ /etc/init.d/httpd start
  ```
- Check shibd status (XML should be returned on success):
  ```
  $ curl -k \
    https://sp#.example.org/Shibboleth.sso/Status \
    --interface lo
  ```
- Access session handler from your browser:
  `https://sp#.example.org/Shibboleth.sso/Session`
  After certificate warning, you get "A valid Session was not found" error
- See how a Shibboleth error looks like (you get an exception):
  `https://sp#.example.org/Shibboleth.sso/Foobar`

Notes: ____________________

Important directories

23

- `/etc/shibboleth/`
  - Master and supporting configuration files
  - Locally maintained metadata files
  - HTML templates (customize them to adapt look&feel to your application)
  - Logging configuration files (*.logger)
  - Credentials (certificates and private keys)
- `/var/run/shibboleth/`
  - UNIX socket
  - remote metadata backups
- `/var/log/shibboleth/`
  - shibd.log and transaction.log files
- `/var/log/httpd/`
  - `native.log` (is written by mod_shib web server module)

Notes:

Key/Certificate Generation

24

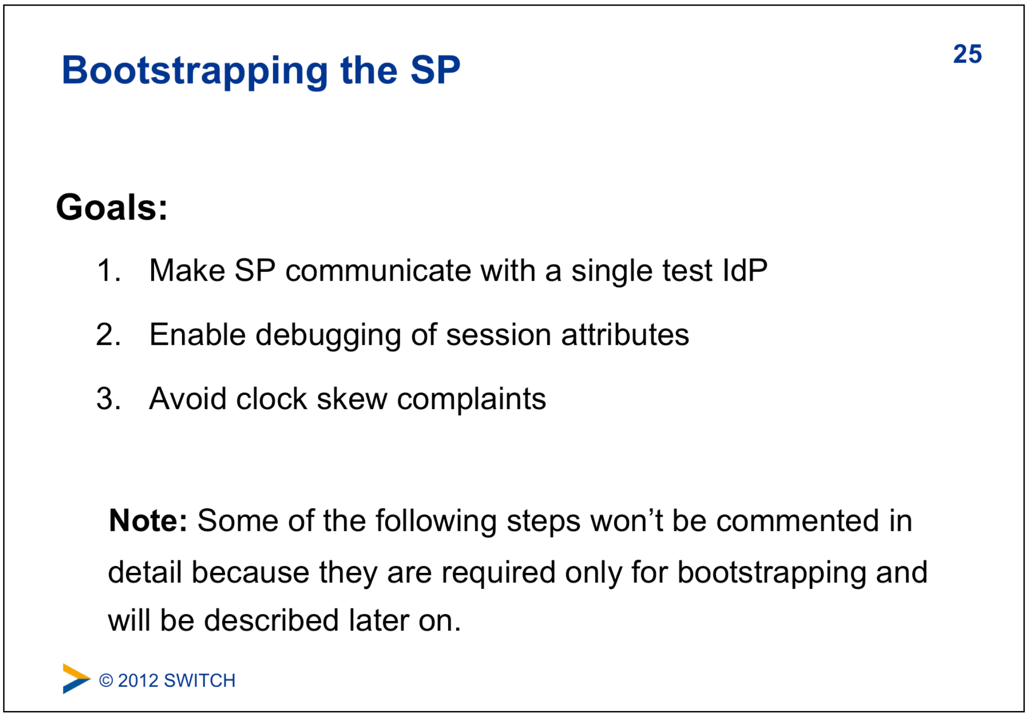
- Script to generate certificate and private key:
  `/etc/shibboleth/keygen.sh`
- Runs automatically during installation
- For this workshop, copy over a pre-generated set for your SP:

```
$ cp /opt/installfest/sps/sp#/sp.key \
  /etc/shibboleth/sp-key.pem

$ cp /opt/installfest/sps/sp#/sp.crt \
  /etc/shibboleth/sp-cert.pem
```

Answer ‘yes’ to overwrite existing files

Notes:



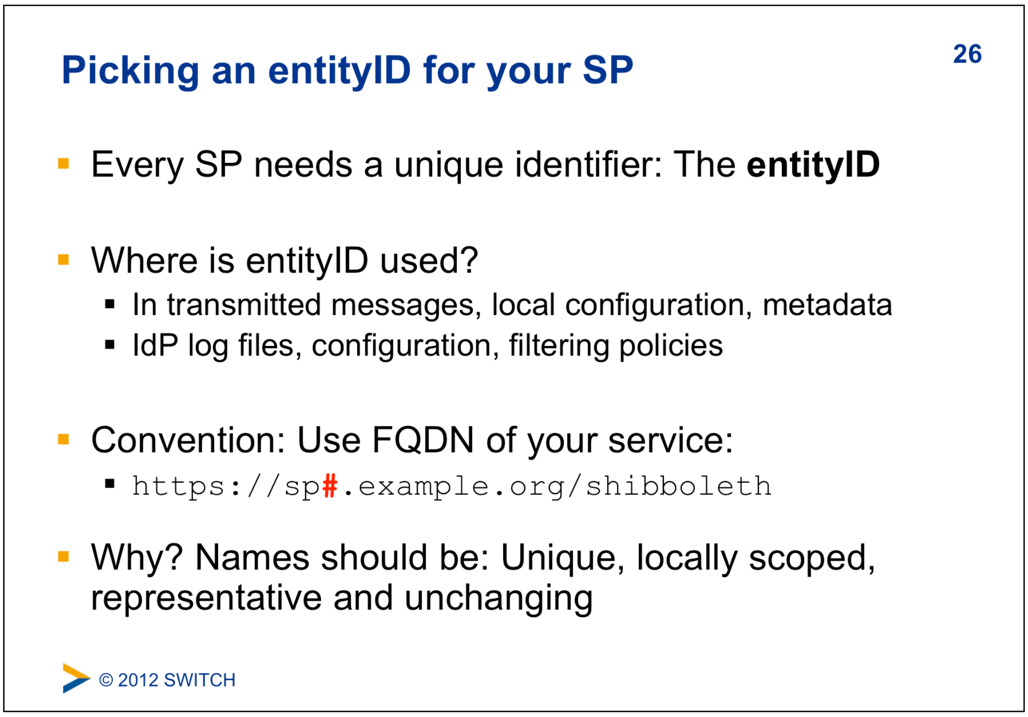
Bootstrapping the SP

Goals:

1. Make SP communicate with a single test IdP
2. Enable debugging of session attributes
3. Avoid clock skew complaints

Note: Some of the following steps won’t be commented in detail because they are required only for bootstrapping and will be described later on.

Notes:



Picking an entityID for your SP

- Every SP needs a unique identifier: The **entityID**
- Where is entityID used?
  - In transmitted messages, local configuration, metadata
  - IdP log files, configuration, filtering policies
- Convention: Use FQDN of your service:
  - `https://sp#.example.org/shibboleth`
- Why? Names should be: Unique, locally scoped, representative and unchanging

Notes:

Bootstrapping the SP Chapter I

27

- Relax some requirements, set your entityID and default IdP entityID

```
$ vim /etc/shibboleth/shibboleth2.xml

Line 6: (Do NOT do this for a production service)
clockSkew="1800000">

Line 23:
<ApplicationDefaults \
    entityID="https://sp#.example.org/shibboleth"
    homeURL="https://sp#.example.org/secure/"

Line 44:
<SSO entityID="https://testidp.example.org/idp/shibboleth"

Line 59:
<Handler type="Session" Location="/Session"
    showAttributeValues="true"/>
```

Notes: ______________________________

____

____

____

Bootstrapping the SP Chapter II

28

- Get the testidp metadata remotely:

```
$ vim /etc/shibboleth/shibboleth2.xml
```

Line 75: (Do NOT do disable Signature filter for a production service)

Uncomment whole `<MetadataProvider>` element!

```
<MetadataProvider type="XML" uri="https://testidp.example.org/testidp-
    metadata.xml" backingFilePath="/etc/shibboleth/testidp-
    metadata.xml" reloadInterval="7200">
<!-- <MetadataFilter type="RequireValidUntil" …  /> -->
<!-- <MetadataFilter type="Signature" …  /> -->
</MetadataProvider>
```

- Normally: Provide your SP's metadata to federation/IdPs
  But in this workshop, this was already done for you.
  - Metadata self-generated by your Service Provider
  - `https://sp#.example.org/Shibboleth.sso/Metadata`

Notes: ______________________________

____

____

____

Quick Test

- Make sure configuration works (should return "…is loadable"):

```
$ shibd -tc /etc/shibboleth/shibboleth2.xml
```

  Service Provider reloads shibboleth2.xml automatically when it changed

- Try it with a browser:

  `https://sp#.example.org/secure/`

  `/secure/` is protecty by Shibboleth "by default". See bottom of file `/etc/httpd/conf.d/shib.conf`
  Therefore, you should be forced to authenticate. Login at Test IdP with demouser/password and you should get access to this directory.

- Then call the Shibboleth session handler to see the attributes:

  `https://sp#.example.org/Shibboleth.sso/Session`
  You should see various attributes like affiliation, entitlement, eppn, etc.

Notes:

AAI Resource Registry

Purpose of the SWITCHaai Resource Registry and how to use it

Please consult the table of contents to find this presentation in your hand-outs.

Notes:

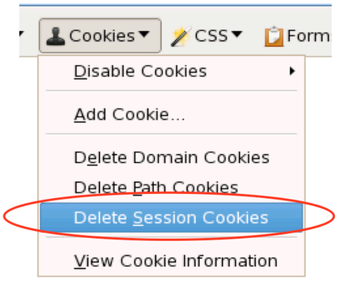
Logging Out

- To logout locally from the SP and kill your session:

```
https://sp#.example.org/Shibboleth.sso/Logout
```

But this won't delete your session on the IdP!

- **Close the browser and restart it again!**



- Or delete all your session cookies
  - Recommendation for testing:
    Use Firefox Web Developer extension

Notes: ____

____

• Alternatively, comment out on about 146 in shibboleth2.xml the SAML2 Logout Initiator

Use a Discovery Service (WAYF)

- Change the default SessionInitiator:

```
$ vim /etc/shibboleth/shibboleth2.xml

Line 44:
<SSO discoveryProtocol="SAMLDS" \
   discoveryURL="https://ds.example.org/DS/WAYF"/>
    SAML2 SAML1
</SSO>
```

Remove the entityID attribute in the <SSO> element in order to use a Discovery Service

Restart Apache and Shibboleth

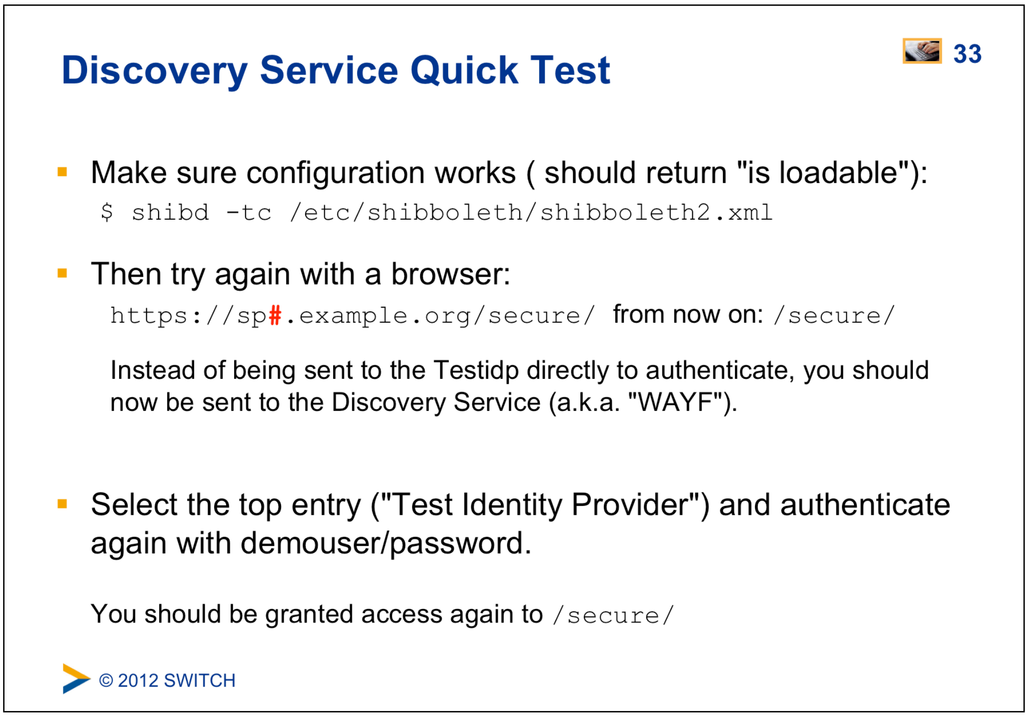
```
$ /etc/init.d/shibd restart
$ /etc/init.d/httpd restart
```

Notes: ____

____

____

____



Discovery Service Quick Test

- Make sure configuration works (should return "is loadable"):

```
$ shibd -tc /etc/shibboleth/shibboleth2.xml
```

- Then try again with a browser:

  `https://sp#.example.org/secure/` from now on: `/secure/`

  Instead of being sent to the Testidp directly to authenticate, you should now be sent to the Discovery Service (a.k.a. "WAYF").

- Select the top entry ("Test Identity Provider") and authenticate again with demouser/password.

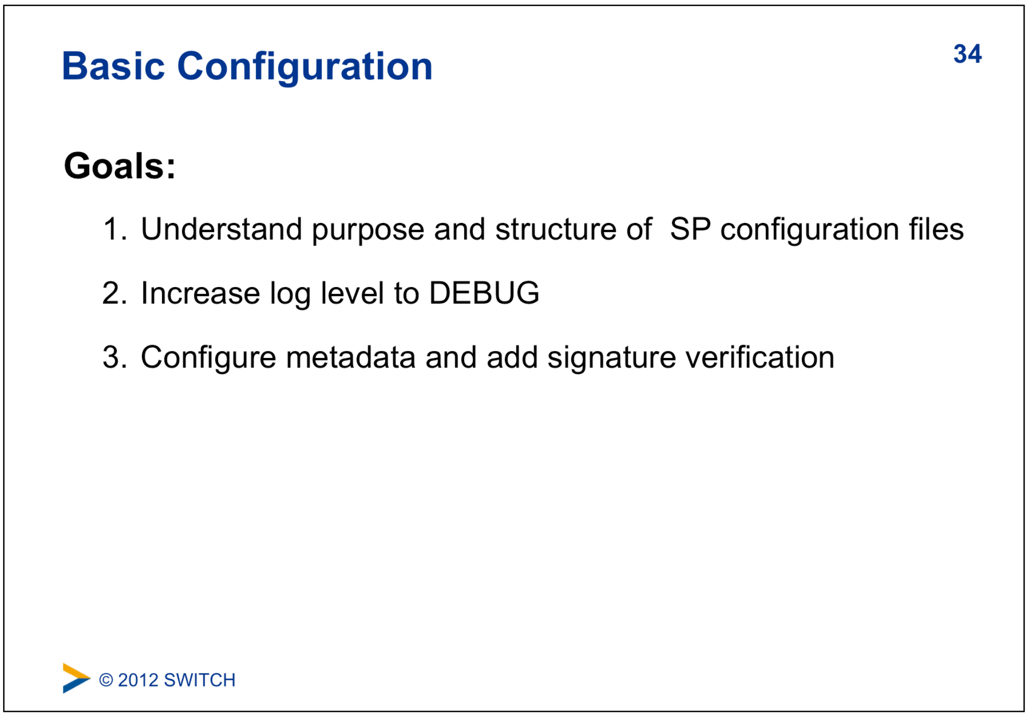
  You should be granted access again to `/secure/`

Notes: ______________________________

____

____

____



Basic Configuration

Goals:

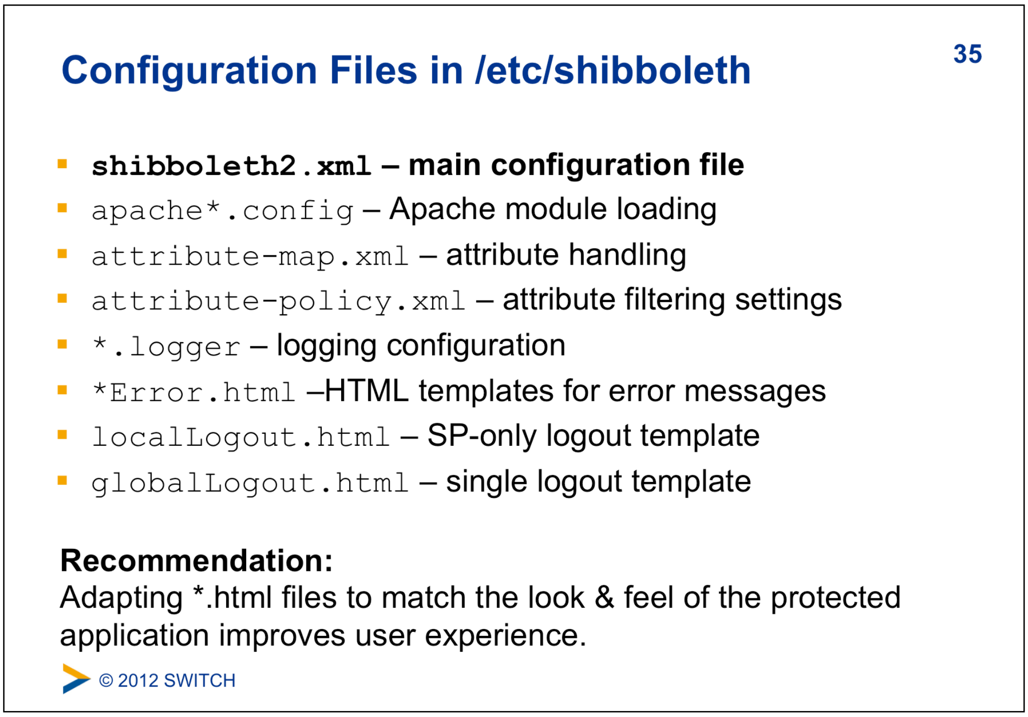
1. Understand purpose and structure of SP configuration files
2. Increase log level to DEBUG
3. Configure metadata and add signature verification

Notes: ______________________________

____

____

____



Configuration Files in /etc/shibboleth

- **`shibboleth2.xml` – main configuration file**
- `apache*.config` – Apache module loading
- `attribute-map.xml` – attribute handling
- `attribute-policy.xml` – attribute filtering settings
- `*.logger` – logging configuration
- `*Error.html` –HTML templates for error messages
- `localLogout.html` – SP-only logout template
- `globalLogout.html` – single logout template

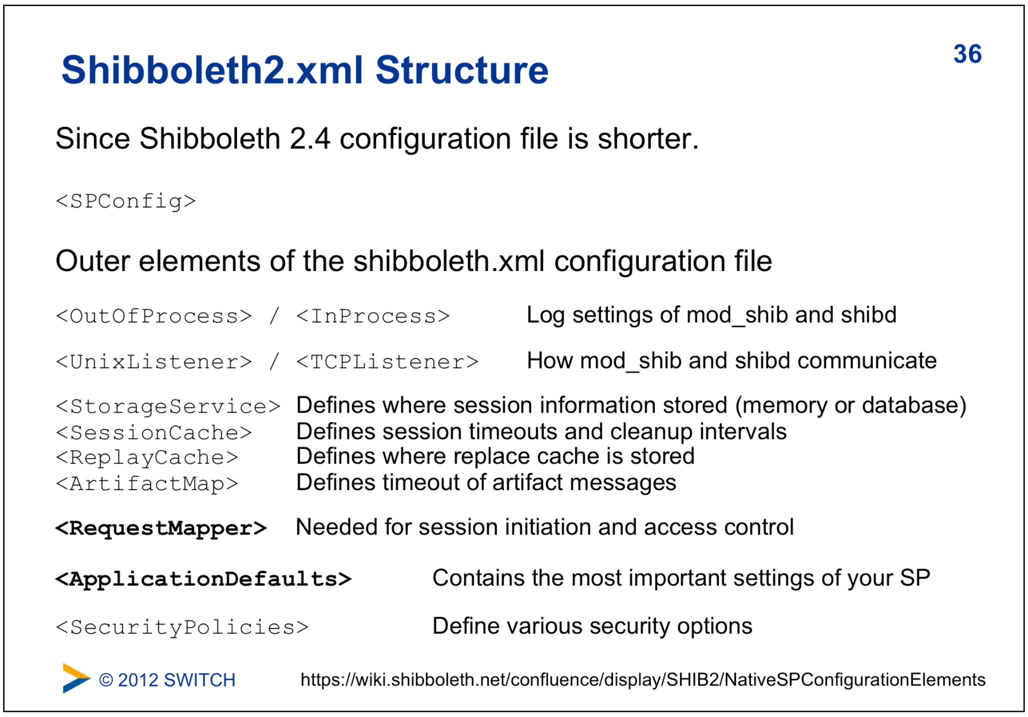
Recommendation:
Adapting *.html files to match the look & feel of the protected application improves user experience.

Notes: ____________________

____

____

____



Shibboleth2.xml Structure

Since Shibboleth 2.4 configuration file is shorter.

`<SPConfig>`

Outer elements of the shibboleth.xml configuration file

`<OutOfProcess> / <InProcess>`	Log settings of mod_shib and shibd
`<UnixListener> / <TCPListener>`	How mod_shib and shibd communicate
`<StorageService>`	Defines where session information stored (memory or database)
`<SessionCache>`	Defines session timeouts and cleanup intervals
`<ReplayCache>`	Defines where replace cache is stored
`<ArtifactMap>`	Defines timeout of artifact messages
`<RequestMapper>`	Needed for session initiation and access control
`<ApplicationDefaults>`	Contains the most important settings of your SP
`<SecurityPolicies>`	Define various security options

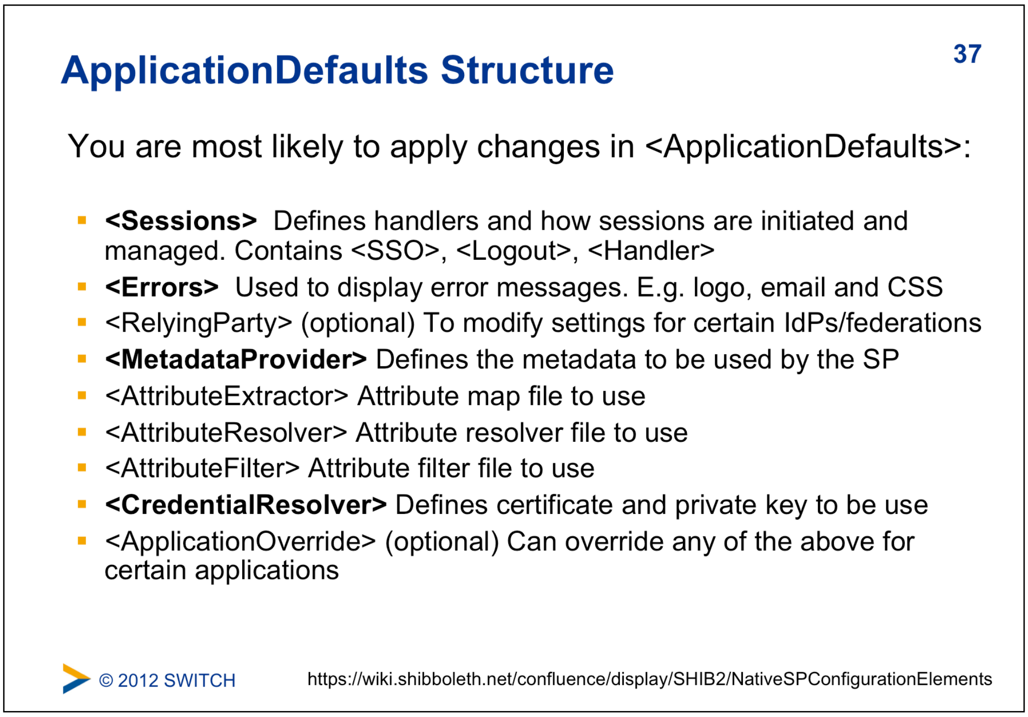
https://wiki.shibboleth.net/confluence/display/SHIB2/NativeSPConfigurationElements

Notes: ____________________

____

____

____



ApplicationDefaults Structure

You are most likely to apply changes in <ApplicationDefaults>:

- **<Sessions>** Defines handlers and how sessions are initiated and managed. Contains <SSO>, <Logout>, <Handler>
- **<Errors>** Used to display error messages. E.g. logo, email and CSS
- <RelyingParty> (optional) To modify settings for certain IdPs/federations
- **<MetadataProvider>** Defines the metadata to be used by the SP
- <AttributeExtractor> Attribute map file to use
- <AttributeResolver> Attribute resolver file to use
- <AttributeFilter> Attribute filter file to use
- **<CredentialResolver>** Defines certificate and private key to be use
- <ApplicationOverride> (optional) Can override any of the above for certain applications

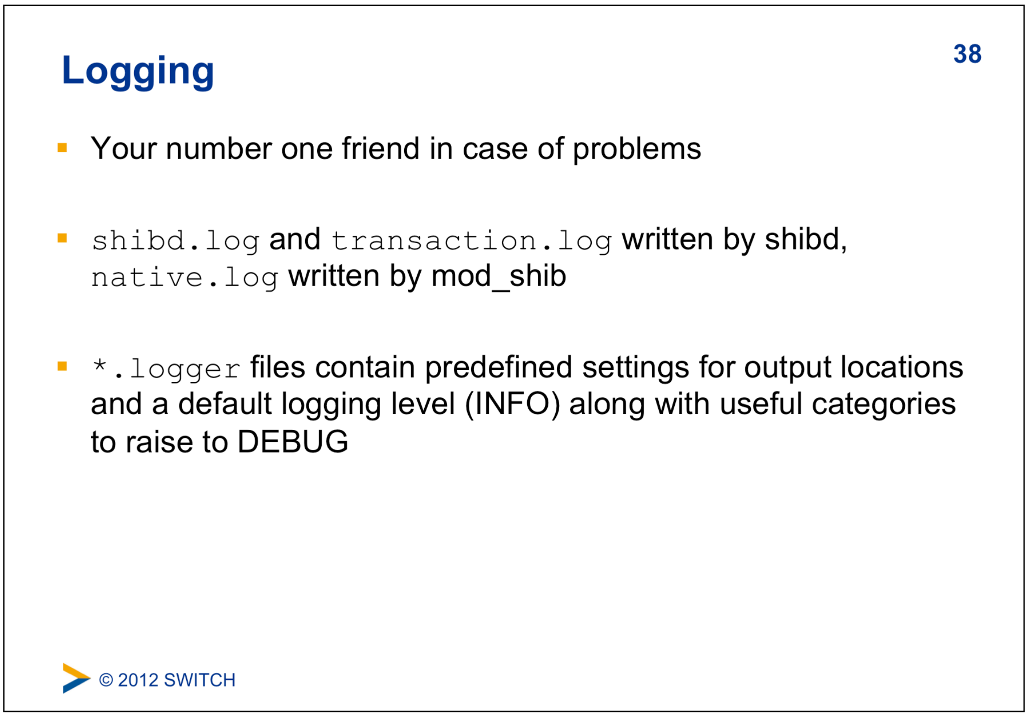
https://wiki.shibboleth.net/confluence/display/SHIB2/NativeSPConfigurationElements

Notes: ______________________________

____

____

____



Logging

- Your number one friend in case of problems
- `shibd.log` and `transaction.log` written by shibd, `native.log` written by mod_shib
- `*.logger` files contain predefined settings for output locations and a default logging level (INFO) along with useful categories to raise to DEBUG

Notes: ______________________________

____

____

____

Logging: Tracing Messages

- Raise categories:

```
$ vim /etc/shibboleth/shibd.logger

Line 2:
log4j.rootCategory=DEBUG, shibd_log, warn_log
Line 14:
# tracing of SAML messages and security policies
log4j.category.OpenSAML.MessageDecoder=DEBUG
log4j.category.OpenSAML.MessageEncoder=DEBUG
log4j.category.OpenSAML.SecurityPolicyRule=DEBUG
```

- To make shibd reload `*.logger` changes:

```
$ touch /etc/shibboleth/shibboleth2.xml  (reloads configuration)
$ tail -f /var/log/shibboleth/shibd.log
```

- Logout (close browser), access `/secure/` and have a look at the log file:

  `https://sp#.example.org/secure`

  You should see the encrypted XML assertion received by your SP.

Notes: ____

____

____

____

SP Metadata Features

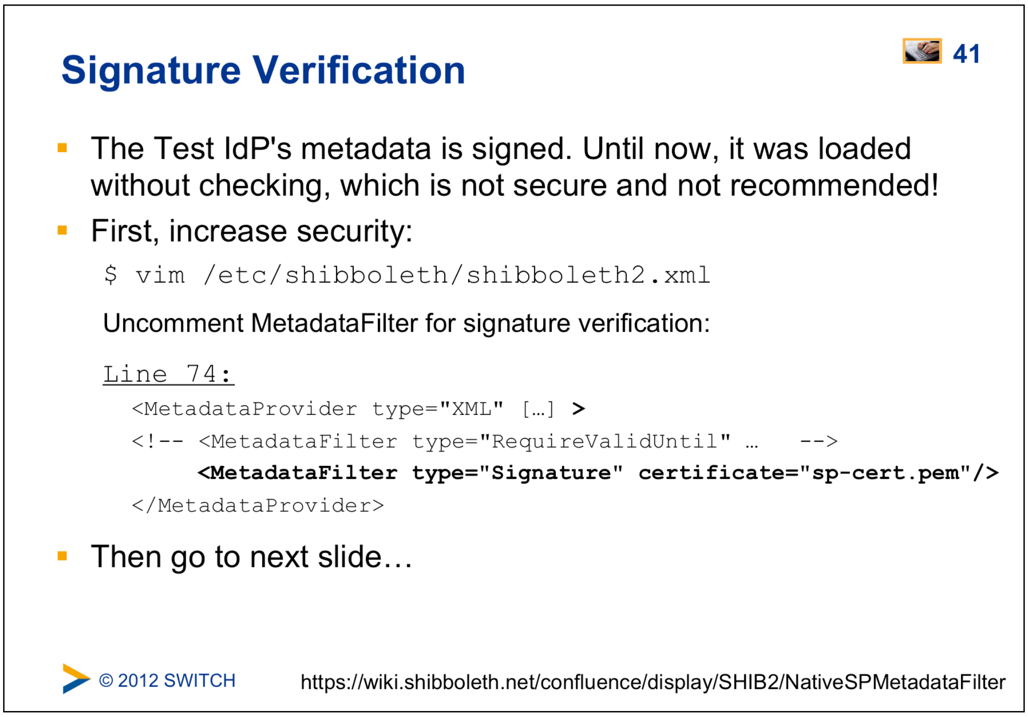
- Metadata describes the other components (IdPs) that the Service Provider can communicate with
- **Four primary methods built-in:**
  - Local metadata file (you download/edit it by hand)
  - Downloaded remotely from URL (periodic refresh, local backup)
  - Dynamic resolution of entityID (=URL), hardly used
  - "Null" source that disables security ("OpenID" model), hardly used
- Security comes from metadata filtering, either by you or the SP:
  - Signature verification
  - Expiration dates
  - White and blacklists

Notes: ____

____

____

____



Signature Verification

- The Test IdP's metadata is signed. Until now, it was loaded without checking, which is not secure and not recommended!
- First, increase security:

```
$ vim /etc/shibboleth/shibboleth2.xml
```

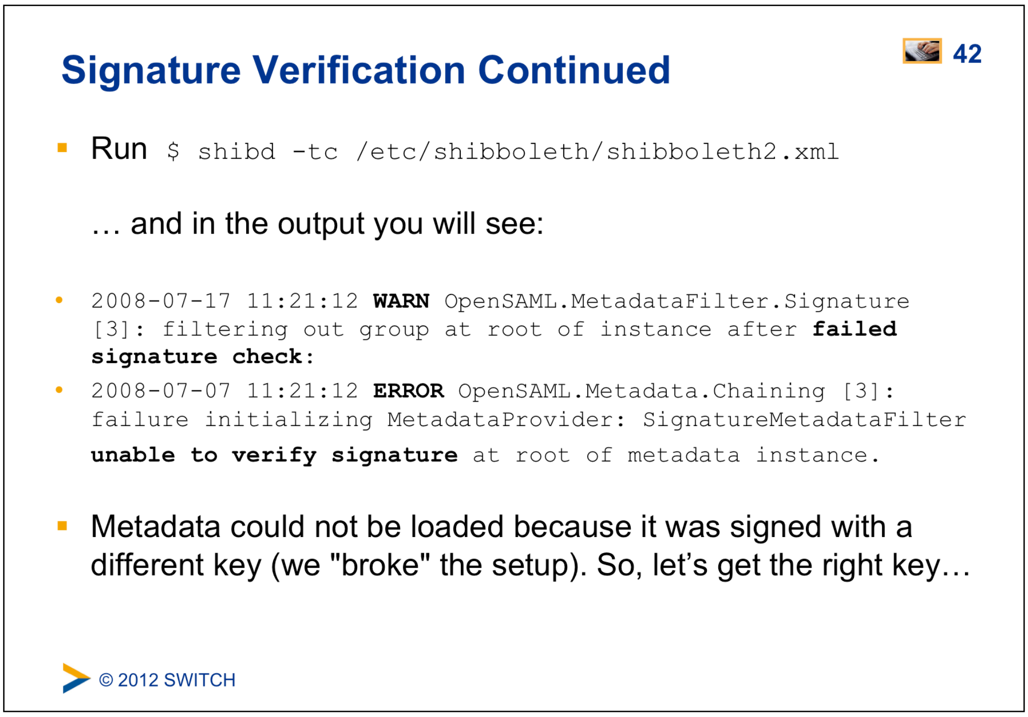
Uncomment MetadataFilter for signature verification:

```
Line 74:
  <MetadataProvider type="XML" […] >
  <!-- <MetadataFilter type="RequireValidUntil" …   -->
       <MetadataFilter type="Signature" certificate="sp-cert.pem"/>
  </MetadataProvider>
```

- Then go to next slide…

https://wiki.shibboleth.net/confluence/display/SHIB2/NativeSPMetadataFilter

Notes:



Signature Verification Continued

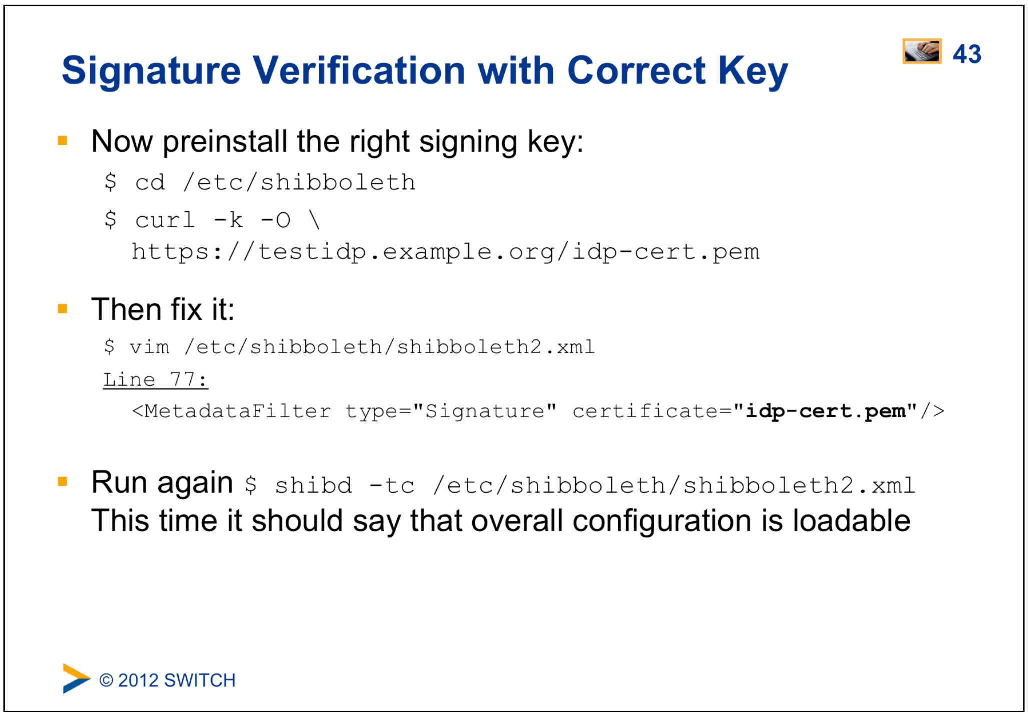
- Run `$ shibd -tc /etc/shibboleth/shibboleth2.xml`

  … and in the output you will see:

```
• 2008-07-17 11:21:12 WARN OpenSAML.MetadataFilter.Signature
  [3]: filtering out group at root of instance after failed
  signature check:
• 2008-07-07 11:21:12 ERROR OpenSAML.Metadata.Chaining [3]:
  failure initializing MetadataProvider: SignatureMetadataFilter
  unable to verify signature at root of metadata instance.
```

- Metadata could not be loaded because it was signed with a different key (we "broke" the setup). So, let’s get the right key…

Notes:



Signature Verification with Correct Key

- Now preinstall the right signing key:

```
$ cd /etc/shibboleth
$ curl -k -O \
  https://testidp.example.org/idp-cert.pem
```

- Then fix it:

```
$ vim /etc/shibboleth/shibboleth2.xml
Line 77:
  <MetadataFilter type="Signature" certificate="idp-cert.pem"/>
```

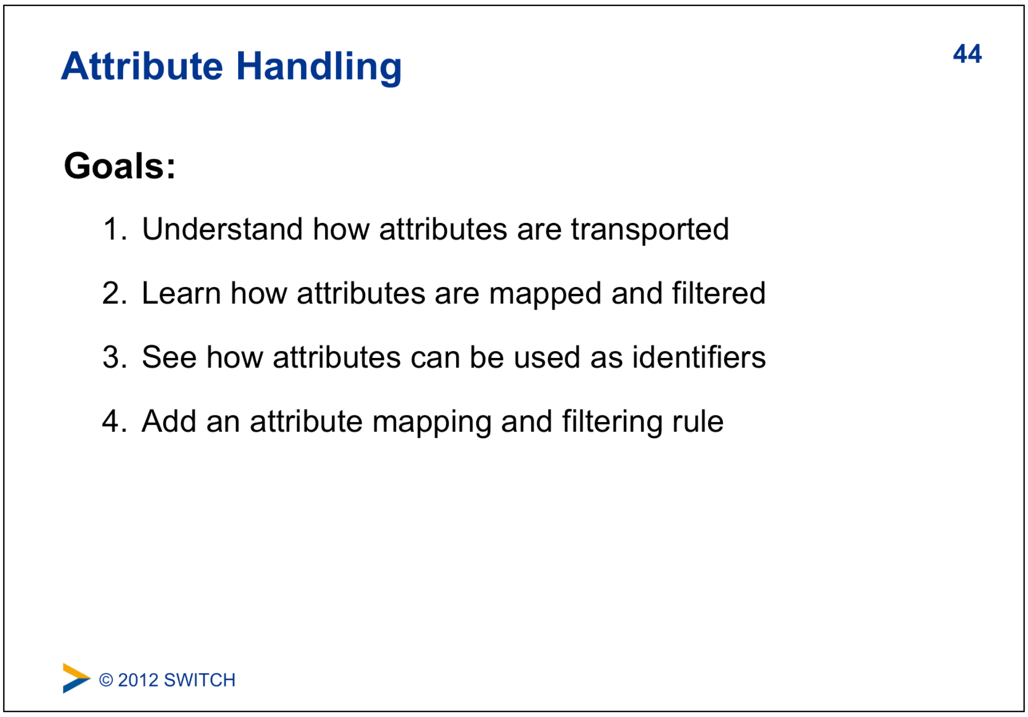
- Run again `$ shibd -tc /etc/shibboleth/shibboleth2.xml`
  This time it should say that overall configuration is loadable

Notes: ______________________________

____

____

____



Attribute Handling

Goals:

1. Understand how attributes are transported
2. Learn how attributes are mapped and filtered
3. See how attributes can be used as identifiers
4. Add an attribute mapping and filtering rule

Notes: ______________________________

____

____

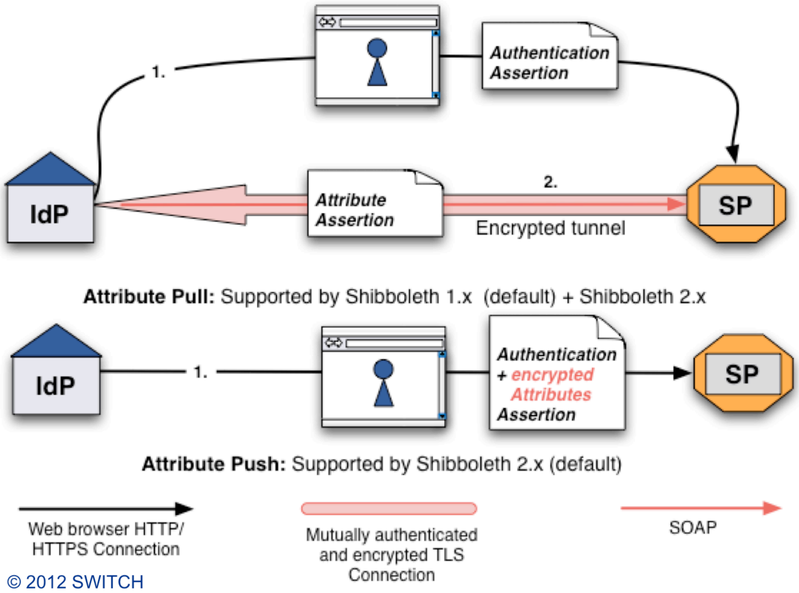
____

SP Attribute Terminology

- **Attribute Push**
  Delivering attributes with SSO assertion via web browser
- **Attribute Pull**
  Querying for attributes after SSO via back-channel (SP -> IdP)
- **Attribute Extraction**
  Decoding SAML information into neutral data structures mapped to environment or header variables
- **Attribute Filtering**
  Blocking invalid, unexpected, or unauthorized values based on application or community criteria
- **Attribute Resolution**
  Resolving a SSO assertion into a set of additional attributes (e.g. queries)

Notes: ______________________________

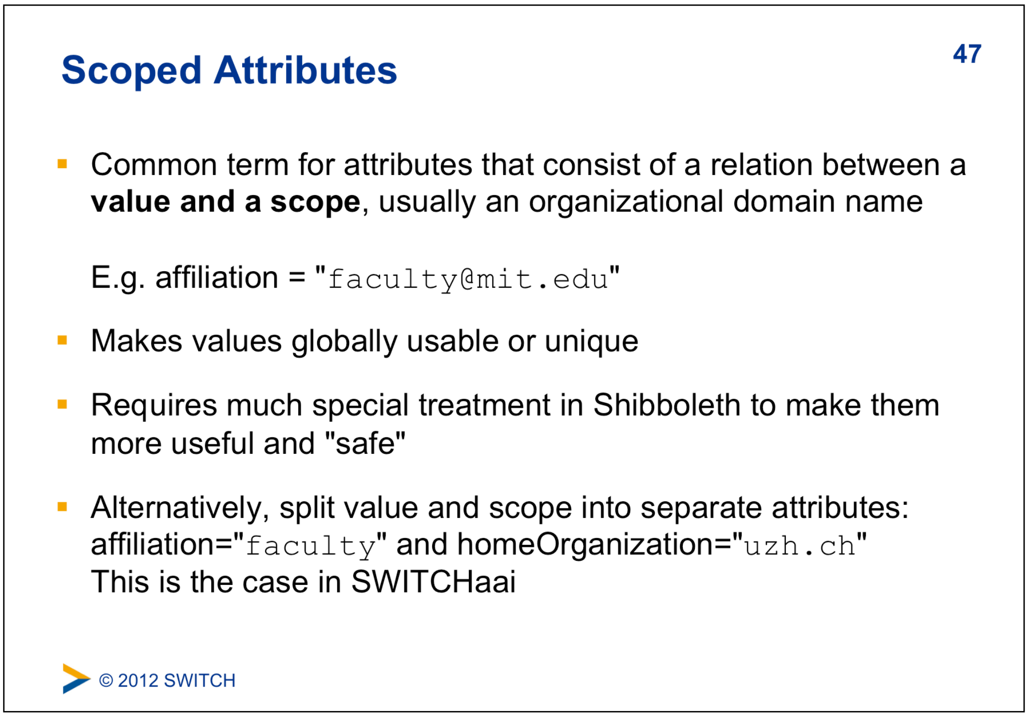
Attribute Pull vs Push


1.
Authentication Assertion
IdP
Attribute Assertion
2.
Encrypted tunnel
SP
IdP
1.
Authentication + encrypted Attributes Assertion
SP
Web browser HTTP/ HTTPS Connection
Mutually authenticated and encrypted TLS Connection
SOAP

Attribute Pull: Supported by Shibboleth 1.x (default) + Shibboleth 2.x

Attribute Push: Supported by Shibboleth 2.x (default)

Notes: ______________________________

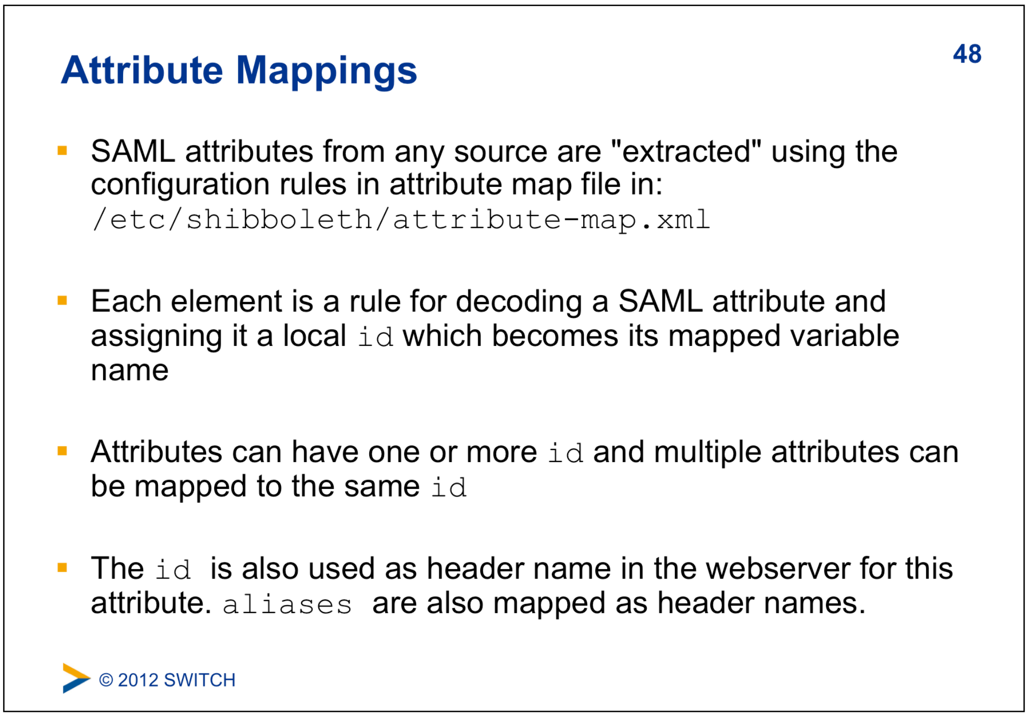


Scoped Attributes

- Common term for attributes that consist of a relation between a **value and a scope**, usually an organizational domain name

  E.g. affiliation = "`faculty@mit.edu`"
- Makes values globally usable or unique
- Requires much special treatment in Shibboleth to make them more useful and "safe"
- Alternatively, split value and scope into separate attributes:
  affiliation="`faculty`" and homeOrganization="`uzh.ch`"
  This is the case in SWITCHaai

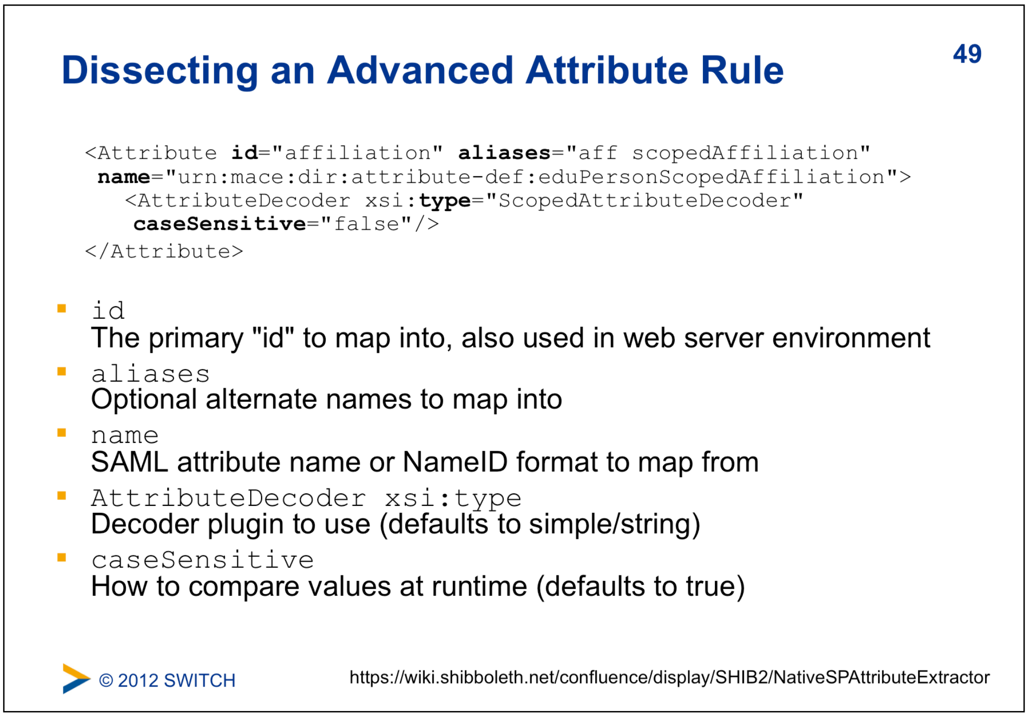
Notes: ____________________



Attribute Mappings

- SAML attributes from any source are "extracted" using the configuration rules in attribute map file in:
  `/etc/shibboleth/attribute-map.xml`
- Each element is a rule for decoding a SAML attribute and assigning it a local `id` which becomes its mapped variable name
- Attributes can have one or more `id` and multiple attributes can be mapped to the same `id`
- The `id` is also used as header name in the webserver for this attribute. `aliases` are also mapped as header names.

Notes: ____________________



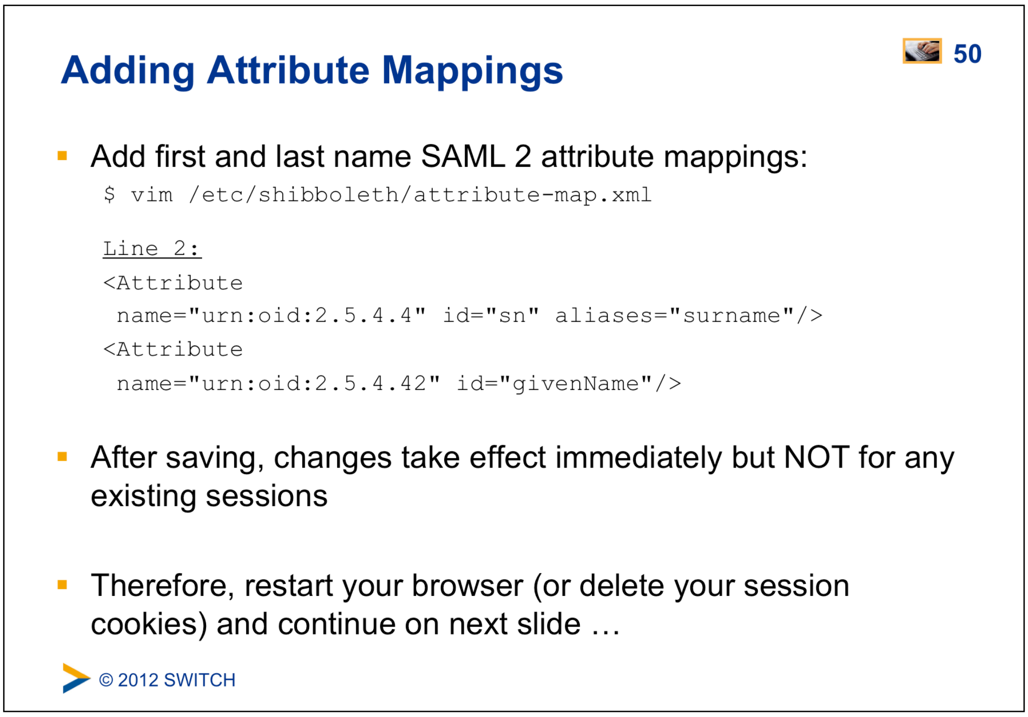
Dissecting an Advanced Attribute Rule

```
<Attribute id="affiliation" aliases="aff scopedAffiliation"
 name="urn:mace:dir:attribute-def:eduPersonScopedAffiliation">
   <AttributeDecoder xsi:type="ScopedAttributeDecoder"
    caseSensitive="false"/>
</Attribute>
```

- `id`
  The primary "id" to map into, also used in web server environment
- `aliases`
  Optional alternate names to map into
- `name`
  SAML attribute name or NameID format to map from
- `AttributeDecoder xsi:type`
  Decoder plugin to use (defaults to simple/string)
- `caseSensitive`
  How to compare values at runtime (defaults to true)

https://wiki.shibboleth.net/confluence/display/SHIB2/NativeSPAttributeExtractor

Notes:



Adding Attribute Mappings

- Add first and last name SAML 2 attribute mappings:

```
$ vim /etc/shibboleth/attribute-map.xml

Line 2:
<Attribute
 name="urn:oid:2.5.4.4" id="sn" aliases="surname"/>
<Attribute
 name="urn:oid:2.5.4.42" id="givenName"/>
```

- After saving, changes take effect immediately but NOT for any existing sessions
- Therefore, restart your browser (or delete your session cookies) and continue on next slide …

Notes:

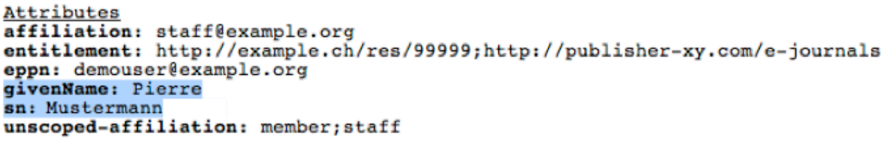
Testing Added Attribute Mapping

- Then access `/secure/` again and log in with demouser/ password. Access should be granted.

- After that, check the Shibboleth Session Handler to see the added attributes are now present:

  `https://sp#.example.org/Shibboleth.sso/Session`

  Now, you also should see the `givenName` and `sn` attributes.



```
Attributes
affiliation: staff@example.org
entitlement: http://example.ch/res/99999;http://publisher-xy.com/e-journals
eppn: demouser@example.org
givenName: Pierre
sn: Mustermann
unscoped-affiliation: member;staff
```

https://wiki.shibboleth.net/confluence/display/SHIB2/NativeSPHandler

Notes:

Uncomment All Attribute Mappings

- Delete the added mapping for sn and givenName on lines 2 and 3 and uncomment all other attribute mappings.

  `$ vim /etc/shibboleth/attribute-map.xml`

  `Around line 54:`
  Remove `<!--`
  `Around line 77:`
  Remove `-->`
  `Around line 78:`
  Remove `<!--`
  `Around line 13§:`
  Remove `-->`

- Then logout, go to `/secure/` and access the Session handler You now also get `cn, mail` and `preferredLanguage`

Notes:

REMOTE_USER

53

- Special single-valued variable that all web applications should support for container-managed authentication of a unique user.
- Any attribute, once extracted/mapped, can be copied to REMOTE_USER
- Multiple attributes can be examined in order of preference, but only the first value will be used.

Notes: ____________________

____

____

____

Changing REMOTE_USER

54

- In case your application needs to have a remote user for authentication, you just could make shibboleth put an attribute (e.g. "mail") as REMOTE_USER:

```
/etc/shibboleth/shibboleth2.xml

Line 25 in <ApplicationDefaults>:
REMOTE_USER="mail eppn persistent-id targeted-id"
```

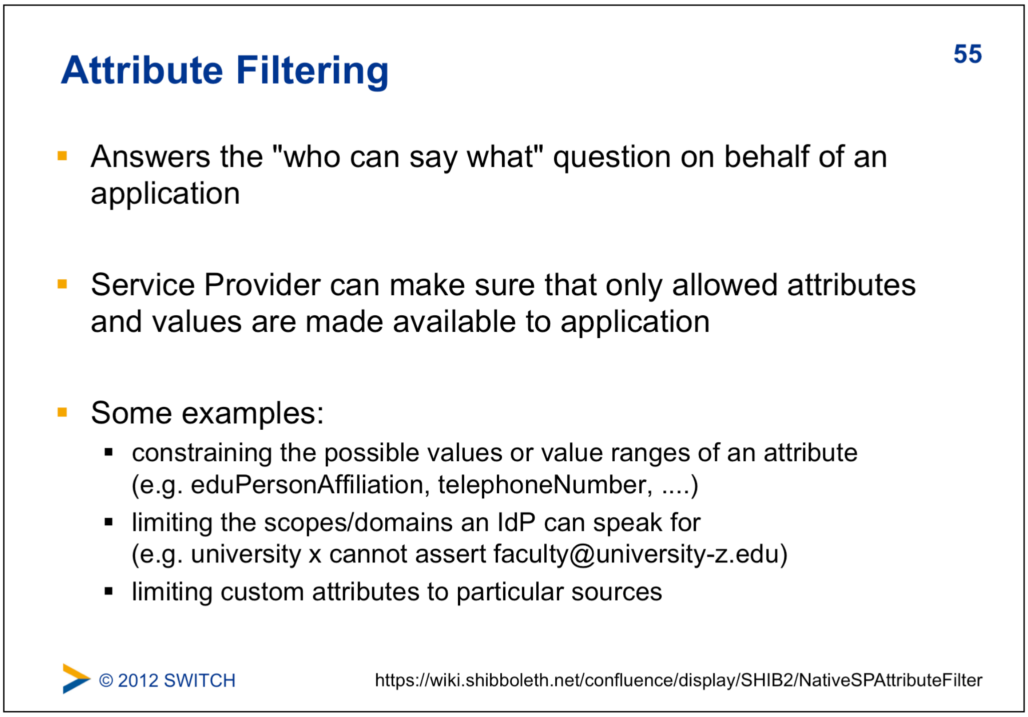
- If mail attribute is available, it will be put into REMOTE_USER
- Attribute `mail` has precedence over `eppn` in this case
- This allows very easy "shibbolization" of some web applications

Notes: ____________________

____

____

____

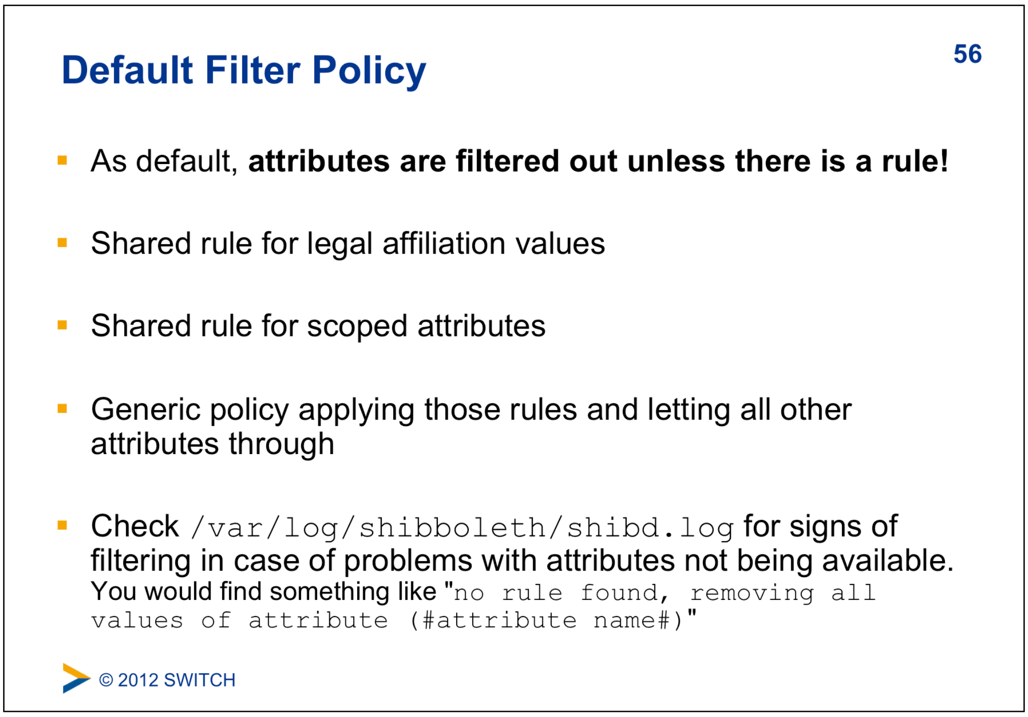


Attribute Filtering

- Answers the "who can say what" question on behalf of an application
- Service Provider can make sure that only allowed attributes and values are made available to application
- Some examples:
  - constraining the possible values or value ranges of an attribute (e.g. eduPersonAffiliation, telephoneNumber, ...)
  - limiting the scopes/domains an IdP can speak for (e.g. university x cannot assert faculty@university-z.edu)
  - limiting custom attributes to particular sources

https://wiki.shibboleth.net/confluence/display/SHIB2/NativeSPAttributeFilter

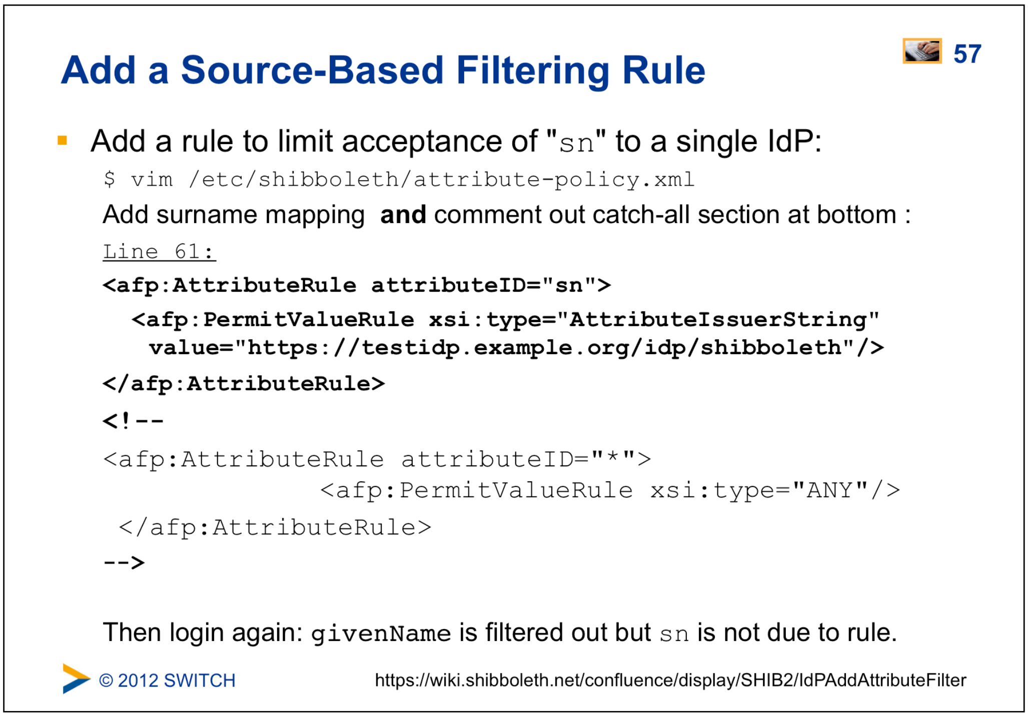
Notes:



Default Filter Policy

- As default, **attributes are filtered out unless there is a rule!**
- Shared rule for legal affiliation values
- Shared rule for scoped attributes
- Generic policy applying those rules and letting all other attributes through
- Check `/var/log/shibboleth/shibd.log` for signs of filtering in case of problems with attributes not being available. You would find something like "`no rule found, removing all values of attribute (#attribute name#)`"

Notes:



57

Add a Source-Based Filtering Rule

- Add a rule to limit acceptance of "`sn`" to a single IdP:

```
$ vim /etc/shibboleth/attribute-policy.xml
```

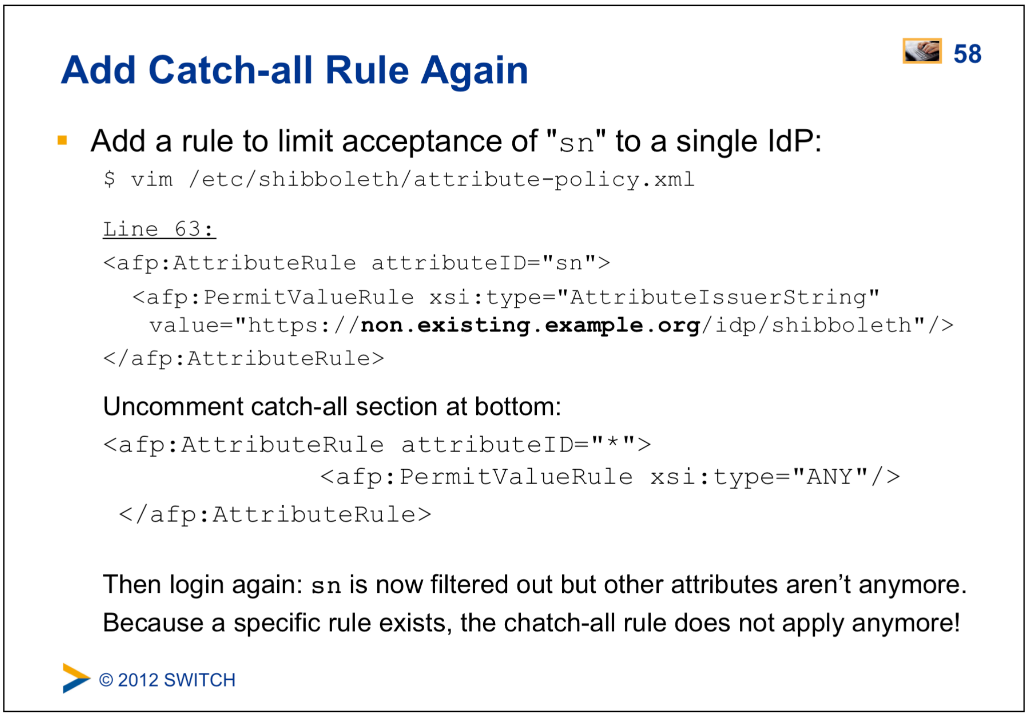
Add surname mapping **and** comment out catch-all section at bottom :

```
Line 61:
<afp:AttributeRule attributeID="sn">
  <afp:PermitValueRule xsi:type="AttributeIssuerString"
   value="https://testidp.example.org/idp/shibboleth"/>
</afp:AttributeRule>
<!--
<afp:AttributeRule attributeID="*">
              <afp:PermitValueRule xsi:type="ANY"/>
 </afp:AttributeRule>
-->
```

Then login again: `givenName` is filtered out but `sn` is not due to rule.

https://wiki.shibboleth.net/confluence/display/SHIB2/IdPAddAttributeFilter

Notes: ____________________



58

Add Catch-all Rule Again

- Add a rule to limit acceptance of "`sn`" to a single IdP:

```
$ vim /etc/shibboleth/attribute-policy.xml

Line 63:
<afp:AttributeRule attributeID="sn">
  <afp:PermitValueRule xsi:type="AttributeIssuerString"
   value="https://non.existing.example.org/idp/shibboleth"/>
</afp:AttributeRule>
```

Uncomment catch-all section at bottom:

```
<afp:AttributeRule attributeID="*">
              <afp:PermitValueRule xsi:type="ANY"/>
 </afp:AttributeRule>
```

Then login again: `sn` is now filtered out but other attributes aren't anymore.
Because a specific rule exists, the chatch-all rule does not apply anymore!

Notes: ____________________

Remove Specific Rule

- Remove rule for (non-) acceptance of "`sn`":

```
$ vim /etc/shibboleth/attribute-policy.xml
```

Delete rule for `sn` (lines 62-64)

- Save file and access `/secure` again
- Now you should see the `sn` attribute again

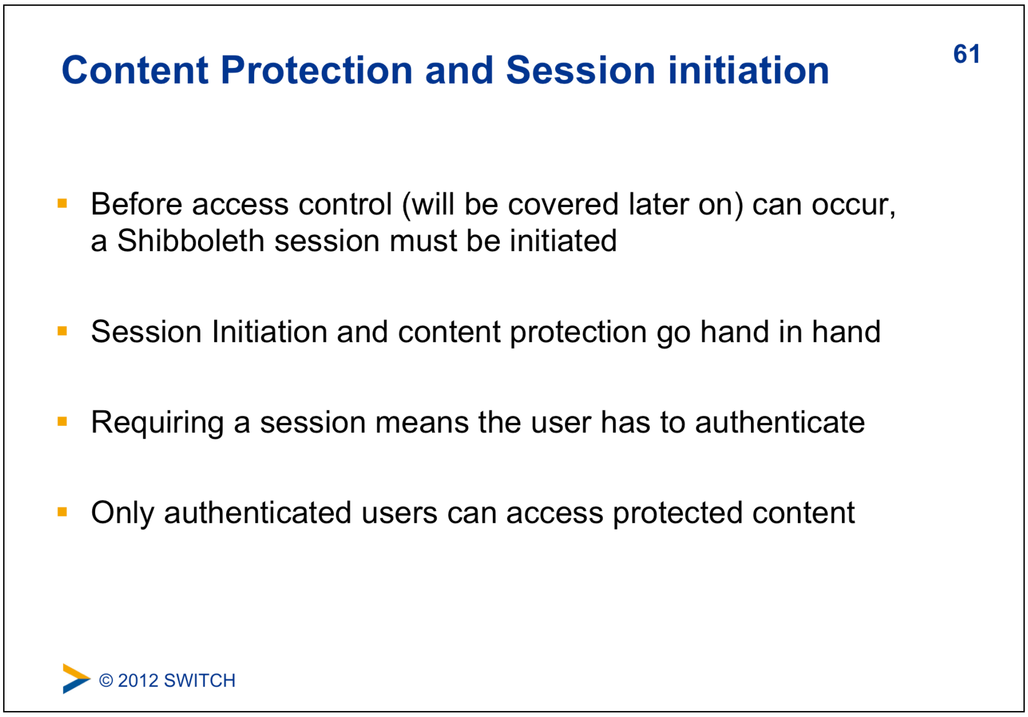
Notes: ______________________________

Session Initiation

Goals:

1. Learn how to initiate a Shibboleth session
2. Understand their advantages/disadvantages
3. Know where to require a session, what to protect

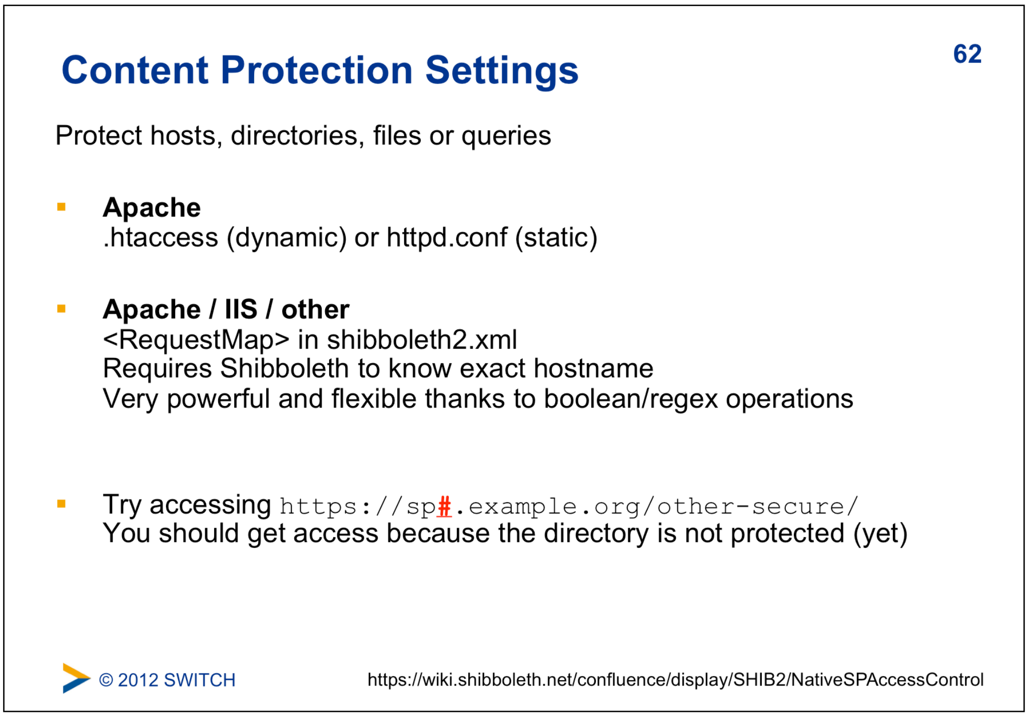
Notes: ______________________________



Content Protection and Session initiation

- Before access control (will be covered later on) can occur, a Shibboleth session must be initiated
- Session Initiation and content protection go hand in hand
- Requiring a session means the user has to authenticate
- Only authenticated users can access protected content

Notes:



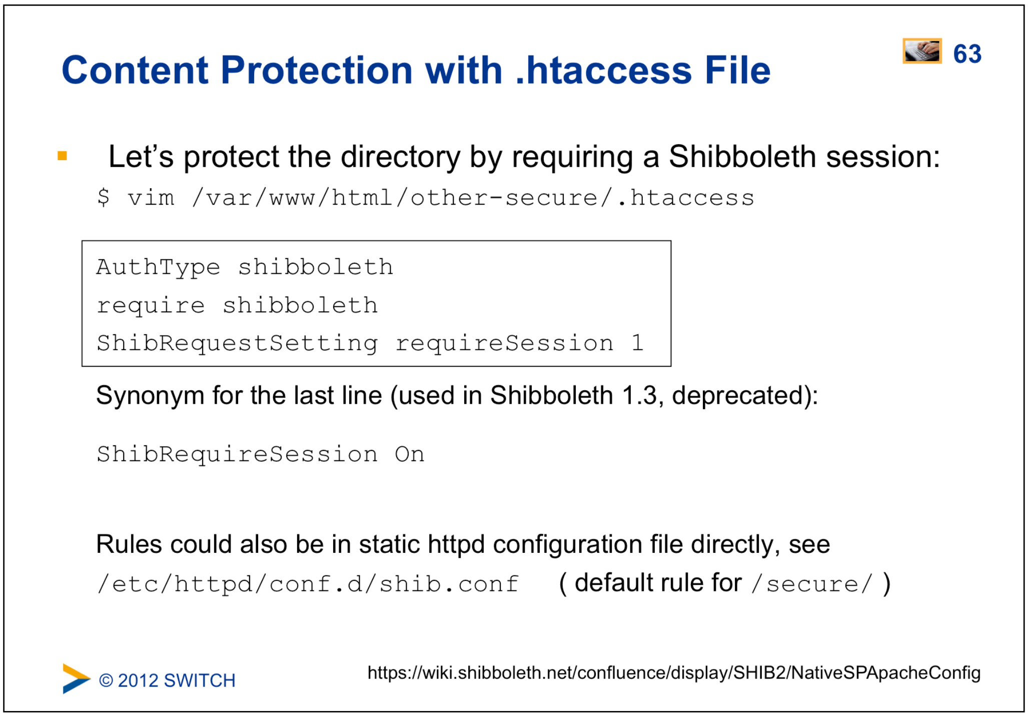
Content Protection Settings

Protect hosts, directories, files or queries

- **Apache**
  .htaccess (dynamic) or httpd.conf (static)
- **Apache / IIS / other**
  <RequestMap> in shibboleth2.xml
  Requires Shibboleth to know exact hostname
  Very powerful and flexible thanks to boolean/regex operations
- Try accessing `https://sp#.example.org/other-secure/`
  You should get access because the directory is not protected (yet)

https://wiki.shibboleth.net/confluence/display/SHIB2/NativeSPAccessControl

Notes:



Content Protection with .htaccess File

- Let’s protect the directory by requiring a Shibboleth session:

```
$ vim /var/www/html/other-secure/.htaccess
```

```
AuthType shibboleth
require shibboleth
ShibRequestSetting requireSession 1
```

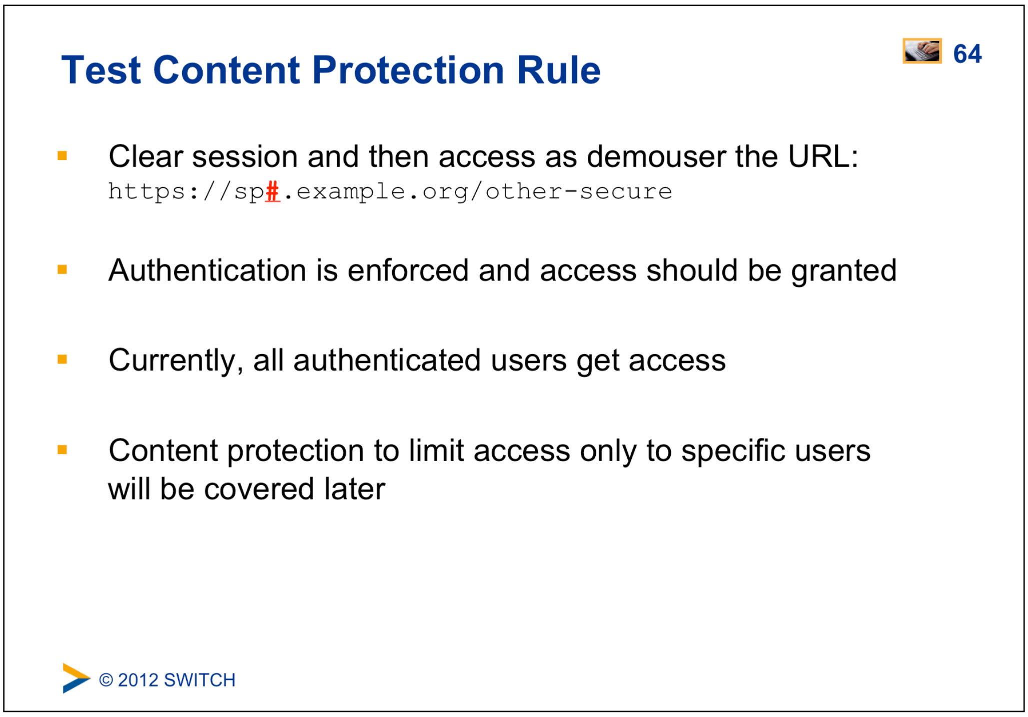
Synonym for the last line (used in Shibboleth 1.3, deprecated):

```
ShibRequireSession On
```

Rules could also be in static httpd configuration file directly, see `/etc/httpd/conf.d/shib.conf` (default rule for `/secure/`)

https://wiki.shibboleth.net/confluence/display/SHIB2/NativeSPApacheConfig

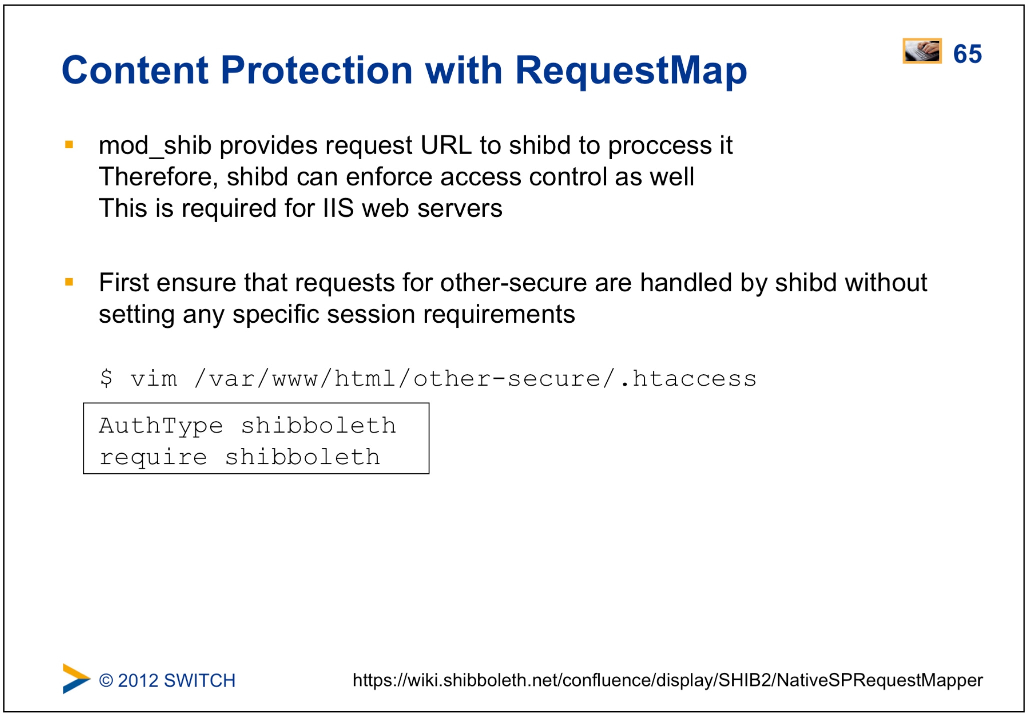
Notes: ______________________________



Test Content Protection Rule

- Clear session and then access as demouser the URL: `https://sp#.example.org/other-secure`
- Authentication is enforced and access should be granted
- Currently, all authenticated users get access
- Content protection to limit access only to specific users will be covered later

Notes: ______________________________



Content Protection with RequestMap

65

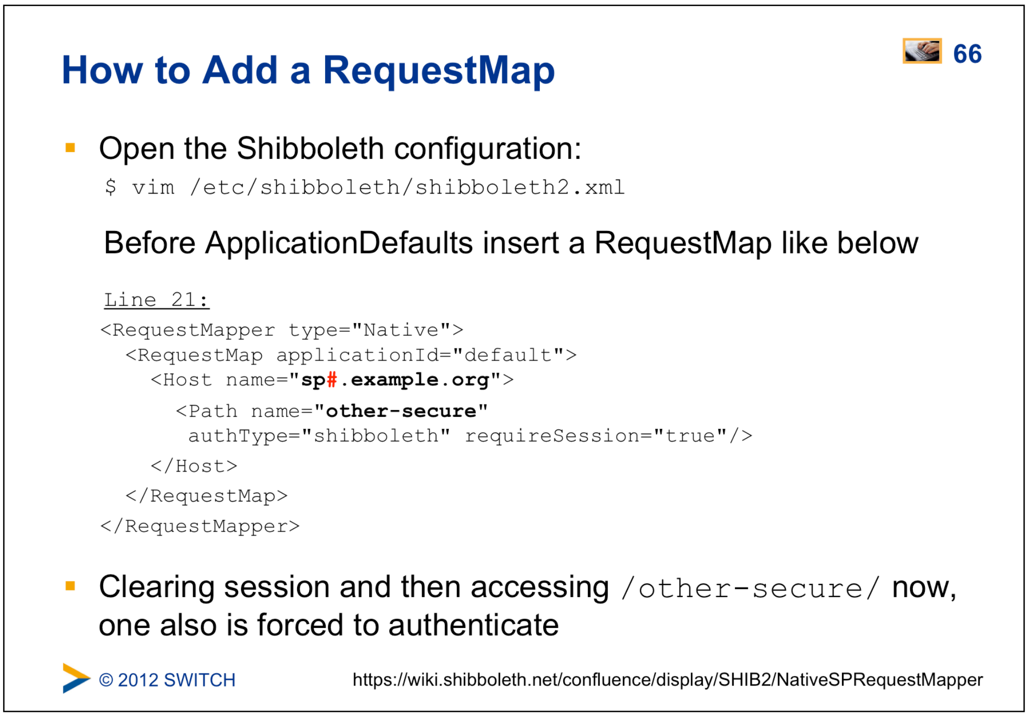
- mod_shib provides request URL to shibd to proccess it
  Therefore, shibd can enforce access control as well
  This is required for IIS web servers
- First ensure that requests for other-secure are handled by shibd without setting any specific session requirements

```
$ vim /var/www/html/other-secure/.htaccess
```

```
AuthType shibboleth
require shibboleth
```

https://wiki.shibboleth.net/confluence/display/SHIB2/NativeSPRequestMapper

Notes:



How to Add a RequestMap

66

- Open the Shibboleth configuration:

```
$ vim /etc/shibboleth/shibboleth2.xml
```

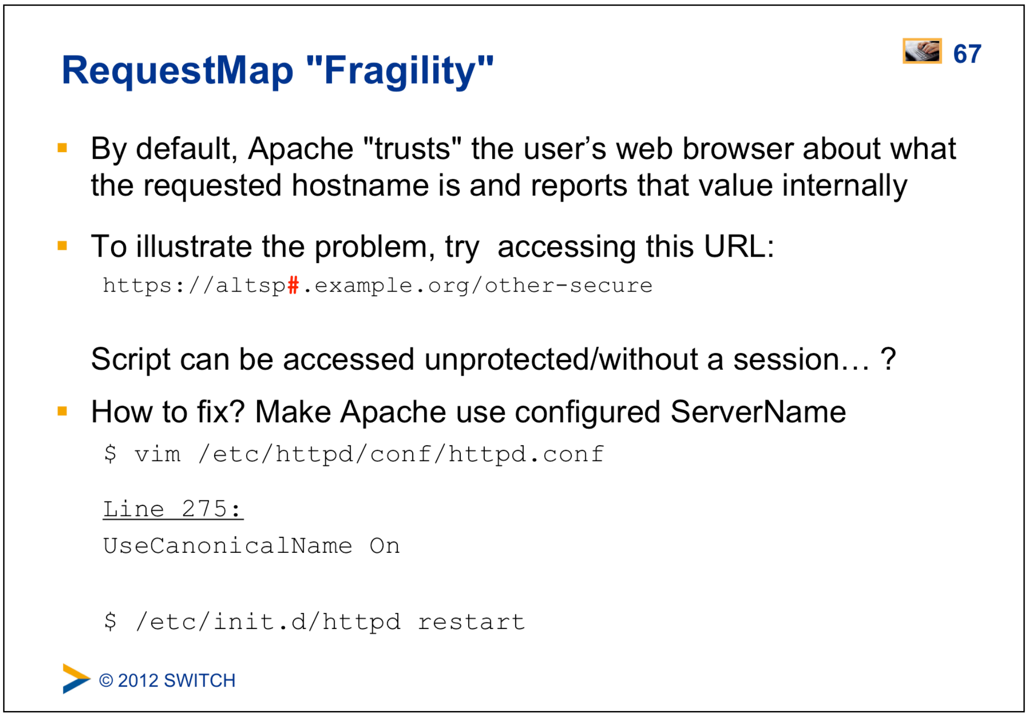
Before ApplicationDefaults insert a RequestMap like below

```
Line 21:
<RequestMapper type="Native">
  <RequestMap applicationId="default">
    <Host name="sp#.example.org">
      <Path name="other-secure"
       authType="shibboleth" requireSession="true"/>
    </Host>
  </RequestMap>
</RequestMapper>
```

- Clearing session and then accessing `/other-secure/` now, one also is forced to authenticate

https://wiki.shibboleth.net/confluence/display/SHIB2/NativeSPRequestMapper

Notes:



RequestMap "Fragility"

67

- By default, Apache "trusts" the user’s web browser about what the requested hostname is and reports that value internally
- To illustrate the problem, try accessing this URL:

```
https://altsp#.example.org/other-secure
```

  Script can be accessed unprotected/without a session… ?

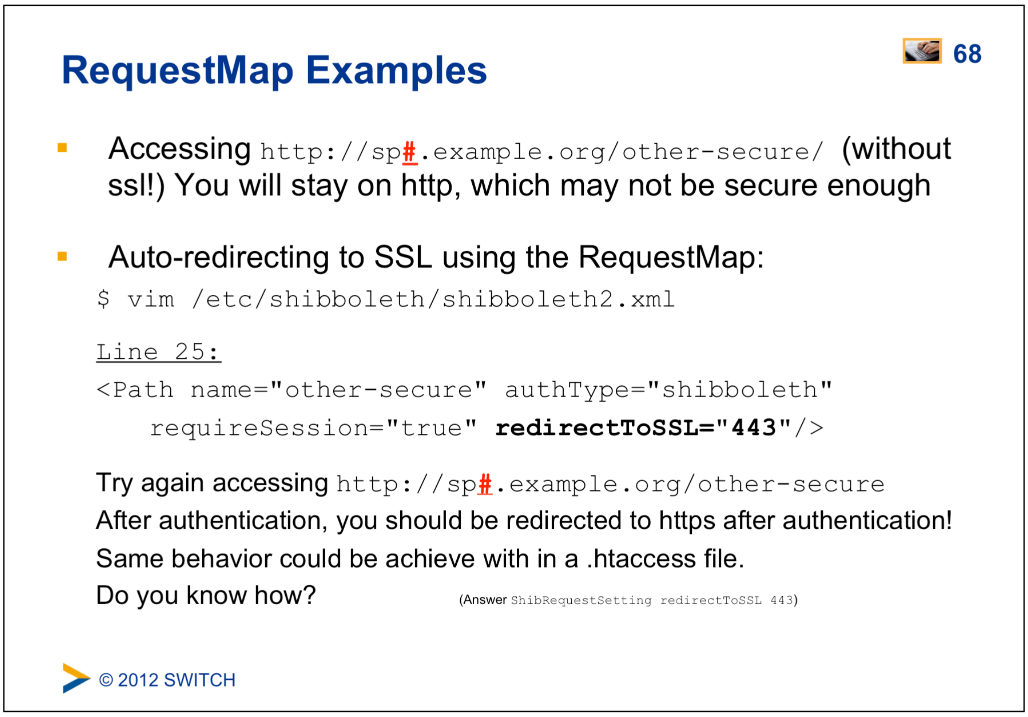
- How to fix? Make Apache use configured ServerName

```
$ vim /etc/httpd/conf/httpd.conf

Line 275:
UseCanonicalName On

$ /etc/init.d/httpd restart
```

Notes:



RequestMap Examples

68

- Accessing `http://sp#.example.org/other-secure/` (without ssl!) You will stay on http, which may not be secure enough
- Auto-redirecting to SSL using the RequestMap:

```
$ vim /etc/shibboleth/shibboleth2.xml

Line 25:
<Path name="other-secure" authType="shibboleth"
   requireSession="true" redirectToSSL="443"/>
```

Try again accessing `http://sp#.example.org/other-secure`
After authentication, you should be redirected to https after authentication!
Same behavior could be achieve with in a .htaccess file.
Do you know how? (Answer `ShibRequestSetting redirectToSSL 443`)

Notes:

Other Content Settings

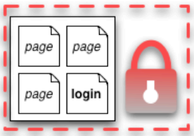
- Requesting types of authentication
  - E.g enforce X.509 user certificate authentication
- Custom error handling pages to use
- Redirection-based error handling
  - In case of an error, redirect user to custom error web page with error message/type as GET arguments
- **forceAuthn**
  - Disable Single-Sign on and force a re-authentication
- **isPassive**
  - Check whether a user has an SSO session and if he has, automatically create a session on SP without any user interaction
- Use a specific IdP to use for authentication

https://wiki.shibboleth.net/confluence/display/SHIB2/NativeSPContentSettings

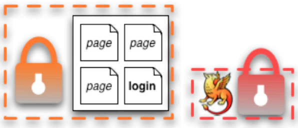
Notes:

Where to Require a Shibboleth Session

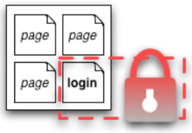
- **Whole application with "required" Shibboleth session**
  - Easiest way to protect a set of documents
  - No other authentication methods possible like this
  - Problems with lost HTTP POST requests



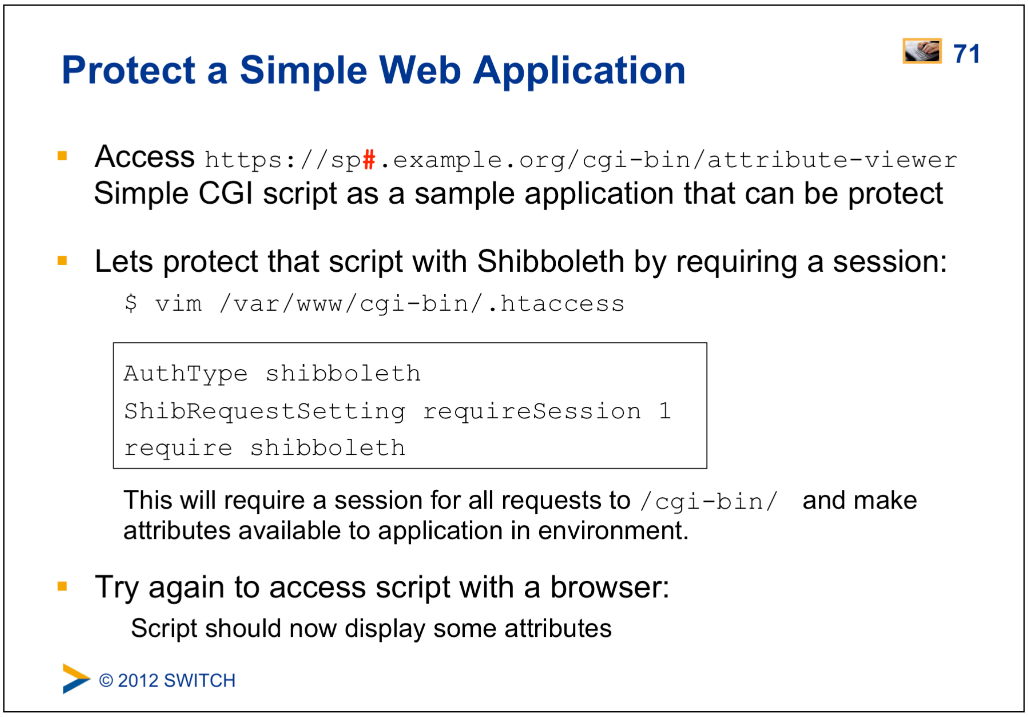
- **Whole application with "lazy" Shibboleth session**
  - Also allows for other authentication methods
  - Authorization can only be done in application



- **Only page that sets up application session**
  - Well-suited for dual login
  - Application can control session time-out
  - **Generally the best solution**



Notes:



71

Protect a Simple Web Application

- Access `https://sp#.example.org/cgi-bin/attribute-viewer`
  Simple CGI script as a sample application that can be protect
- Lets protect that script with Shibboleth by requiring a session:

```
$ vim /var/www/cgi-bin/.htaccess
```

```
AuthType shibboleth
ShibRequestSetting requireSession 1
require shibboleth
```

This will require a session for all requests to `/cgi-bin/` and make attributes available to application in environment.

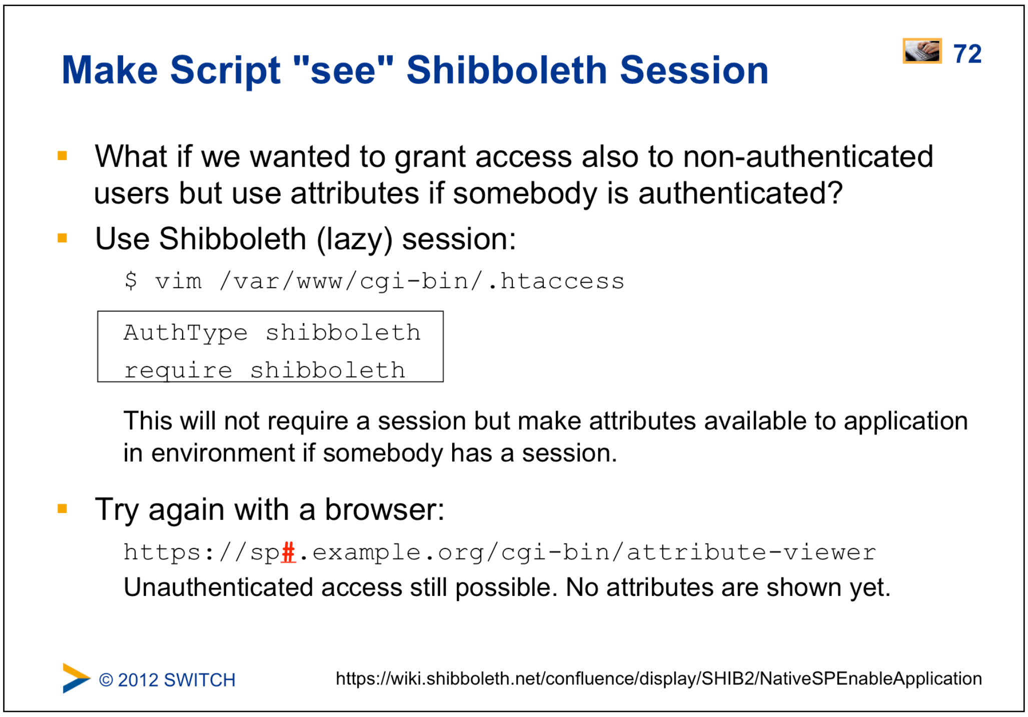
- Try again to access script with a browser:
  Script should now display some attributes

Notes: ______________________________

____

____

____



72

Make Script "see" Shibboleth Session

- What if we wanted to grant access also to non-authenticated users but use attributes if somebody is authenticated?
- Use Shibboleth (lazy) session:

```
$ vim /var/www/cgi-bin/.htaccess
```

```
AuthType shibboleth
require shibboleth
```

This will not require a session but make attributes available to application in environment if somebody has a session.

- Try again with a browser:
  `https://sp#.example.org/cgi-bin/attribute-viewer`
  Unauthenticated access still possible. No attributes are shown yet.

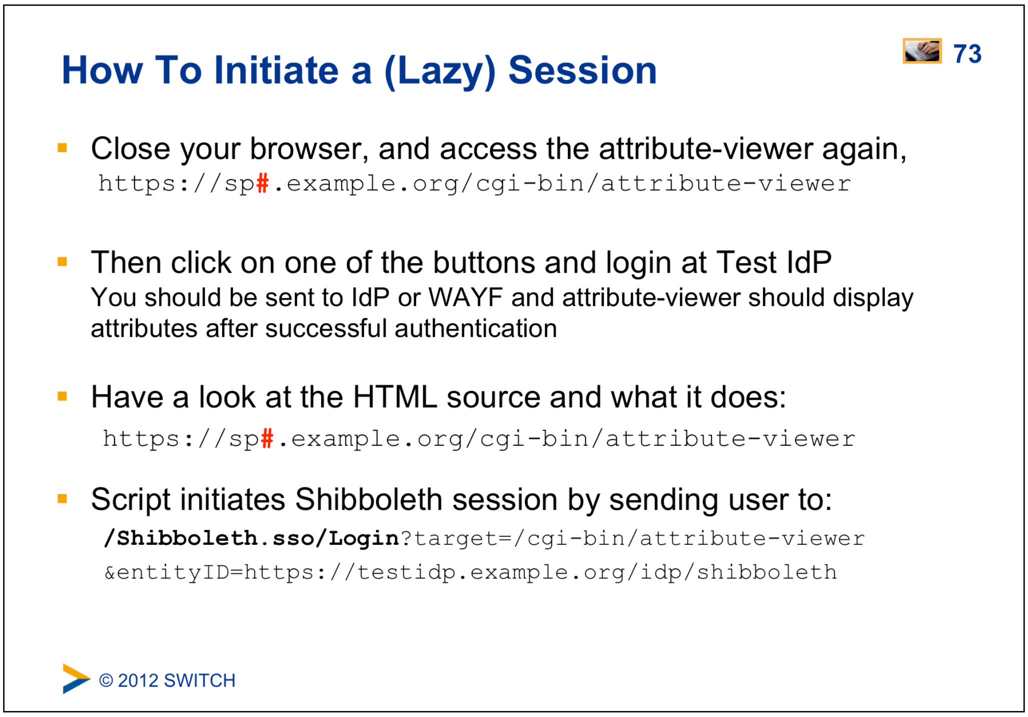
https://wiki.shibboleth.net/confluence/display/SHIB2/NativeSPEnableApplication

Notes: ______________________________

____

____

____



How To Initiate a (Lazy) Session

73

- Close your browser, and access the attribute-viewer again,
  `https://sp#.example.org/cgi-bin/attribute-viewer`

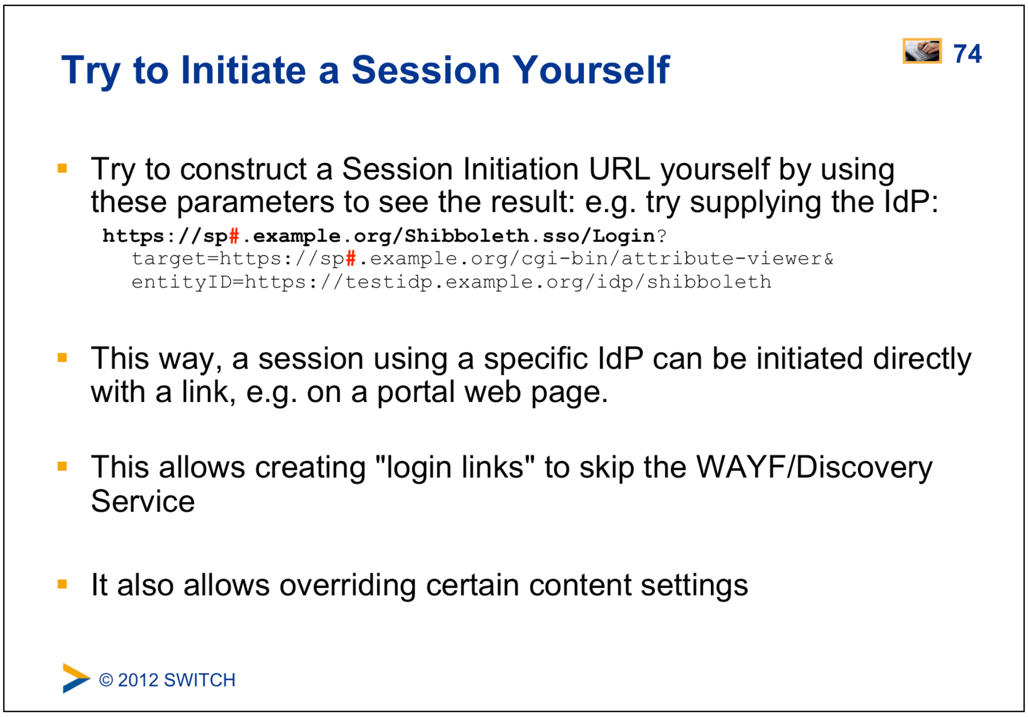
- Then click on one of the buttons and login at Test IdP
  You should be sent to IdP or WAYF and attribute-viewer should display attributes after successful authentication

- Have a look at the HTML source and what it does:
  `https://sp#.example.org/cgi-bin/attribute-viewer`

- Script initiates Shibboleth session by sending user to:

```
/Shibboleth.sso/Login?target=/cgi-bin/attribute-viewer
&entityID=https://testidp.example.org/idp/shibboleth
```

Notes:



Try to Initiate a Session Yourself

74

- Try to construct a Session Initiation URL yourself by using these parameters to see the result: e.g. try supplying the IdP:

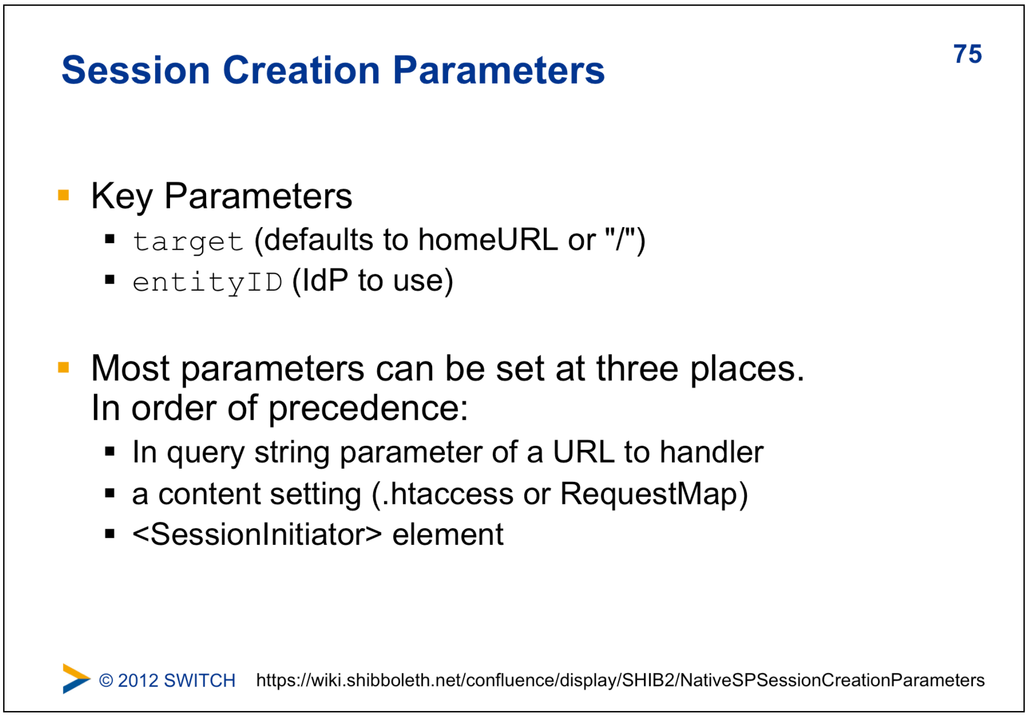
```
https://sp#.example.org/Shibboleth.sso/Login?
   target=https://sp#.example.org/cgi-bin/attribute-viewer&
   entityID=https://testidp.example.org/idp/shibboleth
```

- This way, a session using a specific IdP can be initiated directly with a link, e.g. on a portal web page.

- This allows creating "login links" to skip the WAYF/Discovery Service

- It also allows overriding certain content settings

Notes:

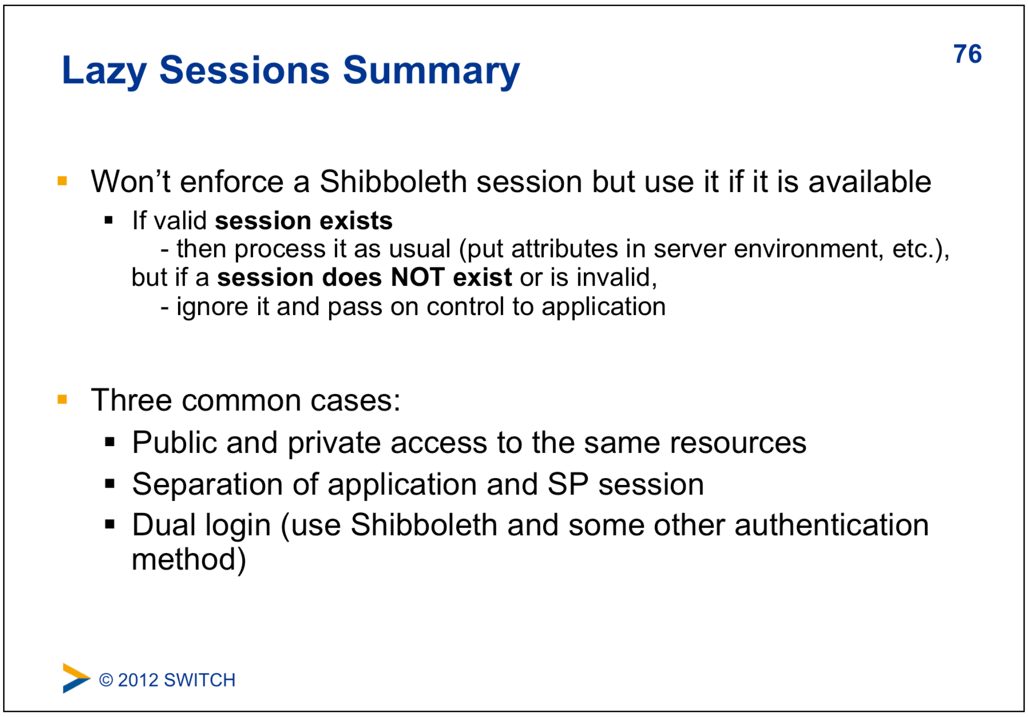


Session Creation Parameters

- Key Parameters
  - `target` (defaults to homeURL or "/")
  - `entityID` (IdP to use)
- Most parameters can be set at three places. In order of precedence:
  - In query string parameter of a URL to handler
  - a content setting (.htaccess or RequestMap)
  - <SessionInitiator> element

https://wiki.shibboleth.net/confluence/display/SHIB2/NativeSPSessionCreationParameters

Notes:



Lazy Sessions Summary

- Won’t enforce a Shibboleth session but use it if it is available
  - If valid **session exists**
    - then process it as usual (put attributes in server environment, etc.),
    but if a **session does NOT exist** or is invalid,
    - ignore it and pass on control to application
- Three common cases:
  - Public and private access to the same resources
  - Separation of application and SP session
  - Dual login (use Shibboleth and some other authentication method)

Notes:

Using Lazy Sessions

- In place of an API to "doLogin", the SP uses redirects:
  `https://testsp1.example.org/Shibboleth.sso/Login`
- When you/your application want a login to happen, redirect the browser to a SessionInitiator (`/Login` by convention) with any parameters you want to supply

Notes: ______________________________

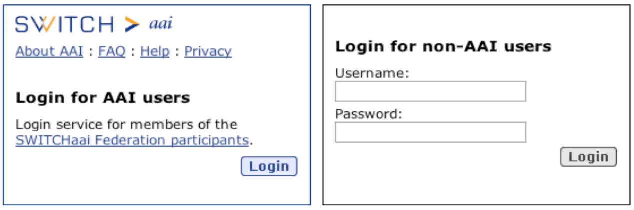
____

____

____

Some Concerns Regarding Dual Login

- Can be a viable option in case application must also be used by non-Shibboleth users
- Generally not recommended due to issues with:
  - **Usability:** Difficult to teach the users how to authenticate
  - **Security:** Shibboleth users shouldn't enter their password in the login form for the non-Shibboleth users…



Login for AAI users

Login service for members of the SWITCHaai Federation participants.

Login for non-AAI users

Username:

Password:

Notes: ______________________________

____

____

____

Virtual Home Organization and Guest Login

Excursion about dealing with user who don’t have an AAI account already.

Please consult the table of contents to find this presentation in your hand-outs.

Notes: ____

____

____

____

Access Control

Goals:

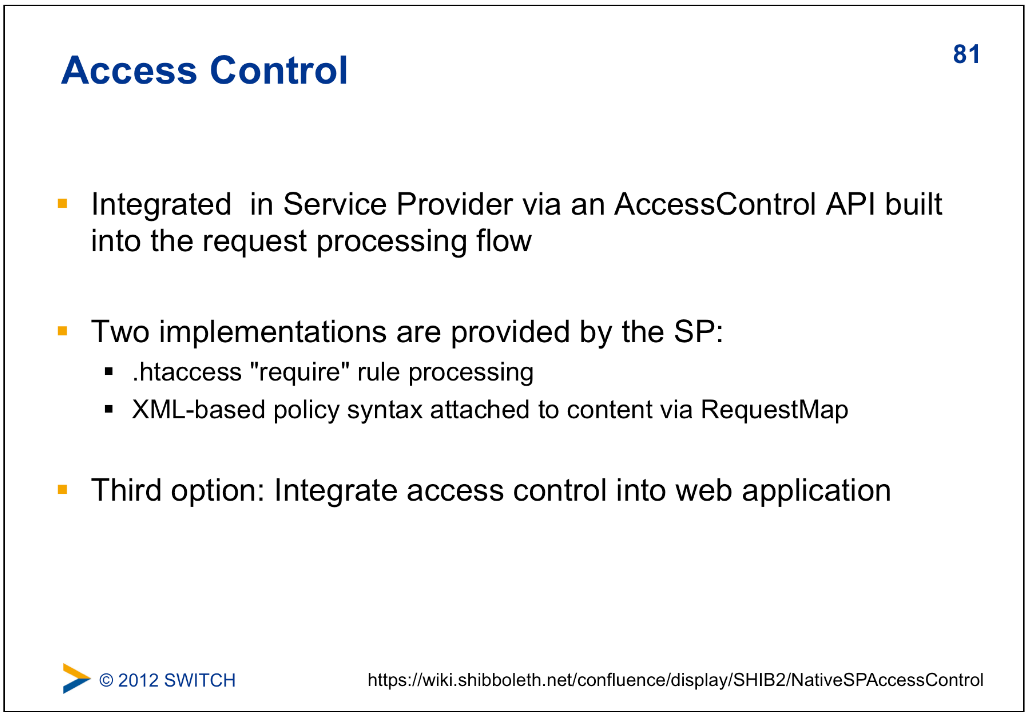
1. Create some simple access control rules
2. Get an overview about the three ways to authorize users
3. Understand their advantages/disadvantages

Notes: ____

____

____

____

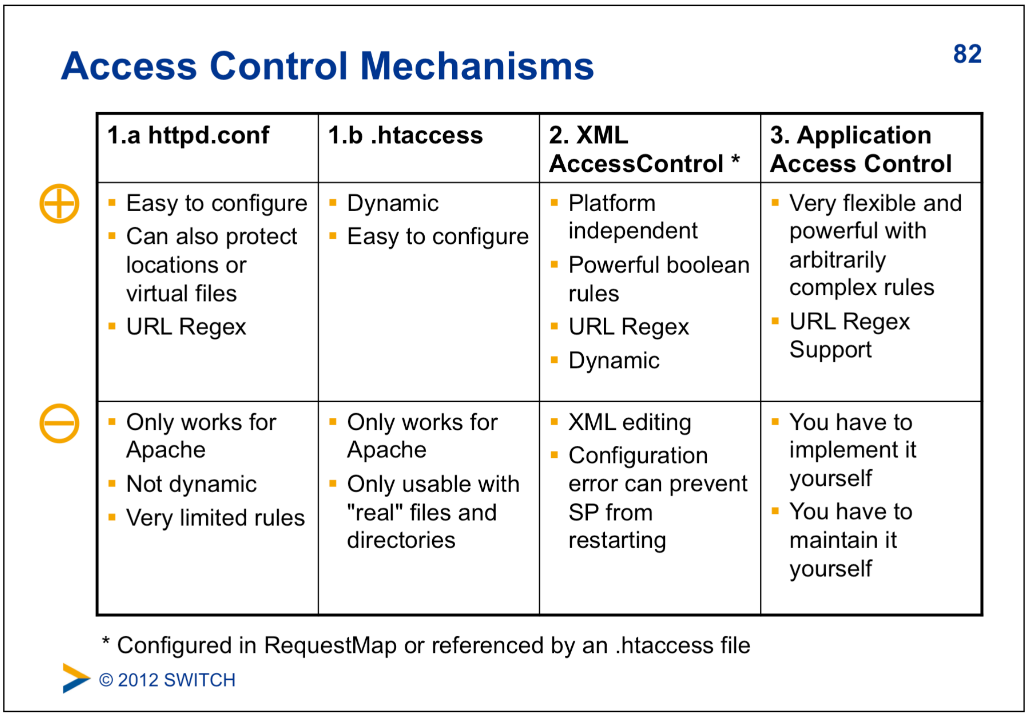


Access Control

- Integrated in Service Provider via an AccessControl API built into the request processing flow
- Two implementations are provided by the SP:
  - .htaccess "require" rule processing
  - XML-based policy syntax attached to content via RequestMap
- Third option: Integrate access control into web application

https://wiki.shibboleth.net/confluence/display/SHIB2/NativeSPAccessControl

Notes:

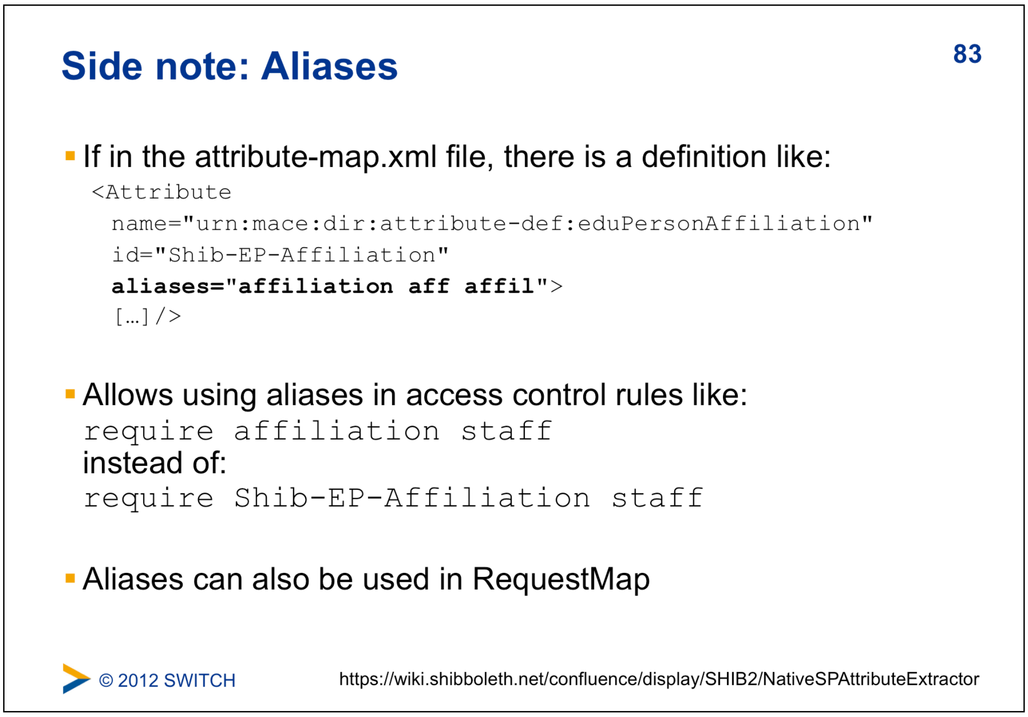


Access Control Mechanisms

	1.a httpd.conf	1.b .htaccess	2. XML AccessControl *	3. Application Access Control
⊕	Easy to configure Can also protect locations or virtual files URL Regex	Dynamic Easy to configure	Platform independent Powerful boolean rules URL Regex Dynamic	Very flexible and powerful with arbitrarily complex rules URL Regex Support
⊖	Only works for Apache Not dynamic Very limited rules	Only works for Apache Only usable with "real" files and directories	XML editing Configuration error can prevent SP from restarting	You have to implement it yourself You have to maintain it yourself

* Configured in RequestMap or referenced by an .htaccess file

Notes:



Side note: Aliases

- If in the attribute-map.xml file, there is a definition like:

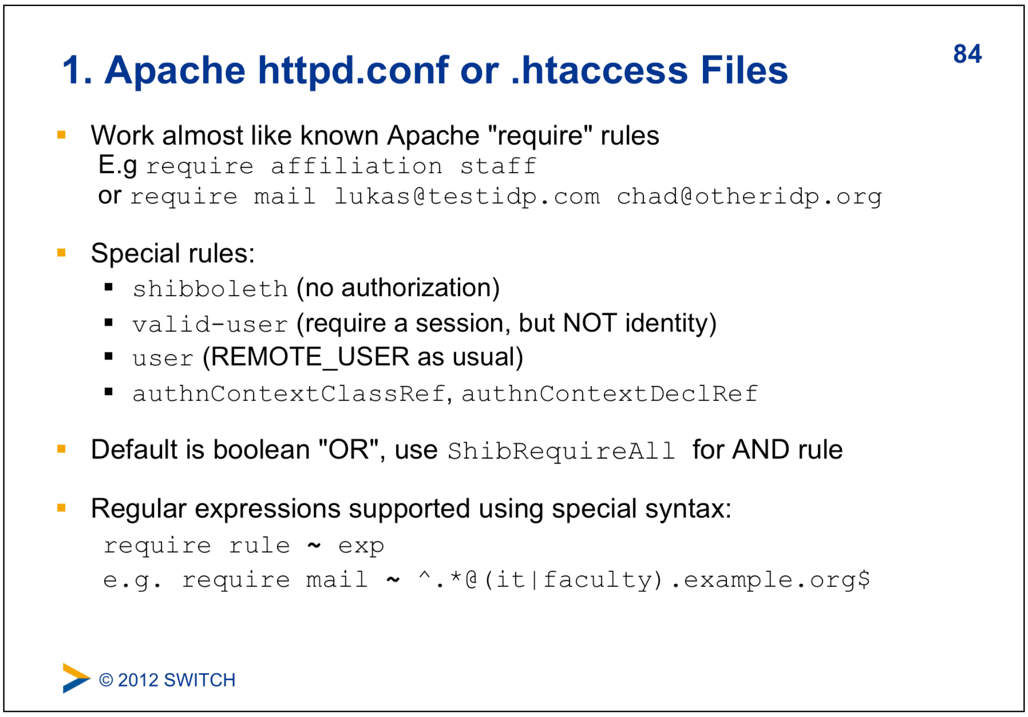
```
<Attribute
  name="urn:mace:dir:attribute-def:eduPersonAffiliation"
  id="Shib-EP-Affiliation"
  aliases="affiliation aff affil">
  […]/>
```

- Allows using aliases in access control rules like:
  `require affiliation staff`
  instead of:
  `require Shib-EP-Affiliation staff`

- Aliases can also be used in RequestMap

https://wiki.shibboleth.net/confluence/display/SHIB2/NativeSPAttributeExtractor

Notes:



1. Apache httpd.conf or .htaccess Files

- Work almost like known Apache "require" rules
  E.g `require affiliation staff`
  or `require mail lukas@testidp.com chad@otheridp.org`

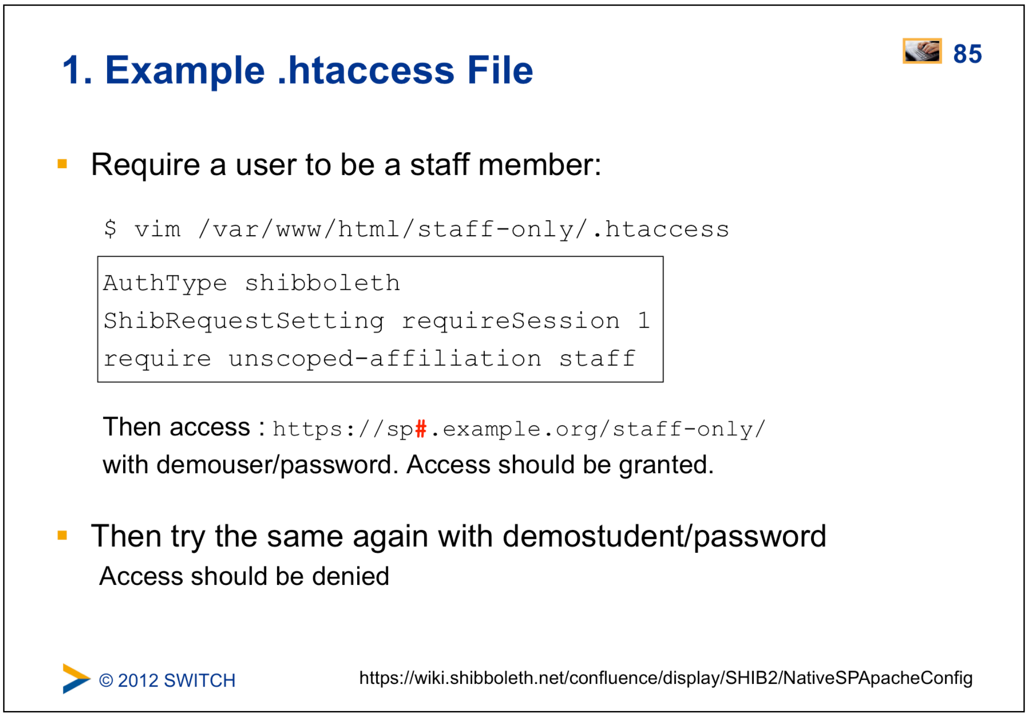
- Special rules:
  - `shibboleth` (no authorization)
  - `valid-user` (require a session, but NOT identity)
  - `user` (REMOTE_USER as usual)
  - `authnContextClassRef`, `authnContextDeclRef`

- Default is boolean "OR", use `ShibRequireAll` for AND rule

- Regular expressions supported using special syntax:

```
require rule ~ exp
e.g. require mail ~ ^.*@(it|faculty).example.org$
```

Notes:



85

1. Example .htaccess File

- Require a user to be a staff member:

```
$ vim /var/www/html/staff-only/.htaccess
```

```
AuthType shibboleth
ShibRequestSetting requireSession 1
require unscoped-affiliation staff
```

Then access : `https://sp#.example.org/staff-only/`
with demouser/password. Access should be granted.

- Then try the same again with demostudent/password
  Access should be denied

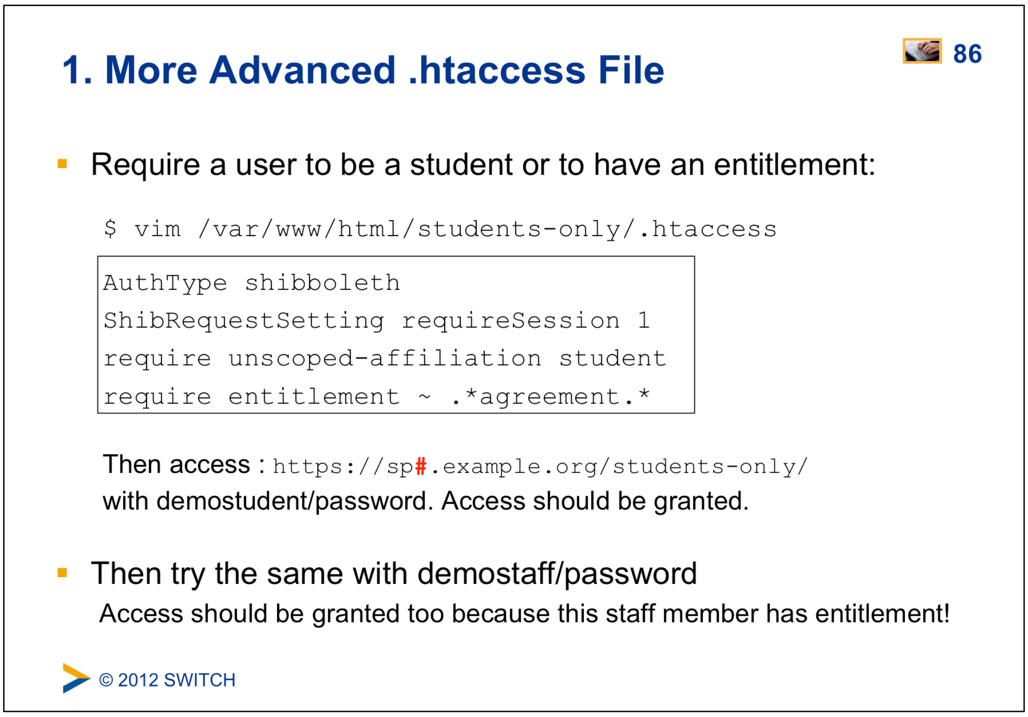
https://wiki.shibboleth.net/confluence/display/SHIB2/NativeSPApacheConfig

Notes: ____________________

____

____

____



86

1. More Advanced .htaccess File

- Require a user to be a student or to have an entitlement:

```
$ vim /var/www/html/students-only/.htaccess
```

```
AuthType shibboleth
ShibRequestSetting requireSession 1
require unscoped-affiliation student
require entitlement ~ .*agreement.*
```

Then access : `https://sp#.example.org/students-only/`
with demostudent/password. Access should be granted.

- Then try the same with demostaff/password
  Access should be granted too because this staff member has entitlement!

Notes: ____________________

____

____

____

Solutions for Access Control and Authorization

Excursion about using the Group Management Tool (GMT) or the SWITCHtoolbox.

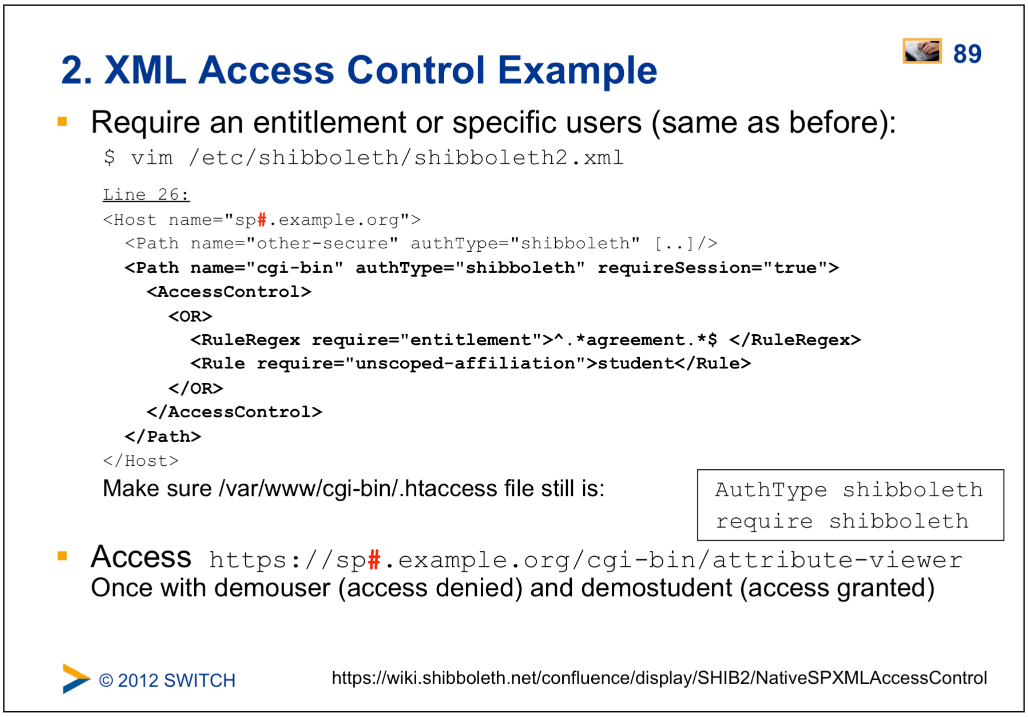
Please consult the table of contents to find this presentation in your hand-outs.

Notes:

2. XML Access Control

- Can be used for access control independent from web server and operating system
- XML Access control rules can be embedded inside RequestMap or be dynamically loaded from external file
- Boolean operators (AND,OR,NOT) can be used
- .htaccess files can reference to XMLAccessControl files Allows outsourcing access control rules to non-root users

Notes:



2. XML Access Control Example

- Require an entitlement or specific users (same as before):

```
$ vim /etc/shibboleth/shibboleth2.xml
```

```
Line 26:
<Host name="sp#.example.org">
  <Path name="other-secure" authType="shibboleth" [..]/>
  <Path name="cgi-bin" authType="shibboleth" requireSession="true">
    <AccessControl>
      <OR>
        <RuleRegex require="entitlement">^.*agreement.*$ </RuleRegex>
        <Rule require="unscoped-affiliation">student</Rule>
      </OR>
    </AccessControl>
  </Path>
</Host>
```

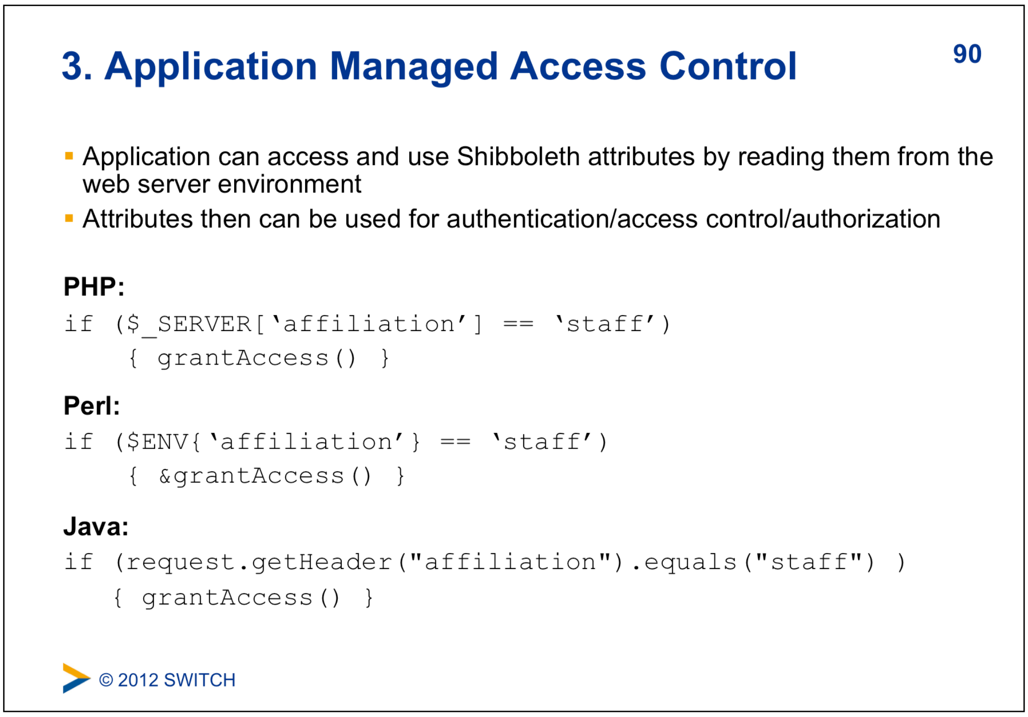
Make sure /var/www/cgi-bin/.htaccess file still is:

```
AuthType shibboleth
require shibboleth
```

- Access `https://sp#.example.org/cgi-bin/attribute-viewer`
  Once with demouser (access denied) and demostudent (access granted)

https://wiki.shibboleth.net/confluence/display/SHIB2/NativeSPXMLAccessControl

Notes:



3. Application Managed Access Control

- Application can access and use Shibboleth attributes by reading them from the web server environment
- Attributes then can be used for authentication/access control/authorization

PHP:

```
if ($_SERVER[‘affiliation’] == ‘staff’)
    { grantAccess() }
```

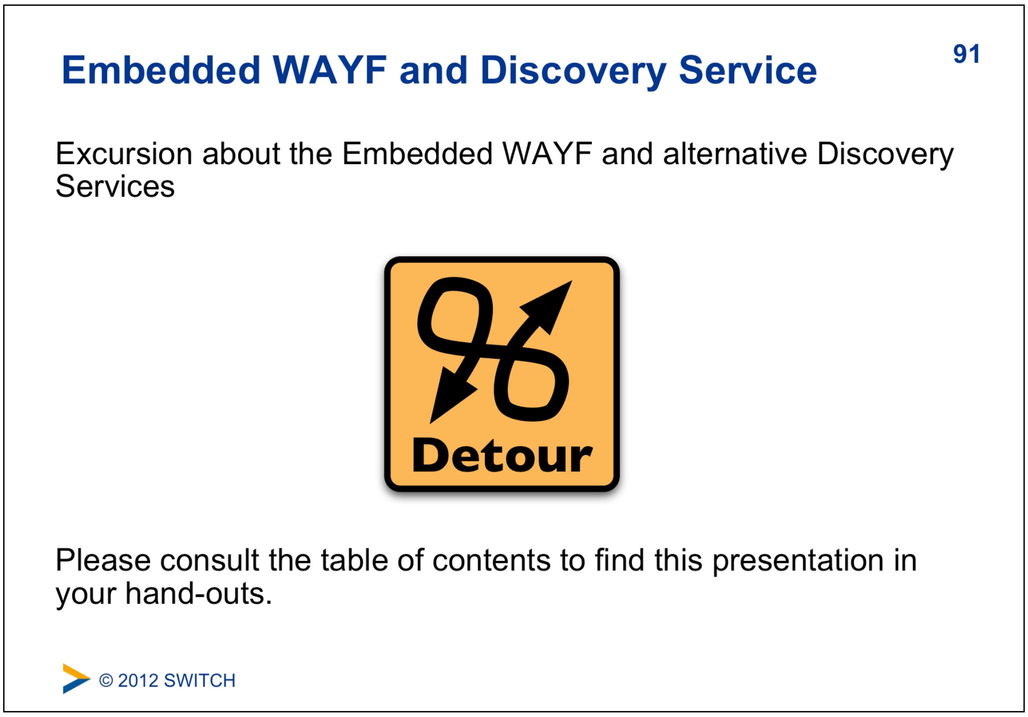
Perl:

```
if ($ENV{‘affiliation’} == ‘staff’)
    { &grantAccess() }
```

Java:

```
if (request.getHeader("affiliation").equals("staff") )
   { grantAccess() }
```

Notes:



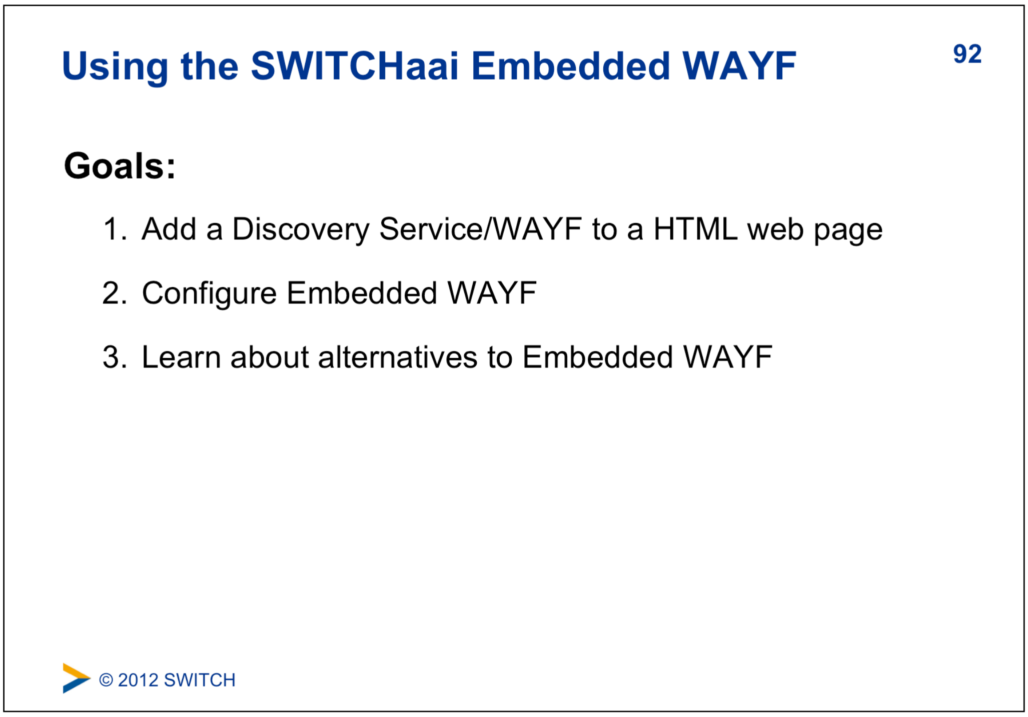
91

Embedded WAYF and Discovery Service

Excursion about the Embedded WAYF and alternative Discovery Services

Please consult the table of contents to find this presentation in your hand-outs.

Notes: ____________________



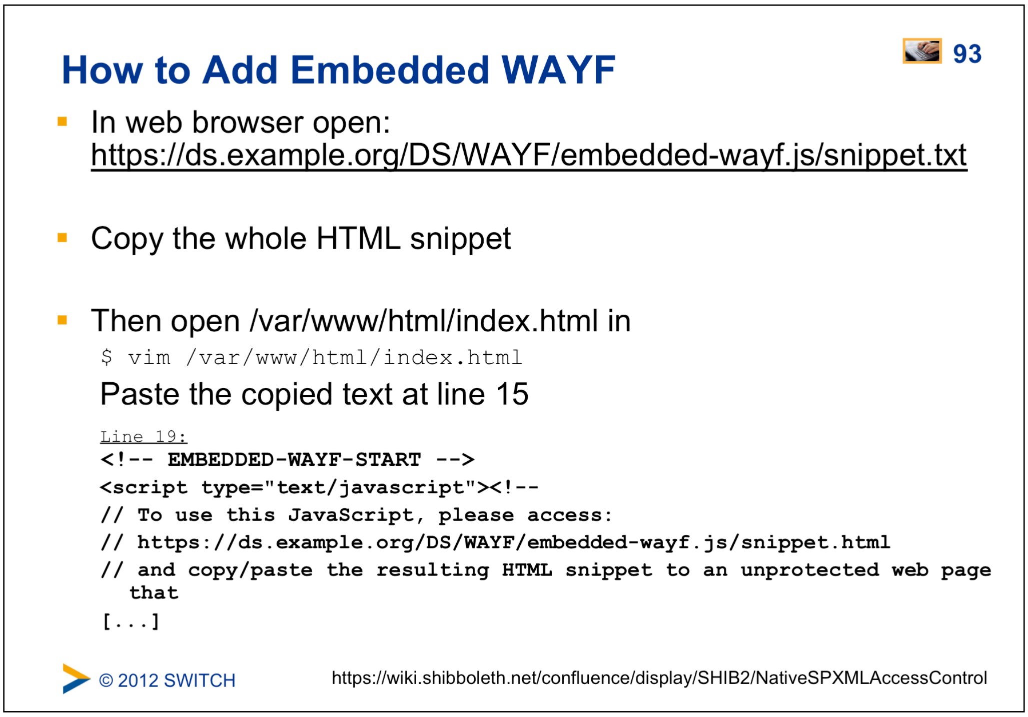
92

Using the SWITCHaai Embedded WAYF

Goals:

1. Add a Discovery Service/WAYF to a HTML web page
2. Configure Embedded WAYF
3. Learn about alternatives to Embedded WAYF

Notes: ____________________



93

How to Add Embedded WAYF

- In web browser open:
  https://ds.example.org/DS/WAYF/embedded-wayf.js/snippet.txt
- Copy the whole HTML snippet
- Then open /var/www/html/index.html in

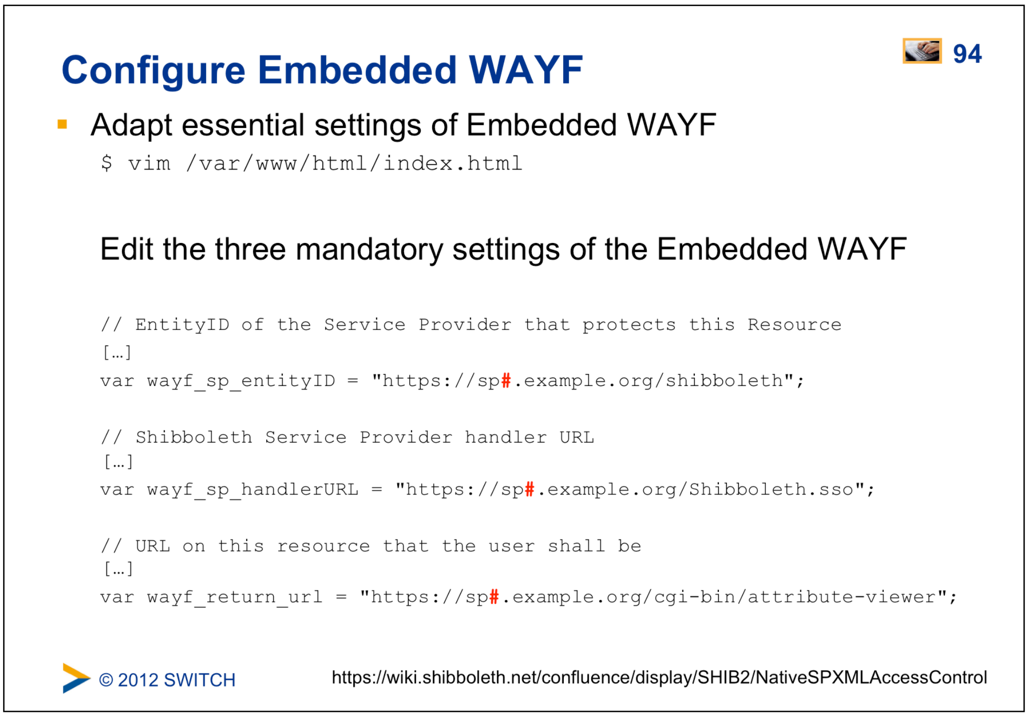
  ```
  $ vim /var/www/html/index.html
  ```

  Paste the copied text at line 15

  ```
  Line 19:
  <!-- EMBEDDED-WAYF-START -->
  <script type="text/javascript"><!--
  // To use this JavaScript, please access:
  // https://ds.example.org/DS/WAYF/embedded-wayf.js/snippet.html
  // and copy/paste the resulting HTML snippet to an unprotected web page
    that
  [...]
  ```

© 2012 SWITCH https://wiki.shibboleth.net/confluence/display/SHIB2/NativeSPXMLAccessControl

Notes:



94

Configure Embedded WAYF

- Adapt essential settings of Embedded WAYF

  ```
  $ vim /var/www/html/index.html
  ```

  Edit the three mandatory settings of the Embedded WAYF

  ```
  // EntityID of the Service Provider that protects this Resource
  […]
  var wayf_sp_entityID = "https://sp#.example.org/shibboleth";

  // Shibboleth Service Provider handler URL
  […]
  var wayf_sp_handlerURL = "https://sp#.example.org/Shibboleth.sso";

  // URL on this resource that the user shall be
  […]
  var wayf_return_url = "https://sp#.example.org/cgi-bin/attribute-viewer";
  ```

© 2012 SWITCH https://wiki.shibboleth.net/confluence/display/SHIB2/NativeSPXMLAccessControl

Notes:

Test the Embedded WAYF

- Access the URL https://sp#.example.org/
- Select the Test Identity Provider in the drop-down list
- Authenticate with demostudent/password
  You should see access to the attribute-viewer
- Go back to https://sp#.example.org/
  Note how the Embedded WAYF now looks different
- Change some of the Recommended Settings of the Embedded WAYF in /var/www/html/index.html for fun. E.g. color or size

https://wiki.shibboleth.net/confluence/display/SHIB2/NativeSPXMLAccessControl

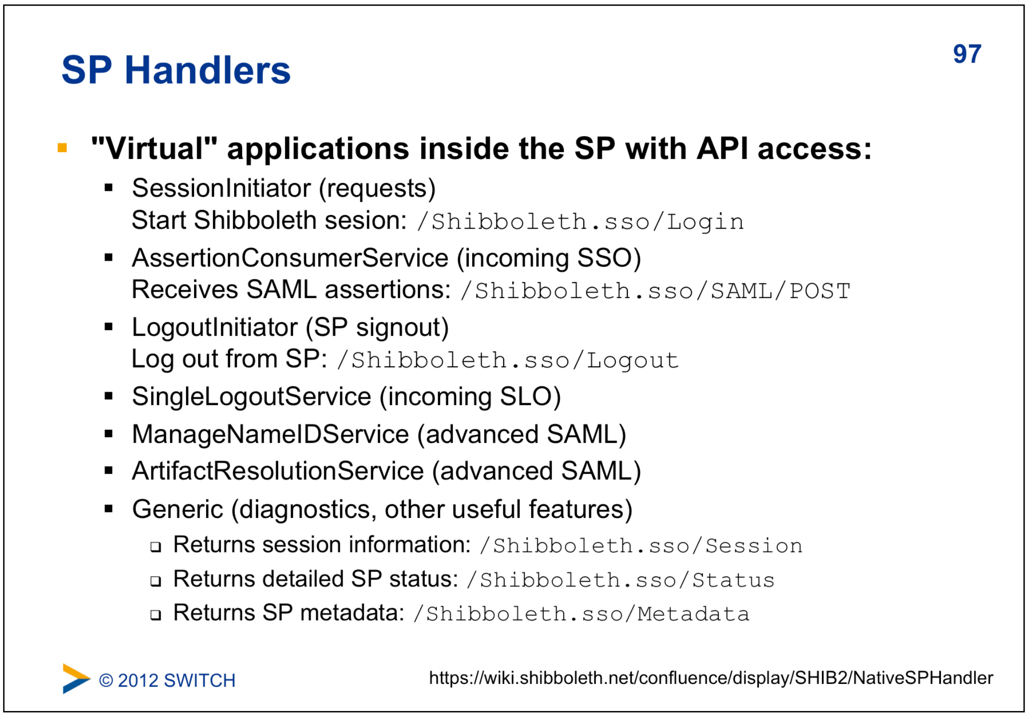
Notes:

Service Provider Handlers

Goals:

1. Understand the idea of a handler
2. Get an overview about the different types of handlers
3. Know how to configure them if necessary

Notes:



SP Handlers

- **"Virtual" applications inside the SP with API access:**
  - SessionInitiator (requests)
    Start Shibboleth sesion: `/Shibboleth.sso/Login`
  - AssertionConsumerService (incoming SSO)
    Receives SAML assertions: `/Shibboleth.sso/SAML/POST`
  - LogoutInitiator (SP signout)
    Log out from SP: `/Shibboleth.sso/Logout`
  - SingleLogoutService (incoming SLO)
  - ManageNameIDService (advanced SAML)
  - ArtifactResolutionService (advanced SAML)
  - Generic (diagnostics, other useful features)
    - Returns session information: `/Shibboleth.sso/Session`
    - Returns detailed SP status: `/Shibboleth.sso/Status`
    - Returns SP metadata: `/Shibboleth.sso/Metadata`

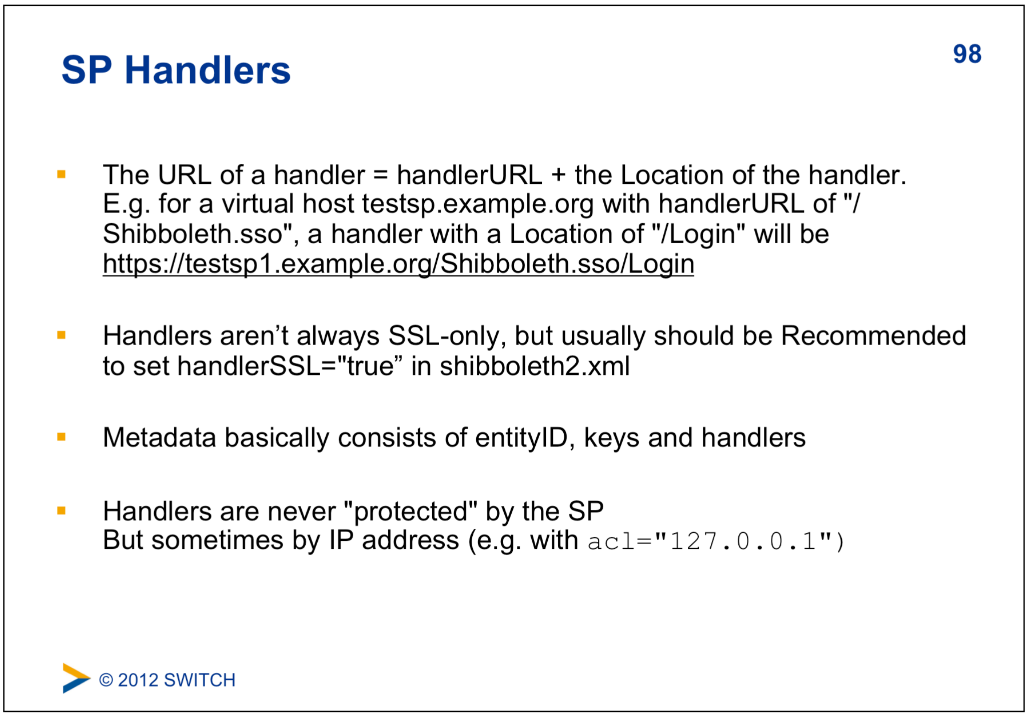
https://wiki.shibboleth.net/confluence/display/SHIB2/NativeSPHandler

Notes: ______________________________

____

____

____



SP Handlers

- The URL of a handler = handlerURL + the Location of the handler. E.g. for a virtual host testsp.example.org with handlerURL of "/Shibboleth.sso", a handler with a Location of "/Login" will be https://testsp1.example.org/Shibboleth.sso/Login

- Handlers aren’t always SSL-only, but usually should be Recommended to set handlerSSL="true” in shibboleth2.xml

- Metadata basically consists of entityID, keys and handlers

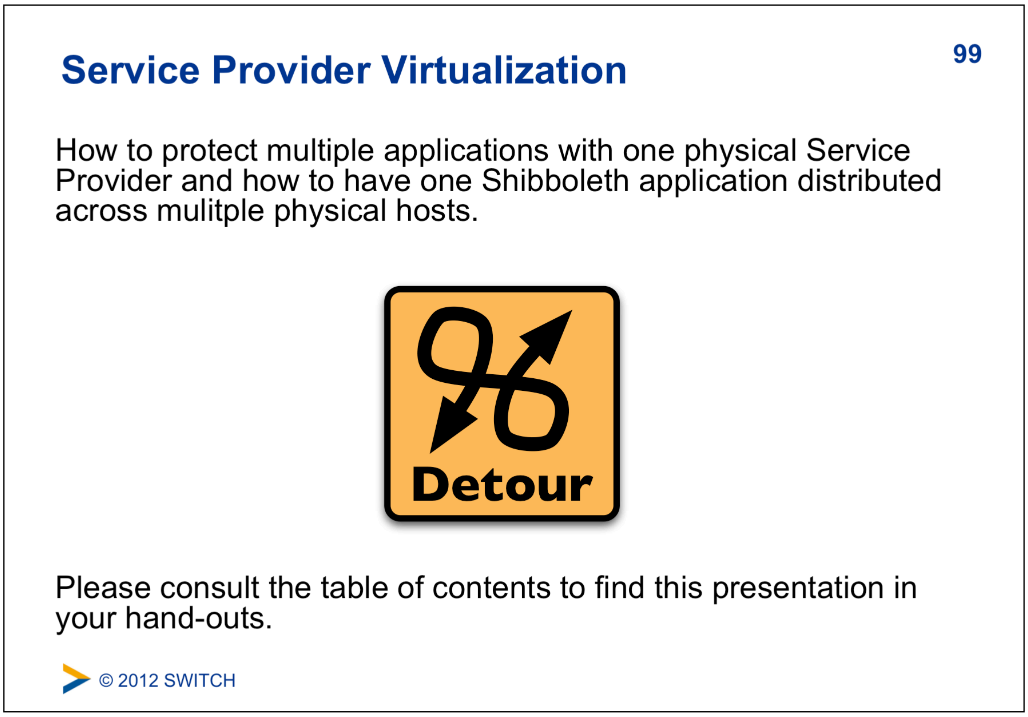
- Handlers are never "protected" by the SP
  But sometimes by IP address (e.g. with `acl="127.0.0.1"`)

Notes: ______________________________

____

____

____



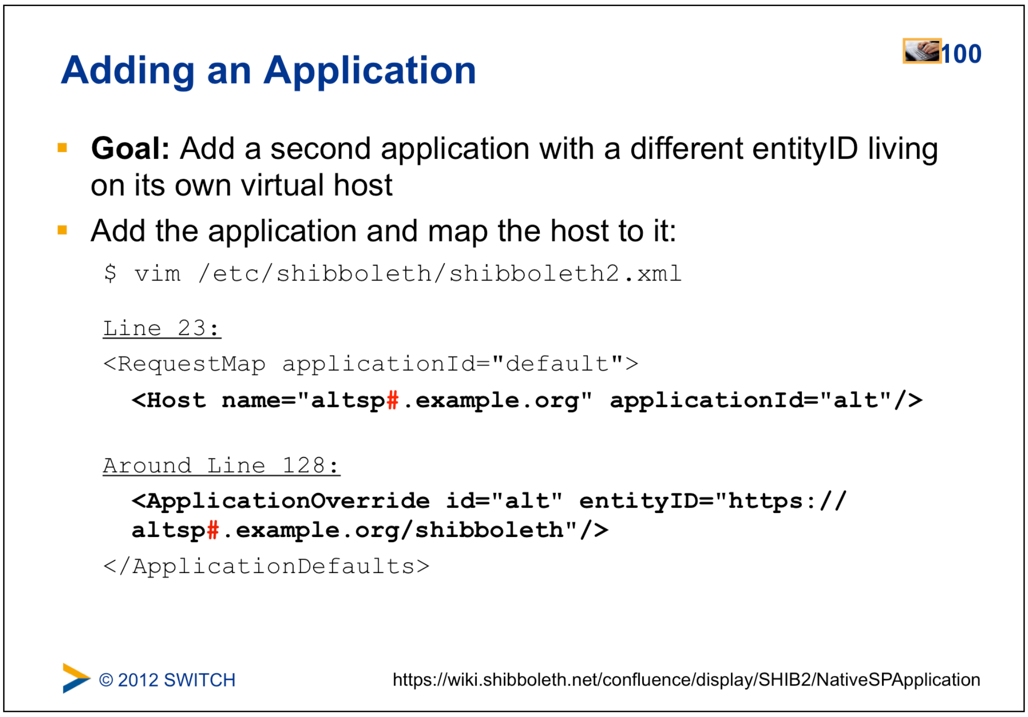
99

Service Provider Virtualization

How to protect multiple applications with one physical Service Provider and how to have one Shibboleth application distributed across mulitple physical hosts.

Please consult the table of contents to find this presentation in your hand-outs.

Notes: ______________________________



100

Adding an Application

- **Goal:** Add a second application with a different entityID living on its own virtual host
- Add the application and map the host to it:

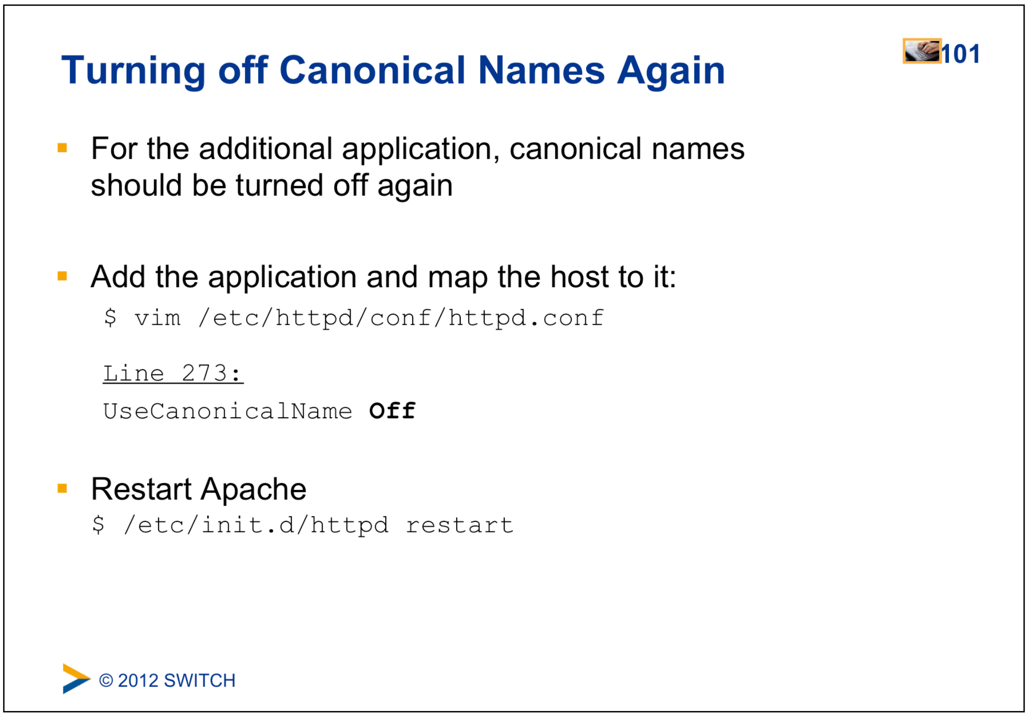
```
$ vim /etc/shibboleth/shibboleth2.xml

Line 23:
<RequestMap applicationId="default">
  <Host name="altsp#.example.org" applicationId="alt"/>

Around Line 128:
  <ApplicationOverride id="alt" entityID="https://
  altsp#.example.org/shibboleth"/>
</ApplicationDefaults>
```

https://wiki.shibboleth.net/confluence/display/SHIB2/NativeSPApplication

Notes: ______________________________



Turning off Canonical Names Again

- For the additional application, canonical names should be turned off again
- Add the application and map the host to it:

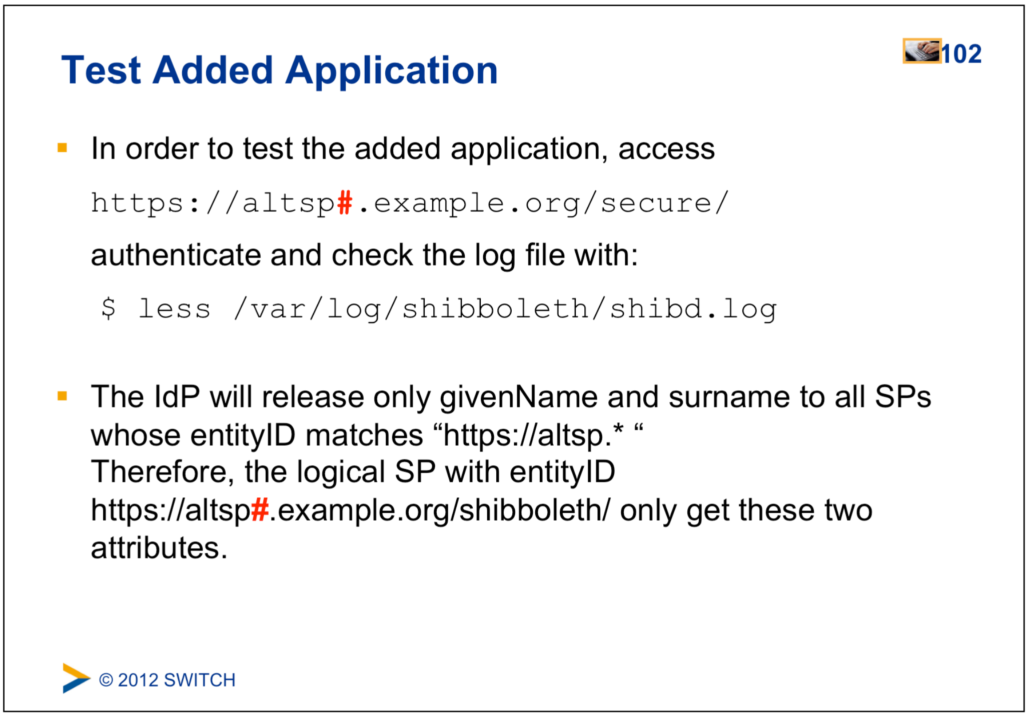
```
$ vim /etc/httpd/conf/httpd.conf

Line 273:
UseCanonicalName Off
```

- Restart Apache

```
$ /etc/init.d/httpd restart
```

Notes:



Test Added Application

- In order to test the added application, access

  `https://altsp#.example.org/secure/`

  authenticate and check the log file with:

```
$ less /var/log/shibboleth/shibd.log
```

- The IdP will release only givenName and surname to all SPs whose entityID matches “https://altsp.* “
  Therefore, the logical SP with entityID https://altsp#.example.org/shibboleth/ only get these two attributes.

Notes: