Vogler, Hartmut

An: Vogler, Hartmut
Betreff: Gedankenstütze zur OpenID Anbindung

ALLE Zugriff (bzw. Einsrünge aus z.B. Outlook) über "offizielle" URLs
auf Darwin laufen immer in die Bereiche ...
https://darwin.telekom.de/darwin/auth/
https://darwin.telekom.de/darwin/public/

Der Bereich /darwin/auth/ steht unter der mod_auth_ae Authorität
und wird über Standard-Basic-Auth authentifiziert.

Das Module mod_auth_openidc braucht seinen "eigenen" Bereich, der
dann über OpenID authentifiziert wird.

Dazu müssen in die httpd.conf die Zeilen ...

```
 Alias /OpenID/darwin/auth/     /opt/w5base/bin/
 Alias /OpenID/darwin/public/   /opt/w5base/bin/
 Alias /OpenID/darwin/static/   /opt/w5base/static/

 <Location /OpenID/darwin>
    Options -Indexes -FollowSymLinks +ExecCGI -Includes
    AuthType openid-connect
    SetHandler fcgid-script
    Require valid-user
 </Location>
```

... aufgenommen werden, um den Bereich /OpenID/ zu "erzeugen".

Die W5Base Code-Ausführungs-Rewrite Rule muss um den neuen
Bereich für /OpenID/ erweitert werden:

```
RewriteRule ^/(darwin)/(public|auth)/([^/]+)/([^/]+)/(.+)$ /$1/$2/fastapp.sh?MOD=$3::$4&FUNC=$5 [QSA,PT]
RewriteRule ^/(OpenID/darwin)/(public|auth)/([^/]+)/([^/]+)/(.+)$ /$1/$2/fastapp.sh?MOD=$3::$4&FUNC=$5
[QSA,PT]
```

Wir werden das ganze als MultiProvider Config auslegen, d.h.
die Config wird so aufgebaut, dass nicht nur ein Provider, sondern
beliebig viele eingebunden werden.

Dazu wird pro VirtualHost diese Grund-Config für openidc ...

```
 OIDCMetadataDir /etc/httpd/openidc

 OIDCRedirectURI https://darwin.telekom.de/OpenID/darwin/auth/redirect_uri
 OIDCDiscoverURL https://darwin.telekom.de/darwin/public/base/start/Main

 OIDCCryptoPassphrase XXXXXXXXXXXXXXXX
```

... definiert. Das OIDCDiscoverURL (base::start::Main) ist zum jetzigen
Zeitpunkt noch nicht 100% fertig - aber das ist auch noch nicht notwendig.

Das Verzeichnis /etc/httpd/openidc wird das Cache-Verzeichnis für die .provider
und .client files, die die MultiProvider Config für OpenIDC darstellen.
Nach aktuellem Stand sollte das Verezeichnis als ...
`install -d -o w5base -g apache -m 2770 /etc/httpd/openidc`
... ausgelegt werden.

Für jeden Provider muss in OIDCMetadataDir eine .provider und eine .client Datei hinterlegt sein. Die Provider-Datei muss am Beispiel von Google mit ...

```
wget -O accounts.google.com.provider https://accounts.google.com/.well-known/openid-configuration
```

... erzeugt werden. Der String vor .provider ist dabei der issuer (ist als Wert in der Datei vorhanden) ohne "http[s]://".

Die accounts.google.com.client Datei muss erstellt werden und muss folgendes beinhalten ...

```
{
  "client_id" : "xxx",
  "client_secret" : "xxx",
  "response_type" : "id_token",
  "scope" : "openid profile email"
}
```

Die client_id und das client_secret bekommt man, wenn die Applikation beim betreffenden Provider definiert wird. Bei Yahoo und Google kann dies jeder Entwickler über die Plattformspezifischen Tools.
Bei Google z.B. über die DeveloperConsole:
https://console.cloud.google.com/apis/dashboard

Bei EntraID kann das nur ein Company-Admin im AD und muss von irgendwelchen kollegierenden zetral gebaut werden.

Alle Dateien in /etc/httpd/openidc sind Site-Spezifisch und müssen bei einer Neuinstallation erhalten werden. Ich bin mir nicht sicher, ob man da auf einen definierten Initial-Stand aufsetzen kann - das müssten wir ausprobieren. Alle Dateien darin (vor allem die .client Files) sind Security-Relevant und müssen "geschützt" behandelt werden - ähnlich wie private-Host-Keys.

Um die Buttons auf der Darwin-Login Seite (Modul base::start) konfigurieren zu können, muss nun in die Config die Zeilen ...

```
LOGINNAME[00default]="W5Base/Darwin Login"
LOGINICON[00default]=https://darwin.telekom.de/darwin/public/base/load/DefaultLogin.gif
LOGINHANDLER[00default]="../../../auth/base/menu/root"

LOGINNAME[01tsso]="Telekom T-SSO Login"
LOGINICON[01tsso]=https://darwin.telekom.de/darwin/public/base/load/DefaultLogin.gif
LOGINHANDLER[01tsso]=https://SSO:SSO@darwin.telekom.de/darwin/auth/base/menu/root

LOGINNAME[02tsso]="Telekom AD-Login (Kerberos)"
LOGINICON[02tsso]=https://w5base-devnull-ng.telekom.de/darwin/public/base/load/DefaultLogin.gif
LOGINHANDLER[02tsso]=https://ADS:ADS@darwin.telekom.de/darwin/auth/base/menu/root

LOGINNAME[03google]="GoogleCloud"
LOGINICON[03google]="/darwin/public/base/load/DefaultLogin.gif"
LOGINHANDLER[03google]=https://OpenID:OpenID@darwin.telekom.de/darwin/auth/base/menu/root?https://accounts.google.com
LOGINOPENIDC[03google]="/etc/httpd/OpenID-Cache/accounts.google.com.provider"

LOGINNAME[04entraid]="EntraID"
LOGINICON[04entraid]="/darwin/public/base/load/DefaultLogin.gif"
LOGINHANDLER[04entraid]=https://OpenID:OpenID@darwin.telekom.de/darwin/auth/base/menu/root?https://entra.msrotz.com

LOGINNAME[05yahoo]="Yahoo"
LOGINICON[05yahoo]="/darwin/public/base/load/DefaultLogin.gif"
LOGINHANDLER[05yahoo]=https://OpenID:OpenID@darwin.telekom.de/darwin/auth/base/menu/root?https://api.login.yahoo.com
LOGINOPENIDC[05yahoo]="/etc/httpd/OpenID-Cache/api.login.yahoo.com.provider"

LOGINNAME[06openid]="Generic OpenID"
LOGINICON[06openid]="/darwin/public/base/load/DefaultLogin.gif"
LOGINHANDLER[06openid]=https://OpenID:OpenID@darwin.telekom.de/darwin/auth/base/menu/root
LOGINOPENIDC[06openid]="/etc/httpd/OpenID-Cache"
```

... hinzugefügt werden. Der letzte Punkt "Generic OpenID" funktioniert aktuell noch
nicht, da /darwin/public/base/start/Main noch nicht als "echter" Discoverer fungieren kann (es müsste dazu /etc/httpd/openidc als Verzeichnis gelesen und ausgewertet werden).

In den LOGINHANDLER Zeilen ist erkennbar, dass aus dem /darwin/auth Bereich dann auf /OpenID/darwin gewechselt wird, wenn dort als Username "OpenID" verwendet - Passwort entweder Leer oder ebenfalls "OpenID".
Diese Switch Funktion entsteht durch das angeben von OpenID in der ModAuthAE Config mit der Zeile ...
aeSSOBasicAuthUser sso SSO ciam CIAM ad AD ads ADS oid OpenID
Damit wird dann "OpenID" ein User, der generell funktioniert und ohne Passwort aktzeptiert wird.

Damit anhand der in aeSSOBasicAuthUser angegebenen "Dummy"-User das Switchen auf die unterschiedlichen Auth-Systeme funktioniert, sind die Rewrite Regeln ...

```
 RewriteCond %{REQUEST_URI} !^/OpenID/
 RewriteCond %{LA-U:REMOTE_USER} ^(OID|oid|OpenID)$
 RewriteRule "^/(.*)" https://darwin.telekom.de/OpenID/$1 [R,NE,END]

 ### Classic T-SSO  Begin ######
 RewriteCond %{REQUEST_URI} !^/OpenID/
 RewriteCond %{LA-U:REMOTE_USER} ^(sso|ciam|SSO|CIAM)$
 RewriteRule "^/(.*)" https://myportal-websso.corp.telekom.de/darwin.telekom.de/$1 [R,NE,END]

 ### Kerberos T-SSO (EMEA Login) Begin ######
 RewriteCond %{REQUEST_URI} !^/OpenID/
 RewriteCond %{LA-U:REMOTE_USER} ^(ad|AD|ads|ADS)$
 RewriteCond "%{QUERY_STRING}" ^(.*)$
 RewriteRule "^/(.*)" https://myportal-
websso.corp.telekom.de/login/direct/kerb?target=https://myportal-
websso.corp.telekom.de/darwin.telekom.de/$1?%1 [QSD,B,NE,R,END]
```

... notwendig. Wichtig ist hier die Ausschließende Condition !^/OpenID/ am Anfang, da ansonden durch den LA-U (lookup) check auf den Usernamen eine Schleife entstehen würde.
Die letzten beiden Regeln steuern das Switchen auf T-SSO bzw. T-SSO Kerberos Login. Dies wird mittelfristig entfallen bzw. nicht mehr funktionieren.