Administrator's Guide to Codename Circumference

December 2010 — Circumference 1.5

Attention users reading the `text/plain` export of this document: this document has been generated from `circum_adminguide.lyx`. A full-quality PDF version is available as `adminguide.pdf` in the documentation tarball, on the website, or in the "doc" branch of the source repository, to reduce the size of the source tarball!

Circumference[1] is an implementation of the Diameter AAA (Authentication, Authorization and Accounting) protocol with focus on the Diameter WebAuth Application.

Contents

[1] The name is a pun on the Diameter protocol. Diameter itself "is a pun on the RADIUS protocol, which is the predecessor (a diameter is twice the radius)"**(author?)** [WP], and so is circumference, which is about 3.14× the diameter.

1 What is included

- Diameter base library, libcircum
  - libcircumplus, a C++ wrapper
- Diameter server, circumd
  - WebAuth module for circumd
- Apache httpd 2.x modules that implement WebAuth clients
  - auth_circum for Digest authentication
  - authn_circum for Basic authentication (using httpd's own auth_basic as a caller)
- Documentation
  - Administrator's Guide (full PDF on website, the documentation tarball, or the "doc" branch in the repository)
  - Developer's Guide (website/doc tarball/repo)
  - auto-generatable documentation from source using doxygen
  - Manual pages for shell commands

2 Installation

It is recommended that you install Circumference by picking a suitable RPM/DEB or other appropriate package with precompiled binaries. There are however times when this is not possible, for example when there is no such package. In this case, we suggest to download a source tarball from the homepage.

2.1 Operating System support

The software has been tested on contemporary Linux 2.6 distributions, Solaris 10 and OpenBSD 4.3.

2.2 From repository checkout

Repository checkouts are usually only useful to those who know what to do with it — doing development on the project, walking its history, or other forms that specifically require a repository. The source code is managed using the Git version control system created by Linus Torvalds. You will need to have it installed obviously; the package is called git or git-core, depending on distribution. It is recommended to use at least Git 1.6.x. To download the repository, issue

```
git clone git://circum.git.sf.net/gitroot/circum/circum
```

As a basic rule, repositories should not carry any files that can be auto-generated, and so does ours. It specifically does not come with a generated `configure` script. To do so, make sure you have the following installed:

- autoconf 2.59 or up
- automake 1.10.2 or up
- libtool 2.2.6 or up
- pkg-config 0.19 or about

Older versions might work, but we did not test these specifically, just whatever our Linux distributions provided. After these four components, of which the first three are collectively referred to as autotools, you will be able to run `./autogen.sh`, which will build the `configure` script. From here on, continue with the next section.

2.3 From tarball

After extraction of the tarball, or by having generated `configure` yourself, you can run the latter. Codename Circumference has a number of prerequisites for it to be built:

- Always required:
  - GNU C Compiler (preferably 4.x, but 3.3.5 also works)
  - libHX 3.13 or up
  - libpcre 7.6 or about (7.0 is set as a limit)
  - libxml 2.6 or up
  - openssl 0.9.7 or up
  - pkg-config 0.19 or about
- To optionally build the Apache httpd2 modules for WebAuth:
  - apache2-devel 2.0 or up
- To optionally build libcircumplus:
  - g++ 4.3 or up
- To optionally build WebAuth Basic Authentication with support for using PAM for authentication:
  - pam 0.99.10 or about
- To optionally generate documentation from the source:
  - doxygen 1.5.5 or about, graphviz-gd

It is assumed that you at least have made brief contact with autotools before; in essence, the only thing that is required now to build is, is to run `./configure`, optionally with some extra options that control where the package will be installed or how it is compiled.

2.4 Paths

Paths are set at "configure-time", and because they might get compiled into the binaries, cleaning the build by use of `make clean` might be necessary after reconfiguring with new paths. Even if you do not compile the package yourself but use a provided precompiled package, you should know the locations of the files, as we will be referring to it in this manual later on. Consult your package manager manual on how to list the files for a given package. You can run the "`minitest`" program to dump the path configuration.

The default installation "prefix" set by autotools is `/usr/local`, this in itself is just an indicator for a base path, and otherwise only serves as a variable. The directories where files actually get copied to are various. Directories default to variables that are dependent upon another, such as "bindir" being defined to "`${`*prefix*`}/bin`" by default. This string is expanded at configure time, and allows to specify `--prefix=/opt/circum` and have bindir at the location we would expect it to be, `/opt/circum/bin`. To explicitly override, you use `--bindir=$HOME/bin`, for example. Important locations are:

prefix Defaults to `/usr/local`. The configure flag for it is `--prefix`.
Packages for distributions will use `--prefix=/usr`, because it is not a local install in that case. If you build from source, leave it at `/usr/local`, or use `/opt/circum`. This makes it easiest to find source-based installed programs at a later date. If you want to install it to your home directory, use something like `--prefix=$HOME`. (We usually prefer `--prefix=$HOME/myroot` to not clutter our `~`.)

bindir Default: `${`*prefix*`}/bin`. Flag: `--bindir`.
All user-executable programs will be put here.

sbindir Default: `${`*prefix*`}/sbin`. File: `--sbindir`.
All programs intended only for the superuser will be put here.

sysconfdir Default: `${`*prefix*`}/etc`. Flag: `--sysconfdir`.

datadir Default: `${`*prefix*`}/share`. Flag: `--datadir`.

mandir Default: `${`*datadir*`}/man`. Flag: `--mandir`.
This is for manual pages.

pkgsysconfdir Default: `${`*sysconfdir*`}/circum`. Flag: `--with-pkgsysconfdir`.
This is the place for configuration files.

pkgdatadir Default: `${`*datadir*`}/circum`. Flag: `--with-pkgdatadir`.
Data files such as AVP/ID XML dictionaries go in here.

pkglibdir Default: `${`*libdir*`}/circum`. Flag: `--with-pkglibdir`.
This variable/flag is currently unused.

pkglibexecdir Default: `${`*libexecdir*`}/circum`. Flag: `--with-pkglibexecdir`.
This directory is for modules to the circumd Diameter Server.

pkgconfigdir Default: `${`*libdir*`}/pkgconfig`. Flag: `--with-pkgconfigdir`.
Files for the "pkg-config" development helper program go here. The location is mandated by pkg-config, but you can specify an alternate path if you know how to tune pkg-config to look into non-standard directories too. (This will be needed for third-party developments that build upon the Circumference libraries.)

apachelibdir Defaults to what the apxs build helper returns as the location for Apache httpd2 modules. This is for example `/usr/lib/apache2/modules`. The `configure` flag is `--with-apache=/your/directory`.

These are a lot of directories, but they all have their purpose. If in doubt, do not change any defaults except perhaps the prefix, where it is customary to either stick with `/usr/local` or `/opt/circum` (in case of Codename Circumference)**(author?)** [FHSopt], or your somewhere in your home directory if you cannot do a system-wide-visible install.

It is imperative that you specify the correct libdir when building from source! Especially in 64-bit bi-arch environments such as AMD64 and sparc64 (but not, for example, IA-64), you have to explicitly specify lib64 using something like `--libdir='${`*prefix*`}/lib64'`, because autotools does not figure that one out on itself. Failure to obey may result in overwriting previously existing 32-bit libraries with 64-bit libraries by the same name, and thus making programs dependent on those 32-bit libraries cease to work.

2.5 Building 32-bit with a 64-bit bi-arch compiler

Just tell the compiler and linker to use 32-bit mode:

```
./configure CFLAGS="-m32" CXXFLAGS="-m32" LDFLAGS="-m32"
```

(But **do not** specify lib64 if compiling 32-bit libraries!) This requires that the 32-bit parts of GCC are installed, which may not always be the case. Ensure that you have a package named like "`gcc-32bit`" installed, and/or that you can call '`gcc -m32 -xc /dev/null`' without getting a link-time error that it cannot find a suitable (32-bit) libgcc.

```
/usr/lib64/gcc/x86_64-suse-linux/4.3/../../../../x86_64-suse-linux/bin/ld:
skipping incompatible /usr/lib64/gcc/x86_64-suse-linux/4.3/libgcc.a when
searching for -lgcc
/usr/lib64/gcc/x86_64-suse-linux/4.3/../../../../x86_64-suse-linux/bin/ld:
cannot find -lgcc
collect2: ld returned 1 exit status
```

If you get an error about the absence of "`main`" instead, everything is set up already.

2.6 Apache build tool

If the Apache build tool apxs is not reachable through `$PATH` as either "`apxs2`" or "`apxs`", you can specify an exact location using `--with-apxs`, e.g.

```
./configure --with-apxs=/opt/apache2/bin/apxs
```

`configure` checks for `apxs2` first, because that is surely guaranteed to be the httpd 2.x version, while `apxs` may refer to any version of httpd. OpenBSD's `apxs` for example is the 1.x version, since this operating system ships both httpd 1.x and 2.x, but when installing httpd2 from source, you will also get a binary that is just called `apxs`.

2.7 Run make

After having run `./configure` with all the flags and options that you desire, run `make` to build the software. When this is done too, you can run `make install` to copy it to the designated target location set at configure time.

To extract the in-code API documentation, you must explicitly run `doxygen` (no arguments required, though) in the root of the software package distribution, which will write the files to `doc/html/`.

2.8 For developers

2.8.1 Debugging

To enable debugging, you will need to pass `-O0` and `-ggdb3` to `CFLAGS` and `CXXFLAGS`.

```
./configure CFLAGS="-O0 -ggdb3" CXXFLAGS="-O0 -ggdb3"
```

"`-O0`" is used to explicitly disable any compiler optimizations. This is to make sure the compiler does not reorder instructions, so that stepping through the code with `gdb` does not jump around lines on every step because of such optimizations. "`-ggdb3`" is to produce debug information to the fullest.

2.8.2 Run from working directory

Since programs are compiled with the configured paths, they rely on having been installed to that prefix area. If however, you want to directly run the programs from the extracted tarball, and doing so without running '`make install`' either — perhaps you are a developer and want to save time —, you just call `configure` with the paths pointing to your very source directory so that programs can find their data files:

```
./configure --with-pkgdatadir=$PWD/share \
            --with-pkglibexecdir=$PWD/circumd/.libs \
            --with-pkgsysconfdir=$PWD
```

`$PWD` is used to ensure an absolute path is passed to `configure`. This also means that moving the source directory requires a reconfigure. Using relative paths is at your own risk. There is a shortcut program called `configure-developer` in the source tree root to do this long line. It also includes the debugging flags.

To actually use gdb on a program inside the working directory, you need to call `libtool` instead:

```
libtool --mode=execute gdb circumd/circumd
```

2.9 Building on Solaris

Make sure you have all the software requirements. If not provided by your Solaris installation, you can turn to `http://blastwave.org/` and install add-on software from there, which will be installed to `/opt/csw`. Most likely, you need extra environment settings, such as

```
export PATH="$PATH:/opt/csw/bin:/opt/csw/gcc4/bin:/usr/ccs/bin";
export PKG_CONFIG_PATH="/usr/lib/pkgconfig:/opt/csw/lib/pkgconfig";
export LD_LIBRARY_PATH="/opt/csw/lib";
```

And add

```
CFLAGS="-I/opt/csw/include" CXXFLAGS="-I/opt/csw/include"
LDFLAGS="-L/opt/csw/lib"
```

when calling `configure`. You might need extra flags if certain dependencies are not found; this however is beyond the scope of this document. Circumference was compile-tested with Solaris 10's make and linker (`/usr/ccs/bin/ld`).

2.10 Building on OpenBSD

Additional software can be downloaded via the OpenBSD ports system. It may be necessary that you pass

```
libcrypto_CFLAGS=" " libcrypto_LIBS="-lcrypto"
libssl_CFLAGS=" " libssl_LIBS="-lssl"
```

to `configure` as arguments, because OpenBSD does not ship any pkg-config files for OpenSSL. The whitespace in `libcrypto_CFLAGS` and `libssl_CFLAGS` is required, as an empty variable is functionally the same as "could not find libcrypto".

3 Base library

3.1 AVP Definition Dictionaries

If you want to make additional AVP Definition Dictionaries available to libcircum, place them into `${`*pkgdatadir*`}` (see section 2.4). Dictionaries are read in unconditionally and in no particular order. libcircum will warn on stdout/stderr of duplicate definitions or other problems with dictionary files. They must be in XML format and adhere to `dictionary.dtd`, which is also available in that directory for reference.

4 Diameter server

4.1 Configuration

The circumd Diameter server uses an XML configuration file. Currently, it merely serves to list the ports on which the server shall listen:

```
<?xml version="1.0" encoding="utf-8" ?>
<circumd>
        <socket type="ipv6" proto="tcp" port="3868"
                mandatory="1" tls="false" />
</circumd>
```

`<socket>` defines a socket to listen on, by specifying the layer-3 and layer-4 protocol as well as a port and whether or not its availability is mandatory to successfully start the server. Multiple sockets may be set up. It accepts the values `ipv4` and `ipv6` for the "type" field, and `tcp` and `sctp` for the "proto" field. The port number must be within the acceptable range of the layer-4 protocol.

The "mandatory" attribute may be `0` or `1`, and specifies whether failure of creating the socket, either due to the operating system not supporting the particular layer-3 or layer-4 protocol, or the port already being in use, will terminate the server. "tls" signifies whether the socket expects TLS connections.

Note: In Linux kernels before 2.6.27(-rc1), IPv6 SCTP sockets were always implicitly bound for IPv4 too, so you might see an error when creating SCTP sockets. See `http://marc.info/?l=linux-netdev&m=121123385730736&w=2` for details.

The default port for the Diameter protocol as assigned by IANA is 3868. You may see that other implementations bind to port 1812 instead (or in addition), but this port is reserved for the RADIUS protocol.

4.2 Starting the server

`circumd` can be started without any arguments and will try to read the configuration file from `${`*pkgsysconfdir*`}/circumd.conf.xml`. The `-c` option may be used to specify an alternate path.

When the software is installed, either through a distribution package or `make install`, a runlevel start script is available in `${`*sysconfdir*`}/init.d/circumd`, which will take care of starting server in the background using `startproc`, as `circumd` always starts in foreground mode.

4.3 Modules

The server will unconditionally try to load all modules ending in the "`.so`" extension from the `${`*pkglibexecdir*`}` directory. There is currently no option to selectively enable or disable modules other than moving them into or out of the directory, respectively.

5 WebAuth server module

5.1 Configuration

A few parameters of the WebAuth Server are configurable through an Apache-style config file in the format of

```
ItemName  Value
```

The location of the file is `${`*pkgsysconfdir*`}/webauth_config.txt`. Empty lines and lines starting with a hash mark (`#`) are ignored. Possible configuration item names:

AcceptAllNonces If set to `true`, the WebAuth Server will allow any nonce received, provided that the response hash correlates with the nonce. This can be useful to reduce the amount of traffic, since clients may generate their own nonces, and thus save one round trip. When enabled, the WebAuth implementation will disable the nonce lifetime checks, subsequently allowing replay attacks. Defaults to `false`.

AuthorizationFile Path to the file containing the rules for authorization. See section 5.5 for details.

BasicUserFile Path to the file containing the list of (`user:realm:password`) tuples usable for Basic authentication. See section 5.2 for details.

DigestUserFile Path to the file containing the list of (`user:realm:password`) tuples usable for Digest authentication. See section 5.3 for details.

SecretLength The length of the internal secret, in bytes. The default is 32.

SendHA1 If set to `true`, the module will send the user's HA1 hash together with the nonce. This way, an authenticating instance (like Apache) may provide authentication and authorization without further asking the Diameter server. Not recommended since the HA1 value is nearly as useful as the plaintext password. Especially sending the HA1 value of a requested user to any requesting client would be a ridiculous security hole. Defaults to `false`.

Timeout Lifetime of a nonce in seconds. Defaults to 60.

5.2 Basic authentication

A simple flat text file database is used for Basic authentication. Every line is of the form

```
user:realm:password
```

The password is stored unencrypted, so appropriate permissions need to be set. Only the Diameter server with the WebAuth module needs to be able to read the file. If authentication against PAM is not configured, or PAM does not know a particular user, or returns an error, authentication against the plaintext file will be attempted.

The location of the file defaults to `${`*pkgsysconfdir*`}/webauth_users.txt`, but can be configured through the WebAuth Server configuration file (see section 5.1).

5.3 Digest authentication

Digest authentication also uses a flat text file database, but is a three-column format:

```
user:realm:hash
```

The hash value can be computed for the specific (user, realm) tuple using the `genuserhash`(1) program shipped with Circumference. Empty realms are permitted. Only the Diameter server/ WebAuth module needs to read this file, and it is advised to set the permissions in a strict way, just like you would with `/etc/shadow`.

The location of the file defaults to `${`*pkgsysconfdir*`}/webauth_users_digest.txt`, but can be changed through the WebAuth Server configuration file. Sample entry from the shipped Digest file:

```
linux:reg_user@realm:08d709649f949cc989e97db40e8cdf44
```

5.4 Basic authentication with PAM backend

If you want to use PAM for Basic authentication, Make sure that pam-devel is selected at configure-time using `--enable-pam` and installed during compilation so that the WebAuth module can get linked against it. Create a file `/etc/pam.d/circumd_webauth` if it does not exist yet and put into it whatever modules you want to authenticate against:

```
#%PAM-1.0
auth sufficient pam_ldap.so
auth sufficient pam_krb5.so
auth required pam_unix2.so
```

Note that circumd must be run with appropriately elevated privileges if you want to successfully authenticate against pam_unix2, since that module tries to read `/etc/shadow`. The module names might vary. Redhat's distributions for example are still stuck with the old pam_unix. Alternatively, you can use the following (name of "`common-auth`" might vary again):

```
#%PAM-1.0
@include common-auth
```

or (what seems to be an older syntax)

```
auth include common-auth
```

to use the common methods set up by the distribution/admin.

5.5 Authorization

The WebAuth Server can authorize against service identifiers received in the AVP, for example username and URI. Authorization rules are listed in the text file `${`*pkgsysconfdir*`}/webauth_authz.txt` by default, but this also can be changed through the WebAuth Server configuration file. The authz rules file contains lines that specify the list of usernames a given rule applies, and the rule itself. These lines must always come in pairs, like this:

```
# Everyone else only gets access to /~shi
<unknown>
(WebAuth-URI=^/~shi.*$)
linux
(!(WebAuth-URI=^/bsd\b)
```

Comment lines starting with a hash mark (`#`) are completely ignored and do not affect the state of the authz file reader. The first line in a pair specifies the user list, consisting of usernames separated by comma. "`<unknown>`" is the default entry that is applied when the username that was used to log into the website is not otherwise present in the authz file. The second line specifies rules that have to apply for access to be granted, using an arbitrarily-nestable infix-based expression syntax, with parenthesis required similar to the prefix-based OpenLDAP search filters.

A terminal expression is of the form "`(key=regex)`" or "`!(key=regex)`" (for inversion) and does the actual match on a service identifier. Possible key names are listed in table **??**. The regex shall be conform to PCRE (Perl-compatible regular expression) as documented in the `pcre`(3) manual page.

List of possible service identifiers supported by the WebAuth Server:

WebAuth-URI The URI that was accessed. This is for example "`/index.html`".

WebAuth-Service Name or identifier for the Diameter client to authorize for. (unused)

Remote-User Identification of the remote user requesting authorization.

User-Name The username that the user agent (e.g. web browser) used to log in.

Remote-Address The IP address or hostname of the user agent.

Non-terminal expressions are of the form "`(L*R)`" or "`!(L*R)`" where L and R are subexpressions (either terminal or non-terminal), and "*" denotes an operator out of the allowed set: "`&`" (ampersand) for a conjunction (binary AND) or "`|`" (pipe symbol) for a disjunction (binary OR). The EBNF syntax is available in figure 1.

```
    rule  ::=  nexpr | expr
   nexpr  ::=  [ "!" ] "(" rule operator rule ")"
    expr  ::=  [ "!" ] "(" key "=" regex ")"
operator  ::=  "&" | "|"
```

Figure 1: EBNF for the rule syntax

6 WebAuth Apache httpd2 modules

The Apache 2 WebAuth modules mod_auth_circum and mod_authn_circum are configured through the Apache configuration files, either through the system-wide ones in or about `/etc/apache2`, or the per-directory `.htaccess` files, if enabled in the Apache httpd. As usual, the format of the lines in the configuration entries are:

```
ItemName  Value
```

Authorization with the httpd2 modules is not supported at this point.

6.1 Basic Authentication with mod_authn_circum

The mod_authn_circum module will use mod_auth_basic and register as a "provider" for Basic authentication. Most configuration keys belong to auth_basic, and you are encouraged to look into the Apache httpd2 manual for description of their purpose. Only one is defined in authn_-circum:

CircumServerName Hostname, IP address or URI; any form libcircum accepts.

```
<Location />
        AuthType Basic
        AuthName Secret
        Require valid-user
        AuthBasicProvider circum
        DiameterServerName localhost
</Location>
```

6.2 Digest Authentication with mod_auth_circum

The configuration items that are made available by the modules are:

AuthName The authentication realm (e.g. "Members only"). You cannot use colons (`:`) in it, because they are used in the WebAuth server's Digest user database as delimiters.

AuthDigestNonceLifetime Maximum lifetime of the server nonce, in seconds.

DiameterServerName Hostname, IP address or URI; any form libcircum accepts.

A configuration entry would therefore look like:

```
<Location />
        AuthType CircumDiameterDigest
        AuthName reg_user@realm
        Require valid-user
        DiameterServerName localhost
</Location>
```

This example applies to the web root directory and all directories below it. "`CircumDiameter-Digest`" is the magic string name that triggers mod_auth_circum for this particular <Location>, <Directory>, or other context you put it in.

6.3 Accepted targets

libcircum accepts the specifications as listed in figure 2 as remote targets to connect to.

uri	::=	[a3proto "`://`"] host ["`:`" port] [transport] [protocol]
a3proto	::=	"`aaa`" | "`aaas`"
host	::=	hostname | ipv4address | "`[`" ipv6address "`]`"
transport	::=	"`;transport=`" l4proto
l4proto	::=	"`tcp`" | "`sctp`"
protocol	::=	"`;protocol=diameter`"

Example: `aaas://[2001:db8::1]:3868;transport=sctp`

Figure 2: EBNF for remote targets accepted by libcircum

7 Troubleshooting

Circumference provides a tool called "minitest", which tests the viability of packet generation, packing, unpacking, extraction, verification, and connecting to remote Diameter servers. It is shipped with the source tarball and normally also gets installed to `${`*pkglibexecdir*`}/minitest`. This utility should help you in diagnosing most basic problems with Circumference without having to have anything else installed and ready, such as the Apache httpd2 modules. A syntax overview is available by running minitest with the `-?` parameter.

References

[FHSopt] FHS – 3.12 `/opt`: Add-on application software packages
`http://www.pathname.com/fhs/2.2/fhs-3.12.html`

[WP] Wikipedia: Diameter (protocol)
`http://en.wikipedia.org/wiki/Diameter_(protocol)`