NAME

mod_auth_tkt − apache ticket authentication module

DESCRIPTION

mod_auth_tkt is a lightweight cookie-based authentication module, written in C, for apache versions 1.3.x, 2.0.x, and 2.2.x It implements a single-signon framework that works across multiple apache instances, different apache versions, and multiple machines.

mod_auth_tkt itself is completely repository-agnostic, as the actual authentication is done by a user-supplied CGI or script in your language of choice (examples are provided in Perl, with contrib libraries for use with python and PHP). This allows authentication against virtually any kind of user repository you can imagine (password files, ldap directories, databases, etc.)

mod_auth_tkt supports inactivity timeouts (including the ability to control how aggressively the ticket is refreshed), the ability to include arbitrary user data within the cookie, configurable cookie names and domains, token-based access to subsections of a site, and optional 'guest' access for unauthenticated users.

CONFIGURATION

mod_auth_tkt is configured in your apache configuration files using the following set of directives (all mod_auth_tkt directives begin with 'TKTAuth'):

Server Directives

mod_auth_tkt supports two apache server-level directives, one required − TKTAuthDigest, the shared secret used for digest hashing − and one optional − TKTAuthDigestType, the type of digest to use in ticket hashes. Both may be global or specific to a virtual host.

TKTAuthSecret <secret>

String − the secret used for digest hashing. This should be kept secret and changed periodically. e.g.

```
TKTAuthSecret "w b@5b15#664038f.f9d8U19b7e25 664eY9ad2%4393e,a2ef"
```

TKTAuthDigestType [MD5 | SHA256 | SHA512]

String, one of MD5 | SHA256 | SHA512. The digest/hash type to use in tickets. The default is MD5, which is faster, but has now been shown to be vulnerable to collision attacks. Such attacks are not directly applicable to mod_auth_tkt, which primarily relies on the security of the shared secret rather than the strength of the hashing scheme. More paranoid users will probably prefer to use one of the SHA digest types, however.

The default is likely to change in a future version, so setting the digest type explicitly is encouraged.

Note that using one of the SHA digest types with the perl CGI scripts requires a version of Apache::AuthTkt >= 2.1.

Directory Directives

All directory-level directives are optional, except that either TKTAuthLoginURL or TKTAuthGuestLogin (or both) must be set to cause mod_auth_tkt to be invoked for a particular directory. As usual, directory-level directives may be set in Directory or Location sections, or in .htaccess files.

AuthType None / require <users>

mod_auth_tkt requires the following standard apache authentication directives to trigger authentication:

```
AuthType None
require valid-user      # or require user1, user2, etc.
```

TKTAuthLoginURL <url>

Standard URL to which unauthenticated users are redirected. This is a required directive unless you are using guest mode via 'TKTAuthGuestLogin on'. e.g.

```
TKTAuthLoginURL https://www.example.com/auth/login.cgi
```

TKTAuthTimeoutURL <url>

URL to which users are redirected in the event their ticket times out. Default: TKTAuthLoginURL. e.g.

```
TKTAuthTimeoutURL https://www.example.com/auth/login.cgi?timeout=1
```

TKTAuthPostTimeoutURL <url>

URL to which users are redirected in the event their ticket times out during a POST operation. This case is distinguished to allow you to handle such cases specially − you probably don't want to redirect back to the referrer after login, for instance. Default: TKTAuthTImeoutURL. e.g.

```
TKTAuthPostTimeoutURL https://www.example.com/auth/login.cgi?posttimeout=1
```

TKTAuthUnauthURL <url>

URL to which users are redirected in the event that they are not authorised for a particular area e.g. incorrect tokens.

```
TKTAuthUnauthURL https://www.example.com/auth/login.cgi?unauth=1
```

TKTAuthGuestLogin <boolean>

Flag to turn on 'guest' mode, which means that any user without a valid ticket is authenticated anyway as the TKTAuthGuestUser user. This is useful for allowing public access for guests and robots, while allowing more personalised or privileged access for users who login. Default: off. e.g.

```
TKTAuthGuestLogin on
```

TKTAuthGuestCookie <boolean>

Flag to indicate whether or not to issue a ticket cookie for guest users. Issuing a cookie is primarily useful where you are using UUID-ed guest users where you want them to keep the initial guest username you issue them for tracking purposes. e.g.

```
TKTAuthGuestCookie on
```

Default is 'off', unless you use a TKTAuthGuestUser with a UUID (see next), in which case it's 'on'. Setting explicitly is recommended, however.

TKTAuthGuestUser <string>

Username to be used for the guest user (in the ticket uid, REMOTE_USER environment variable, etc).

On apache 2.0.x and 2.2.x (but not on apache 1.3.x), the TKTAuthGuestUser may also contain a special sprintf-like pattern '%U', which is expanded to 36−character UUID, allowing individualised guest usernames. The `%U` may also include an integer <= 36 to limit the number of characters used in the UUID e.g. `%12U`, `%20U` etc.

Default: 'guest'. Examples:

```
TKTAuthGuestUser visitor
TKTAuthGuestUser guest−%12U
```

TKTAuthGuestFallback <boolean>

Flag to indicate that a timed out user ticket should automatically fallback to 'guest' status, and issue a new guest ticket, instead of redirecting to the TKTAuthTimeoutURL. Only makes sense with TKTAuthGuestLogin on, of course.

Default: off.

TKTAuthTimeout <seconds>

The ticket timeout period, in seconds. After this period, the ticket is considered stale, and the user is redirected to the TKTAuthTimeoutURL (if set, else to the TKTAuthLoginURL). Note that the ticket can be automatically refreshed, however, using the next setting.

The following units can also be specified on the timeout (with no spaces between timeout and unit): y/years, M/months, w/weeks, d/days, h/hours, m/minutes, and s/seconds.

This timeout is protected by the ticket hashing, so cannot be trivially modified, unlike the TKTAuthCookieExpires setting below.

Setting TKTAuthTimeout to 0 means never timeout, but this is strongly discouraged, as it allows for trivial replay attacks. Set it to a week or two if you really don't want timeouts.

Default: 2h. Examples:

```
TKTAuthTimeout 86400
TKTAuthTimeout 1w
TKTAuthTimeout 1w 4d 3h
```

TKTAuthTimeoutRefresh <decimal>

A number between 0 and 1 indicating whether and how to refresh ticket timestamps. 0 means never refresh (hard timeouts). 1 means refresh tickets every time. .33 (for example) means refresh if less than .33 of the timeout period remains.

This is a politeness setting for those paranoid types who have their browsers set to confirm all cookies − refreshing every time quickly becomes VERY tedious. Default: 0.5. e.g.

```
TKTAuthTimeoutRefresh 0.66
```

TKTAuthCookieName <name>

The name used for the ticket cookie. Default: 'auth_tkt'.

TKTAuthDomain <domain>

The domain to use in ticket cookies, which defines the hosts for which the browser will submit this cookie. Default: the apache ServerName (either global or for a specific virtual host).

TKTAuthCookieExpires <seconds>

NB: This directive is not currently supported on apache 1.3.x!

The period until the cookie expires, used to set the 'expires' field on the ticket cookie, in seconds. This is useful if you want cookies to persist across browser sessions (and your login script must support it too, of course).

The following units can also be specified on the expiry period (with no spaces between period and unit): y/years, M/months, w/weeks, d/days, h/hours, m/minutes, and s/seconds.

Note that his is a **client-side** setting and is not protected by the ticket hashing, so you should always set a TKTAuthTimeout in addition to using an expiry. Cookie expiries are refreshed with tickets if TKTAuthTimeoutRefresh is set.

Default: none. Examples:

```
TKTAuthCookieExpires 86400
TKTAuthCookieExpires 1w
TKTAuthCookieExpires 1w 3d 4h
```

TKTAuthBackArgName <name>

The name used for the back GET parameter. If this is set, mod_auth_tkt will add a GET parameter to all redirect URLs containing a URI-escaped version of the current requested page e.g. if the requested page is http://www.example.com/index.html and TKTAuthBackArgName is set to 'back', mod_auth_tkt will add a parameter like:

```
back=http%3A%2F%2Fwww.example.com%2Findex.html
```

to the TKTAuthLoginURL it redirects to, allowing your login script to redirect back to the requested page upon successful login. Default: 'back'.

TKTAuthBackCookieName <name>

The cookie name to use for the back cookie. If this is set, mod_auth_tkt will set a back cookie containing a URI-escaped version of current requested page when redirecting (see TKTAuthBackArgName above). Default: none.

TKTAuthToken <token>

String indicating a required token for the given location, implementing a simple form of token-based access control. If the user's ticket does not contain one or more of the required tokens in the ticket token list then mod_auth_tkt will redirect to the TKTAuthUnauthURL location (or TKTAuthLoginURL if not set). Your login script is expected to set the appropriate token list up at login time, of course.

Note that this directive can be repeated, and the semantics are that **any** of the required tokens is sufficient for access i.e. the tokens are ORed.

Default: none. e.g.

```
TKTAuthToken finance
TKTAuthToken admin
```

TKTAuthIgnoreIP <boolean>

Flag indicating that mod_auth_tkt should ignore the client IP address in authenticating tickets (your login script must support this as well, setting the client IP address to 0.0.0.0). This is often required out on the open internet, especially if you are using an HTTPS login page (as you should) and are dealing with more than a handful of users (the typical problem being transparent HTTP proxies at ISPs). Default: 'off' i.e. ticket is only valid from the originating IP address. e.g.

```
TKTAuthIgnoreIP on
```

TKTAuthRequireSSL <boolean>

Flag used to indicate that tickets should be refused except in SSL/HTTPS protected contexts (redirects to TKTAuthLoginURL if not, which presumably would be using HTTPS). Default: 'off' (**don't** require SSL). e.g.

```
TKTAuthRequireSSL on
```

See also TKTAuthCookieSecure below.

TKTAuthCookieSecure <boolean>

Flag used to set the 'secure' flag on all ticket cookies issued, indicating to the browser that they should only be sent in SSL/HTTPS protected contexts. Default: 'off' (**don't** set 'secure' flag). e.g.

```
TKTAuthCookieSecure on
```

TKTAuthRequireSSL and TKTAuthCookieSecure are normally used together. One case where it makes sense to use them separately is where you are proxying through a separate SSL-equipped reverse proxy, where you would want to use TKTAuthCookieSecure by itself (since the proxied request will never be via SSL).

TKTAuthDebug <integer>

Turn on mod_auth_tkt debug output messages in your error log, with verbosity increasing with higher integer values. Current range: 1−3.

Note that you will also require apache 'LogLevel debug' set to see these messages.

EXAMPLES

Minimal config using logins:

```
<Location /secret1>
  AuthType None
  require valid-user
  TKTAuthLoginURL https://www.example.com/auth/login.cgi
</Location>
```

Minimal config using guest logins (users can still login explicitly, of course):

```
<Location /secret2>
  AuthType None
  require valid-user
  TKTAuthGuestLogin on
</Location>
```

Example internet configuration:

```
<Location /secret3>
  AuthType None
  require valid-user
  TKTAuthLoginURL https://www.example.com/auth/login.cgi
  TKTAuthTimeoutURL https://www.example.com/auth/login.cgi?timeout=1
  TKTAuthPostTimeoutURL https://www.example.com/auth/login.cgi?timeout=1&post=1
  TKTAuthIgnoreIP on
  TKTAuthTimeout 2h
  TKTAuthCookieExpires 2h
</Location>
```

Example intranet configuration:

```
<Location /secret4>
  AuthType None
  require valid-user
  TKTAuthGuestLogin on
  TKTAuthLoginURL https://www.example.com/auth/login.cgi
  TKTAuthTimeoutURL https://www.example.com/auth/login.cgi?timeout=1
  TKTAuthPostTimeoutURL https://www.example.com/auth/login.cgi?timeout=1&post=1
  TKTAuthTimeout 4h
  TKTAuthCookieExpires 4h
</Location>
```

SUPPORT

Support is available on the mod_auth_tkt mailing list, courtesy of sourceforge:

List
: modauthtkt-users AT lists DOT sourceforge DOT net

List Page and Signup
: https://lists.sourceforge.net/lists/listinfo/modauthtkt-users

List Archive
: http://sourceforge.net/mailarchive/forum.php?forum=modauthtkt-users

BUGS

Ticket payload should include IP address, to make debugging IP address problems easier.

AUTHOR

Gavin Carr <gavin AT openfusion DOT com DOT au>

LICENCE

mod_auth_tkt is licensed under the terms of the Apache Licence.