NAME

apt-transport-https − APT transport for downloading via the HTTP Secure protocol (HTTPS)

DESCRIPTION

This APT transport allows the use of repositories accessed via the HTTP Secure protocol (HTTPS), also referred to as HTTP over TLS. It is available by default since apt 1.5 and was available before that in the package apt−transport−https. Note that a transport is never called directly by a user but used by APT tools based on user configuration.

HTTP is by itself an unencrypted transport protocol (compare **apt-transport-http**(1)), which, as indicated by the appended S, is wrapped in an encrypted layer known as Transport Layer Security (TLS) to provide end−to−end encryption. A sufficiently capable attacker can still observe the communication partners and deeper analysis of the encrypted communication might still reveal important details. An overview over available alternative transport methods is given in **sources.list**(5).

OPTIONS

The HTTPS protocol is based on the HTTP protocol, so all options supported by **apt-transport-http**(1) are also available via Acquire::https and will default to the same values specified for Acquire::http. This manpage will only document the options *unique to https*.

Server credentials

By default all certificates trusted by the system (see ca−certificates package) are used for the verification of the server certificate. An alternative certificate authority (CA) can be configured with the Acquire::https::CAInfo option and its host−specific option Acquire::https::CAInfo::*host*. The CAInfo option specifies a file made up of CA certificates (in PEM format) concatenated together to create the chain which APT should use to verify the path from your self−signed root certificate. If the remote server provides the whole chain during the exchange, the file need only contain the root certificate. Otherwise, the whole chain is required. If you need to support multiple authorities, the only way is to concatenate everything.

A custom certificate revocation list (CRL) can be configured with the options Acquire::https::CRLFile and Acquire::https::CRLFile::*host*. As with the previous option, a file in PEM format needs to be specified.

Disabling security

During server authentication, if certificate verification fails for some reason (expired, revoked, man in the middle, etc.), the connection fails. This is obviously what you want in all cases and what the default value (true) of the option Acquire::https::Verify−Peer and its host−specific variant provides. If you know *exactly* what you are doing, setting this option to "false" allows you to skip peer certificate verification and make the exchange succeed. Again, this option is for debugging or testing purposes only as it removes all security provided by the use of HTTPS.

Similarly the option Acquire::https::Verify−Host and its host−specific variant can be used to deactivate a security feature: The certificate provided by the server includes the identity of the server which should match the DNS name used to access it. By default, as requested by RFC 2818, the name of the mirror is checked against the identity found in the certificate. This default behavior is safe and should not be changed, but if you know that the server you are using has a DNS name which does not match the identity in its certificate, you can set the option to "false", which will prevent the comparison from being performed.

Client authentication

Besides supporting password−based authentication (see **apt_auth.conf**(5)) HTTPS also supports authentication based on client certificates via Acquire::https::SSLCert and Acquire::https::SSLKey. These should be set respectively to the filename of the X.509 client certificate and the associated (unencrypted) private key, both in PEM format. In practice the use of the host−specific variants of both options is highly recommended.

EXAMPLES

```
Acquire::https {
        Proxy::example.org "DIRECT";
        Proxy "socks5h://apt:pass@127.0.0.1:9050";
        Proxy−Auto−Detect "/usr/local/bin/apt−https−proxy−auto−detect";
```

```
          No−Cache "true";
          Max−Age "3600";
          No−Store "true";
          Timeout "10";
          Dl−Limit "42";
          Pipeline−Depth "0";
          AllowRedirect "false";
          User−Agent "My APT−HTTPS";
          SendAccept "false";

          CAInfo "/path/to/ca/certs.pem";
          CRLFile "/path/to/all/crl.pem";
          Verify−Peer "true";
          Verify−Host::broken.example.org "false";
          SSLCert::example.org "/path/to/client/cert.pem";
          SSLKey::example.org "/path/to/client/key.pem"
};
```

SEE ALSO

apt-transport-http(1) **apt.conf**(5) **apt_auth.conf**(5) **sources.list**(5)

BUGS

APT bug page[1]. If you wish to report a bug in APT, please see /usr/share/doc/debian/bug−reporting.txt or the **reportbug**(1) command.

AUTHOR

APT team

NOTES

1. APT bug page
   http://bugs.debian.org/src:apt