Subject: NetProxy package. (HTTP/SOCKS4/4a/5 with BIND support) encapsulation for U++
Posted by Oblivion on Sun, 24 Sep 2017 17:54:39 GMT
View Forum Message <> Reply to Message

Hello,

Netproxy is here with it's new design.

NetProxy package for Ultimate++

This delegate class, formerly known as NetworkProxy, encapsulates two widely used network proxy protocols: Http tunneling and SOCKS.

Features and Highlights

- Uses HTTP_CONNECT method for http tunneling.
- Encapsulates SOCKS proxy protocol version 4/4a, and version 5, as defined in RFC 1928 and RFC 1929.
- In SOCKS mode, NetProxy can work with both IPv4 and IPv6 address families.
- In SOCKS mode, NetProxy allows BIND requests.
- In SOCKS mode, NetProxy allows remote name lookups (DNS).
- Supports both synchronous and asynchronous operation modes.
- Allows SSL connections to the target machine (not to proxy itself) in both Http and SOCKS modes.
- Package comes with full public API document for Topic++, and has a typical BSD license.

Known Issues

- In SOCKS mode, NetProxy currently does not allow UDP association.

History

- 2017-09-23: Initial release of version 2.0:
    A change in naming: NetworkProxy is from now on called NetProxy.
    This change is in parallel with the change in class design.
    There is now a single class that can make both HTTP_CONNECT and SOCKS requests.
    Handling of socks BIND requests is simplified.
    Internal cleanup redesigne allowed some performance gain around %4.

New design brings a single class with both HTTP_CONNECT tunneling, and SOCKS protocol support, a simpler interface. Also better optimization.

Since NetProxy is a "delegate" class, which takes a socket as it's client and upon finishing its job hands it over back to the user, it is in most cases trivial to add proxy support to existing classes and apps.

All you need to do is pass a socket handle and the host information (address, port) using an Event/Gate, or Function (before actual connection).

For example below addition can do the trick (it can work as a completely optional plugin):

```
class NetworkFoo {

public:
    Gate<TcpSocket&>   WhenProxy; // Or WhenConnect
};


//...

NetworkFoo nwf;

nwf.WhenProxy = [=, proxy_host, proxy_port, target_host, target_port](TcpSocket& sock) {

                NetProxy proxy(sock, proxy_host, proxy_port);
                proxy.Timeout(10000)
                    .Auth("user", "password")
                    .Socks5();
                return proxy.Connect(target_host, target_port);
            };
```

Bug reports, criticism, reviews are appreciated.

Best regards,
Oblivion

File Attachments

1) NetProxy.zip, downloaded 628 times