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 423 times

Subject: Re: NetProxy package. (HTTP/SOCKS4/4a/5 with BIND support) encapsulation for U++
Posted by Oblivion on Sat, 06 Jun 2020 21:04:40 GMT
View Forum Message <> Reply to Message

Hi,

NetProxy package has received a small update. (Minor bug fixes.)

Also, I've added two simple Socks5 connection examples to upp-components/Examples directory:

SocksProxyExample - Demonstrates a simple SOCKS5 connection, using NetProxy package.
SocksProxyExampleNB - Demonstrates a simple, non-blocking SOCKS5 connection, using NetProxy package.

SocksProxyExample:

```
#include <Core/Core.h>
#include <NetProxy/NetProxy.h>

using namespace Upp;

// This example demonstrates the basic usage of NetProxy class with socks5 tunnels.
// Default proxy server: Turk Telekom, a well-known ISP in Turkey. No auth required...
// Target server: test.rebex.net -> A well-known FTP/SFTP test server.

CONSOLE_APP_MAIN
{
 StdLogSetup(LOG_COUT|LOG_FILE);
 NetProxy::Trace();

 const char *proxy_server = "88.249.26.113";
 const int   proxy_port   = 1080;

 TcpSocket sock;

 NetProxy socksproxy(sock, proxy_server, proxy_port);
 if(socksproxy.Timeout(10000).Socks5().Connect("test.rebex.net", 21)) {
  RLOG("-------------");
  RLOG(sock.GetLine()); // Get the first line of FTP server HELO...
  RLOG("-------------");
  return;
 }
 RLOG(socksproxy.GetErrorDesc());
}
```

If you have any questions, let me know.

Best regards,
Oblivion

Subject: Re: NetProxy package. (HTTP/SOCKS4/4a/5 with BIND support) encapsulation for U++
Posted by Oblivion on Wed, 10 Jun 2020 11:11:16 GMT
View Forum Message <> Reply to Message

Hi

I've added a Socks5-proxied SSH connection example to upp-components repo.
Example uses the Upp's Core/SSH package and also demonstrates the usage of WhenProxy event of SshSession class.

Here is the code:

```
#include <Core/Core.h>
#include <Core/SSH/SSH.h>
#include <NetProxy/NetProxy.h>

using namespace Upp;

// This example demonstrates a basic SSH2 connection over a Socks5 proxy.

// WARNING: The sole purpose of this example is to demonstrate the  usage of
//          NetProxy package with the SSH package. The example uses an anon.
//          proxy server (with  no authentication) to access  a public  SFTP
//          server.
//
//     NEVER  USE A PUBLIC PROXY SERVER FOR YOUR SSH CONNECTIONS. IN FACT,
//          IT IS HIGHLY RECOMMENDED TO COMPLETELY AVOID USING PROXY SERVERS
FOR
//          SENSITIVE/ENCRYPTED DATA TRANSFERS.

CONSOLE_APP_MAIN
{
 StdLogSetup(LOG_COUT|LOG_FILE);
 Ssh::Trace();
 NetProxy::Trace();

 const char *proxy_server = "88.249.26.113";  // Anonymous socks proxy server of a well-known
ISP in Turkey (Turk Telekom).
 const int   proxy_port   = 1080;
```

```
 const char *ssh_server   = "test.rebex.net"; // A well-known and popular SSH test server.
 const int   ssh_port     = 22;

 SshSession session;
 session.WhenProxy = [&]() -> bool
 {
  NetProxy socksproxy(session.GetSocket(), proxy_server, proxy_port);
  return socksproxy.Timeout(10000).Socks5().Connect(ssh_server, ssh_port);
 };

 if(session.Timeout(30000).Connect(ssh_server, ssh_port, "demo", "password")) {
  SFtp sftp(session);
  String s = sftp.LoadFile("./readme.txt");
  if(!sftp.IsError()) {
   RLOG("---------------------------");
   RLOG(s);
   RLOG("---------------------------");
  }
  return;
 }
 RLOG(session.GetErrorDesc());
}
```

Best regards,
Oblivion

Subject: Re: NetProxy package. (HTTP/SOCKS4/4a/5 with BIND support) encapsulation for U++
Posted by Oblivion on Tue, 23 Feb 2021 21:11:46 GMT
View Forum Message <> Reply to Message

Hi,

NetProxy package and its examples are now also available via UppHub.

Best regards,
Oblivion

Subject: Re: NetProxy package. (HTTP/SOCKS4/4a/5 with BIND support) encapsulation for U++
Posted by superdev on Fri, 14 Feb 2025 01:29:45 GMT
View Forum Message <> Reply to Message

Couldn't make HttpRequest work through proxy. NetProxy successfully connects to proxy server, then HttpRequest receives result(GET) using no proxy.

Subject: Re: NetProxy package. (HTTP/SOCKS4/4a/5 with BIND support) encapsulation for U++
Posted by Oblivion on Fri, 14 Feb 2025 07:51:19 GMT
View Forum Message <> Reply to Message

Hello superdev,

Quote:Couldn't make HttpRequest work through proxy. NetProxy successfully connects to proxy server, then HttpRequest receives result(GET) using no proxy.

If you are trying to use an HTTP_CONNECT-based proxy, HttpRequest class already comes with that feature (supports internally). Just check the docs, you don't really need NetProxy.
If you are trying to use a socks-based proxy with HttpRequest class, unfortunately that is not possible at the moment. (HttpRequest class needs to be patched in order to make it work.)

The reason is, HttpRequest (HR) keeps track of many states and does redirections. It is a one big beast and it wasn't designed with utilizing NetProxy in mind.
So, even if you can connect an HR using NetProxy, HR's internal state is not adjusted accordingly, hence it will not work unless it is patched.

Best regards,
Oblivion
