Remote debugging with IDA Pro.

Since version 4.8, IDA Pro supports remote debugging of x86/AMD64 Windows PE applications and Linux ELF applications over TCP/IP networks. Remote debugging is the process of debugging code running on one networked computer from another networked computer:

- The computer running the IDA Pro interface will be called the "debugger client".
- The computer running the application to debug will be called the "debugger server".

Remote debugging will be particularly useful in the following cases:

- To debug virus/trojans/malwares : in this way, the debugger client will be as isolated as possible from the compromised computer.
- To debug applications encountering a problem on one computer which is not duplicated on other computers.
- To debug distributed applications.
- To always debug from your main workstation, so you won't have to duplicate IDA configuration, documentation and various debugging related resources everywhere.
- In the future, to debug applications on more operating systems and architectures.

This small tutorial will present how to setup and use remote debugging in practice.

The remote IDA debugger server.

In order to allow the IDA client to communicate with the debugger server over the network, we must first start a small server which will handle all low-level execution and debugger operations. The IDA distribution ships with a Windows debugger server (the ***win32_remote.exe*** file) and a Linux debugger server (the ***linux_server*** file). With these, we can:

- Locally debug x86/AMD64 Windows applications and DLLs from the IDA Windows graphical and text versions.
- Remotely debug x86 Linux applications and shared libraries from the IDA Windows graphical and text versions.
- Locally debug x86 Linux applications and shared libraries from the IDA Linux text version.
- Remotely debug x86/AMD64 Windows applications and DLLs from the IDA Linux text version.

So let's first copy the small Windows debugger server file to our debugger server.

This server accepts various command line arguments:

```
C:\> win32_remote -?
IDA Windows32 remote debugger server. Version 1.0. Copyright Datarescue 2004
Error: usage: ida_remote [switches]
  -p... port number
  -P... password
  -v    verbose
```

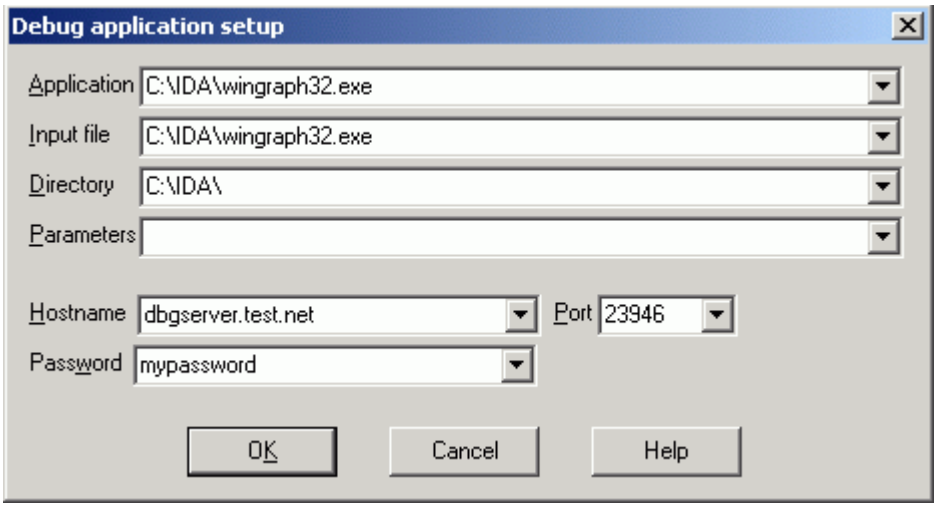
Let's start it by specifying a password, to avoid unauthorized connections:

```
C:\>win32_remote -Pmypassword
IDA Windows32 remote debugger server. Version 1.0. Copyright Datarescue 2004
Listening to port #23946...
```

Note that the remote debugger server can only handle one debugger session at a time. If you need to debug several applications simultaneously on the same host, launch several servers on different network ports by using the *-p* switch.

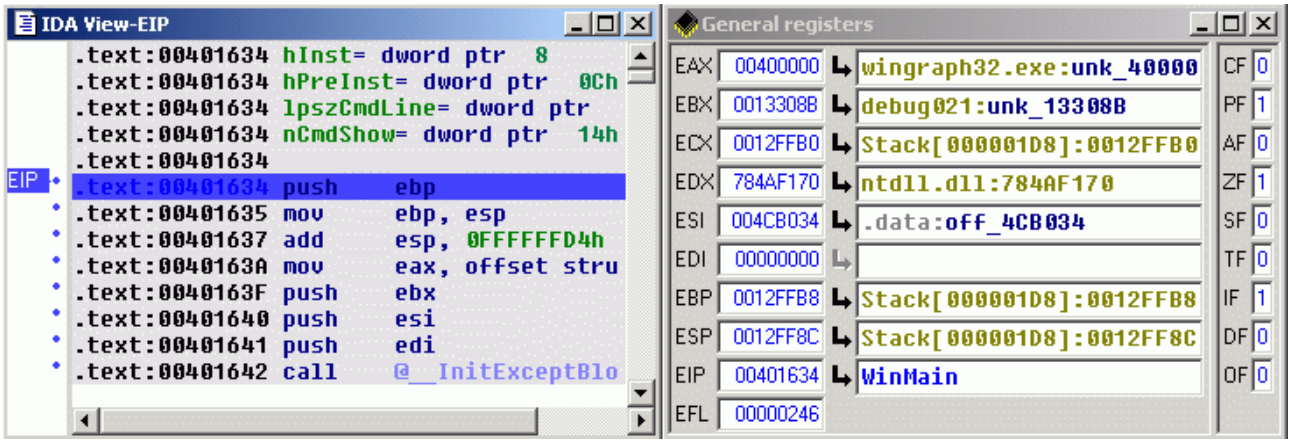
Setting up the debugger client.

First, we copy the executable we want to debug from the debugger server (Windows or Linux) to the debugger client (Windows or Linux). We can then load this file into IDA, as usual. To setup remote debugging, we select the '*Process options...*' menu item in the *Debugger* menu:



Specify the *Application*, *Directory* and *Input file* paths. Note that these file paths should be valid on the remote debugger server. Also do not forget to enter the host name or IP address of the debugger server: remote debugging will only be enabled if these settings are specified ! Finally, we enter the password we chose for the remote IDA debugger server.

Both debugger server and debugger client are now ready to start a remote debugging session. In fact, you can now use all debugger related commands as you would with the local Windows PE debugger or local Linux debugger! For example,we can run the process until EIP reaches the application entry point, by jumping to this entry point then pressing the F4 key:

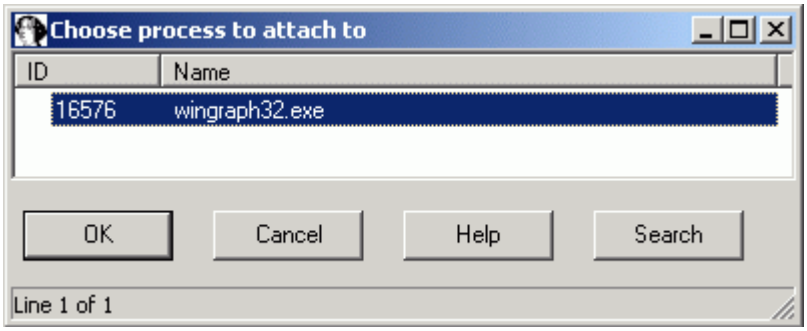


If we now directly terminate the process (by pressing CTRL-F2) and look at *win32_remote*'s output (on the debugger server), we indeed properly observe it accepted then closed our network connection:

```
C:\> win32_remote -Pmypassword
IDA Windows32 remote debugger server. Version 1.0. Copyright Datarescue 2004
Listening to port #23946...
Accepting incoming connection...
Closing incoming connection...
```

Attaching to a running process.

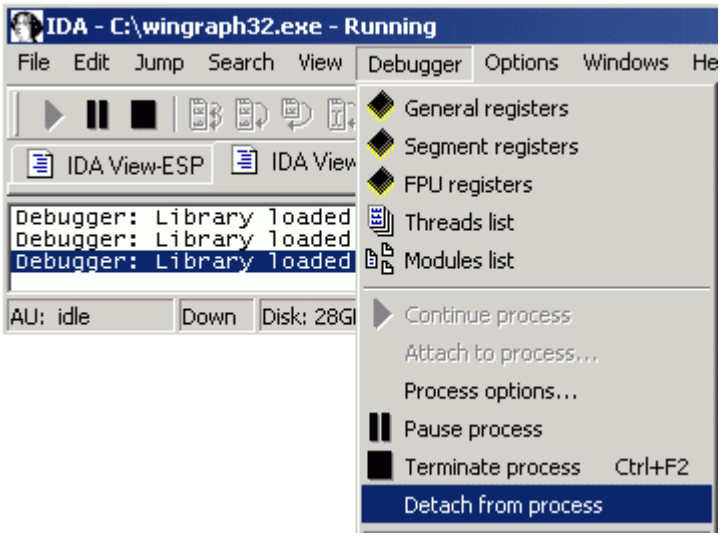
Another interesting possibility is to attach to an already running process on the remote computer. If you click on the '*Attach to process...*' command from the Debugger menu, IDA will display a listing of all remote running processes corresponding to the file in your disassembly database:



Double clicking on a process from the list will automatically suspend the process and attach to it, allowing you to debug it without starting it manually. This attach process works from Windows to Linux, from Windows to Windows, from Linux to Linux and from Linux to Windows.

Detaching from the debugged process.

Finally, if the debugger server is running Windows XP, Windows Server 2003 or Linux, you can also detach from a process you were currently debugging, simply by using the '*Detach from process*' command in the *Debugger* menu:



IDA supports debugging of DLLs on Windows and shared libraries on Linux. On Windows, please note that IDA can also attach to Windows services running either locally or remotely. In particular, the '*Detach from process*' command will be especially useful if you previously attached to a Windows service: it will allow you to stop the debugger without terminating a critical Windows service on the debugger server!

Revision 1.1

DataRescue SA/NV
40 Bld Piercot
4000 Liège, Belgium
T: +32-4-3446510 F: +32-4-3446514