

Making Testing Easier With BiDiBLAH

Discuss in Forums {mos_smf_discuss:J. Peltier}

By Justin Peltier, CTO, Peltier Associates

If you did not get a chance to go to the Black Hat/Defcon conference this year you have probably not heard of BiDiBLAH yet. BiDiBLAH is a new pseudo utility from the folks at SensePost (<http://www.sensepost.com>). For quite some time now the talented developers of SensePost have been involved in the Google Hacking community and they have released a number of scripts/utilities that very handy for trolling for information from Google.

BiDiBLAH is the latest release from the SensePost group and it takes a major step forward from the scripting of Google searches. The BiDiBLAH utility is a framework that can be used to assist in automating the vulnerability assessment/ethical hacking process. BiDiBLAH is a Windows® based utility that allows starting the testing process to be point and click easy. The BiDiBLAH utility does not try to recreate already well known, used, and supported open source vulnerability applications; rather BiDiBLAH uses the existing ethical hacking/vulnerability assessment tools – these include both Nessus (<http://www.nessus.org>) and MetaSploit (<http://www.metasploit.com>). As well as the existing Google Hacking scripts from SensePost.

As of this writing the full version of BiDiBLAH is not currently for sale, but the version that is available is fully functional with two exceptions:

1. The scan time is limited to one hour
2. The save data feature has been disabled

The Installation

Getting BiDiBLAH running is not a trivial task. The primary installation of the utility is straight forward and runs in the install.exe format. Once the installation of the BiDiBLAH utility is complete you will need to make some changes to your operating system to allow the utility to function. The first step is to load a raw packet driver that will allow the BiDiBLAH utility to send the packets necessary for port scanning and banner grabbing. When you have completed this step you network card configuration should have the added protocol like the example below.

Figure 1 – BiDiBLAH Raw Packet Driver Installation

The next step is to disable the Windows® firewall and any other Personal Firewall software that you may have running on your target system. Just disabling the personal firewall is not enough, because the BiDiBLAH utility need to still block incoming RST (reset) packets. The recommendation from the SensePost group is to install the free personal windows firewall – wipfw available from <http://sourceforge.net/projects/wipfw>. I was truly amazed at the functionality of wipfw. It allows more granular filtering than most commercial personal firewalls, but the discussion of this utility will be in another article. All that is needed for purposes of BiDiBLAH is to create one rule as mentioned before to block the RST packets. This can be done from the command line by issuing the following command:

```
ipfw add 00100 deny TCP from any to any tcpflags rst
```

Or through the new wipfw GUI like the figure below:

Once you have the new rule added you can check the config by running:

```
Ipfw list
```

From the command line. This completes the installation of BiDiBLAH. The next step is to configure BiDiBLAH to run the security checks that you want it to.

The Configuration

To configure BiDiBLAH double click on the icon and you should see a pop-up message like this:

As of this writing the pay version has not been released. Simply click on OK to go to the interface. The interface has many tabs across the top that are used for the configuration. To configure you copy of BiDiBLAH change the options in the following tabs:

1. At the Subdomain tab:

-

Enter your Google API key (You can get a key at api.google.com)

-

The Google depth (in multiples of 10) sets how many queries should be returned

-

The Google keywords are words that BidiBLAH use to combine with queries

2. At Forwards tab:

-

Select where your BFDNS files are. The application will look for any file that ends with a .bfdns extension and add its content to the list of names that will be used for brute force.

-

The test depth sets how deep within each file the application will test before assuming a naming scheme

-

If you want to test all the entries you can check the override checkbox

3. At Portscan tab:

-

Enter the source IP where QALive will send packets from. If this is not your IP address, packets will be spoofed from the address that you selected. This could be useful when you are running a tcpdump somewhere else…

-

Enter your source mac address – you can get it doing an ipconfig /all in a DOS window

-

Enter the destination mac address. Because we haven't implemented ARP you need to set this up manually. Most of the time it isn't a big deal though – you will probably be scanning machines on the other side of your default gateway. That makes the destination mac address that of your default gateway. You can get this easily by looking at your ARP table. Do an "arp -a" in a DOS window. If you are scanning locally…sorry (or you can hook a router between you and your local net.

-

Load the port list file – this is a single text file containing the ranges of ports you wish to see as a drop down list (in QALive).

4. At Nessus tab:

-

Select the Nessus server (IP or DNS name), Nessus username and password

-

Select where the application should find the PLG files (Nessus plugin selection file). This will appear in the plugin set drop down list in the Nessus section.

5. At MetaSploit tab:

-

Enter the location of Metasploit framework's web interface

-

Enter the location of your local MSF home – this is used when configuring your exploits

-

If your exploits are already configured you can save the config strings in file and load it

-

You should also load the Metasploit 2 Nessus text file. This matches Nessus plugins to Metasploit exploits

-

The PERL interpreter used for Metasploit needs to be set

-

You can test your Metasploit setup by clicking “load exploits” in the MetaSploit tab – you should see a list of exploits. Double clicking on the exploit brings up the exploit configuration screen.

6. When you are done configuring:

-

Click on the SAVE button in the “Config Load/Save” section – next time you start BiDiBLAH you can now just click on the blue LOAD button and you don’t have to go through the whole mission again.

7. Loading and saving configurations:

-

Choose the “Load Config” tab to load a sample configuration file located in c:\bidiblah\config (if you chose defaults). The location of the BFDNS files, a default set of ports in the portlist file as well as the IP2C DB should be configured correctly. If you installed the application in a different location you need to configure these manually.

-

At any stage you can save the configuration (and load it later again).

Note: Much of the above was taken from the Quick Start guide

Running BiDiBLAH

Now that the configuration is done it is time to use the utility. The first step that I couldn't find documented anywhere and it was also a little non-intuitive is to create a file that contains your target domain name. In this case I simply opened notepad and created a file that contained the domain that I was interested in searching on: peltech.com. I loaded this file in the subdomain tab by clicking on the Import (file) button and browsing over to the newly created file.

The search on my domain was far from interesting as you can see below:

By then again the domain that I choose was mine and it is pretty small. So I changed my search and added Microsoft.com – I'll share these results later. Following up on using peltech.com for the search string I moved to the forward tab. The first step that was necessary here was to import the data from the subdomains tab. This was done by clicking on the Import (app) button. It automatically moved the data from the Sub-domains search into the Forward application.

The results and function of the Forward part of the utility was pretty disappointing. It appears like the utility is trying to gather more information about the target through DNS information mining, but this is just done at a very high level and not nearly to the depth that I would perform in a manual test. The ethereal packet capture below shows the check performed by BiDiBLAH. For reference the machine I was using had the primary DNS server set to the 4.2.2.1 IP address.

The search would have been more effective if the primary query requested the SOA (Start Of Authority) record first and all NS (Nameservers) second. Then followed the responses of these two queries by setting the server first to the SOA result and then to the secondary NS results. In the case above the MX (Mail Exchanger) and the NS are queried by the subsequent queries appear to still continue with using 4.2.2.1 for the DNS server.

Once the Forward search is done the next set of tests appear under the Netblocks tab. This tab was once again a little non-intuitive and the documentation was lacking. To import the information from the Forwards section click on the import (app) button just as on the Forwards section. In the case of my domain it warned that the utility was assuming a class "C" block of IP addresses because it could not find more information. Once you have loaded in your block it is necessary to double click on the netblock itself in the far right pane. This will load the data into the Forwards in the block section. From here there is a button to perform a whois search on the IP block. This left me with a screen that looked like the following:

From here I was stumped for what to do next. I was now about one hour into the BiDiBLAH experience and I had to restart the demo version of the software. The next set of tools was called Reverse. Once again data was imported from the previous section by using the import(app) button. From here a start button was all that was needed to start the next wave of checks. This set of tests was useful, but could be improved with a bit of modification. The ethereal packet capture below shows the types of queries that the utility was sending:

As you can see above the utility was performing RDNS requests. I was happy to see this because it is a slow and time consuming process to do this by hand, but just as the previous DNS – Forward section the utility uses my client's configure primary DNS server and not the SOA or other nameserver associated with the target domain. Notice in the packet capture that the queries are directed to the 4.2.2.1 IP address again.

This lead to no reverse DNS information being returned to the BiDiBLAH utility. The finished result looked the figure below:

On the next tab – Port Scanner – you need to make sure that in your setup you put in your IP address, your MAC address, and a destination MAC address. Then click on the Bind Adapter button and select your adapter from the drop down list. If you miss any of these steps you will see the following message:

Once you have the adapter issues worked out you can use the drop down list below the adapter to set the ports that you would like to scan. As you can see from the packet capture below the utility sends a SYN packet to the ports selected in sequential order.

Randomizing the selected ports and allowing for other port scan types would be useful here or the ability to use nmap for the scanning like Nessus does. Also the scanner missed a few open ports (I am not sure why and did not have the time to investigate). Here is the final output from the port scan.

In the banners section I had to create another text file with the IP addresses of my servers to see if I could continue with the utility. By using just notepad again I inserted a few IPs of live systems to see if BiDiBLAH could grab the banners. Once again I have no idea why this is the case, but the queries looked strange while looking at them through ethereal. Here is what I saw:

I found these queries odd because here is the list of IPs that I fed to the utility as an input:

I did not see them appear anywhere in the packet capture (shown above).

In the next section (targeting) the top IP address – 72.41.28.76 did appear as a potential target. I selected this target and then moved to the following section Nessus. The Nessus scan through the import(app)button loads in the IP address as a correct Nessus target. From the plugins drop down menu there are a few plugins sets that can be selected. The information about the plugins sets seems to be lacking (my best guess is the .nessusrc file can be copied over) but that is just a guess. In some documentation that I found there were options on this screen to configure the nessus plugins set. However on my version these options are missing. This set of tests actually ran quite well. I had the nessus server installed on a RedHat 9 VMWare image running on my current machine. The BiDiBLAH utility was able to log into the nessus server and run the tests without any difficulty. I was impressed with how well done this is. Just as before the port scan did not return a result and so my scanning ended here, as the next set of utilities was Metasploit and I could not run Metasploit without nessus finding an open port. At this point I created a new target on my local system and began the testing from the port scanning section.

The last section the Metasploit section looked promising. The meta2nessus feature seemed to be useful as it loaded the nessus checks and the corresponding Metasploit attack for the vulnerability. As I was running out of time I did not get a chance to continue testing the last two phases.

Summary

BiDiBLAH, as it is, is not quite ready for commercial companies to use with vulnerability assessment. However the functionality is pretty impressive for this early stage. The documentation needs to be a bit better and in large part that is what I think some of what this column actually is. Once these issues are ironed out BiDiBLAH can be a massive time saver for the regular security tester. As noted in the version that I was using to test, the DNS interrogation needs a bit of

a tweak, but this can be fixed by changing the client's IP configuration to use the SOA for the target as the primary DNS, and also the portscan check can miss open ports. In next month's column we will revisit BiDiBLAH and see what we were able to find out with more time to test.

References:

BiDiBLAH Quick Start Guide – available from <http://www.sensepost.com/research/bidiblah>

SensePost Lecture at Black Hat – available from <http://blackhat.com/presentations/bh-usa-05/bh-us-05-sensepost.pdf>

Wipfw – available from <http://wipfw.sourceforge.net/>

Nessus – available from <http://www.nessus.org>

Metasploit – available from <http://www.metasploit.com>