AimSniff

Jason Healy, Director of Networks and Systems

Last Updated Mar 18, 2008

Contents

Chapter 1

AimSniff

1.1 Introduction

This document describes Suffield Academy's use of AimSniff, an open-source system for scanning and classifying AOL Instant Messages.

1.2 Dependencies

These instructions assume a moderate level of Linux system administration experience. Some of the dependencies for AimSniff are not installed by default on many systems, so you must fetch and build them yourself.

For this tutorial, we make several assumptions about the system you are configuring. If your configuration does not match exactly, the instructions may still work. However, you will need to be mindful of any deviations from the following:

- A computer with Debian Linux installed (these instructions written with the pre-release ”Sarge” version of Debian). Please see our Debian Installation Instructions for more information on installing this version of Debian.
- A computer with two NICs installed. One will be used for sniffing the network, and the other will be used for regular communication with other hosts. We require this because our switching equipment does not allow a port to pass normal traffic when it is in ”monitor” mode.
- A MySQL or PostgreSQL database. We have patched AimSniff to store data into PostgreSQL. If you prefer MySQL, use the stock AimSniff package, which uses MySQL by default.

If you've got all these things, then we're ready to go!

1.2.1 Network Setup

If your swtiching hardware does not allow normal traffic when a port is set into "monitor" (or sniffing) mode, you'll need a second NIC in your machine. Because no normal traffic will pass to this interface, we can give it a bogus static address.

Edit the file `/etc/network/interfaces` and create (or modify) a stanza for your second ethernet card. In our case, the unused card was `eth1`:

```
auto eth1
iface eth1 inet static
      address 127.127.127.127
      netmask 255.255.255.255
```

Now, bring the interface up using `ifup eth1`. You should see the interface appear with the bogus address listed above.

1.2.2 AimSniff Dependencies

In order to run AimSniff, you'll need several Perl modules (all of which are documented in the AimSniff README).

Several of these dependencies exist in Debian, and can be installed directly via apt-get. Others, however, must be fetched and built by hand.

Debian Packages

The easiest dependencies to install are those included in Debian. Run the following command (as root) on your machine:

```
apt-get install smbclient libunicode-string-perl libnet-pcap-perl \
                libdbi-perl libproc-process-perl libproc-daemon-perl \
                libunix-syslog-perl
```

The command should appear all on one line, or you may use backslashes to continue to new lines, as we have done above.

This may require installing other dependencies as Debian sees fit. Install any additional required packages.

SMBClient is required for the resolution of NT usernames on a Windows network, and is not strictly required for AimSniff to operate. If you do not need this functionality, you may omit `smbclient` from the install process.

dh-make-perl

To build additional Perl packages that aren't included in Debian, we'll need to download and compile them ourselves. Fortunately, Debian includes a tool, `dh-make-perl`, which automates much of this process.

To install `dh-make-perl`, simply install it using APT:

```
apt-get install dh-make-perl
```

Again, this may install extra dependencies. Go ahead and accept their installation.

Finally, you may wish to customize the `DEBFULLNAME` and `DEBEMAIL` environment variables. While they won't affect whether your packages build or not, they will set the maintainer information for the packages.

NetPacket Package

The `NetPacket` Perl package is required by AimSniff in order to run.

Lazy people should feel free to download a pre-built Debian package from our `debs` repository. If that doesn't work, follow the instructions below to build your own.

The source package can be downloaded from CPAN at the following link:

http://search.cpan.org/CPAN/authors/id/A/AT/ATRAK/NetPacket-0.04.tar.gz

Download the package source to a suitable directory (`/usr/local/src/` might be a good choice, though the location doesn't matter).

Unpack the source distribution:

```
tar -zxf NetPacket-0.04.tar.gz
```

and change into the resulting directory:

```
cd NetPacket-0.04
```

At this point, we're ready to "debianize" the Perl package. Run the following:

```
dh-make-perl
```

That will turn the source package into a Debian source tree. Now you can run:

```
dpkg-buildpackage -rfakeroot
```

(or `-rsudo` if you prefer to use sudo)

Which will create a debian package in the parent directory. Several files starting with `libnetpacket-perl` will be created. The one we're interested in should be called `libnetpacket-perl_0.04--1_all.deb`

Go ahead and install your new package now, by changing to the parent directory and running `dpkg`:

```
cd ..
dpkg -i libnetpacket-perl_0.04--1_all.deb
```

Proc::Simple Package

The `Proc::Simple` Perl package is required by AimSniff in order to run.

Lazy people should feel free to download a pre-built Debian package from our `debs` repository. If that doesn't work, follow the instructions below to build your own.

The source package can be downloaded from CPAN at the following link:

http://search.cpan.org/CPAN/authors/id/M/MS/MSCHILLI/Proc-Simple-1.21.tar.gz

To build the package, follow the instructions above for installing the `NetPacket` package. This time, however, substitute `Proc-Simple-1.21.tar.gz` for the source package, and `libproc-simple-perl_1.21-1_all.deb` for the name of the resulting Debian package file.

Again, a simple install should finish the job:

```
dpkg -i libproc-simple-perl_1.21-1_all.deb
```

1.2.3 Testing It Out

At this point, you have installed all of the dependencies required to run AimSniff in collector mode.

You should download AimSniff and try to run it. If you're using the stock distribution, you may obtain it here:

http://www.aimsniff.com/releases/aimsniff-0.9d.tar.gz

Suffield Academy has customized the collection script to work with our database server (we use PostgreSQL instead of MySQL). The latest version of the script can be downloaded from our `aimsniff-postgres` repository.

Once you've downloaded (and untarred, if you're using the stock distribution) the files, make the `aimSniff.pl` script executable and run it with the following arguments:

```
chmod 755 aimSniff.pl
./aimSniff.pl --nodb
```

The program should begin running. If you're not root, it will not have access to the network device and will quit. This is OK.

If you get any error messages from Perl about being unable to locate a module, then you need to make sure you installed all the dependencies as described above. The message should give you a hint as to what module is missing (*e.g.*, if it can't find `Proc::Simple.pm` then you'll need to redo the installation of that package).

Once the program starts up normally, you're ready to move on to the next step.

1.3 Database Setup

Now that we've installed the dependencies, we're ready to set up the database. AimSniff can run in one of several modes:

- Display only (sniffed data are printed out, but not stored)
- Local file (sniffed data sent to a flat file)
- Database (sniffed data sent to a relational DB)

We'll focus on the third method, as it provides the most flexibility in reporting later on. If you don't want/need a database, consult the AimSniff documentation on how to use the other collection options.

1.3.1 Selecting a DB Server

AimSniff can connect to a local or remote database server in order to store its data. You can either set up a local database on the same machine as AimSniff, or you can use an existing database on another machine.

At Suffield, we use a separate machine, as we already have a centralized database server. Additionally, our AimSniff machine has limited resources (processor, RAM, HD space), so sending the data to another machine keeps the overhead low on our collection machine.

This tutorial assumes you know how to install and configure one of the following relational database packages:

MySQL MySQL is the default database backend for AimSniff. If you already use MySQL, or if you don't have a database set up yet, you may want to use MySQL. If you choose MySQL, download the stock distribution of AimSniff instead of our version (which is modified to use PostgreSQL instead).

PostgreSQL PostgresSQL is our database of choice at Suffield Academy, for a variety of reasons that are beyond the scope of this document. We have modified AimSniff to work with Postgres instead of MySQL. If you wish to use Postgres as your backend, please download our modified copy of AimSniff.

Once you've selected a database, you should install it on your database server (if you haven't done so already). From this point on, we assume you have a database server up and running, and that you have permission to make administrative changes to the database.

Note that the database server may be the same machine as your AimSniff collector. While these instructions treat the AimSniff machine and the database server as logically separate components, they may reside on the same machine.

1.3.2 Using MySQL as the Backend

If you're using MySQL, you'll need to install Perl support for that database. Run the following commands to install the correct packages:

```
apt-get install libdbd-mysql-perl libdbd-mysql
```

Next, ensure that you have downloaded the correct version of AimSniff. You need the stock (unpatched) version of AimSniff, which can be obtained from:

http://www.aimsniff.com/releases/aimsniff-0.9d.tar.gz

Unpack the distribution, and move into the source directory.

You'll need to perform three steps to prepare the database for AimSniff:

1. Create the actual database by running this command on your database server:

   ```
   mysqladmin create aimsniff
   ```

2. Import the table structure file called `table.struct` from the AimSniff source distribution. If your database server is not on the same machine as the file, you may need to provide additional arguments (or copy the struct file onto the database server and run the command there). A sample local command would be:

   ```
   mysql aimsniff < table.struct
   ```

3. Finally, grant access privileges for a new MySQL user to modify the database. Log into MySQL as the `root` user and issue something like the following:

   ```
   GRANT ALL ON aimsniff.* TO username@hostname IDENTIFIED BY 'password';
   ```

   You should pick a username and password to use. Remember these values; we'll need them later to configure AimSniff. The hostname should be the name or IP address of the computer that will be running the sniffer. If this is the same machine as the database server, use `localhost` or `127.0.0.1`.

To test your setup, try to connect from the collector to the database server using the standard MySQL tools:

```
mysql -u username -p -h hostname
```

Be sure to substitute the correct username and hostname for the database server. You will be prompted to enter a password; use the one you selected earlier.

If the connection is sucessful, you're ready to move on to the next step, and configure AimSniff to run on your machine.

1.3.3 Using PostgreSQL as the Backend

If you're using PostgreSQL, you'll need to install Perl support for that database. Run the following commands to install the correct packages:

```
apt-get install libdbd-pg-perl libdbd-pgsql
```

Next, ensure that you have downloaded the correct version of AimSniff. You need our specially patched version, which can be obtained from our `aimsniff-postgres` repository.

You'll need to perform four steps to prepare the database for AimSniff:

1. Create a user who will own and access the database. On the database server, run the following command:

   ```
   createuser -A -D -P -E aimsniff
   ```

   (You may choose a different username if you wish.)

   You will be prompted to enter a password for this user.

2. Create a database to hold the AimSniff data, and assign ownership to the user you just created:

   ```
   createdb -O aimsniff aimsniff "Database for AimSniff collection data"
   ```

   The username is the first `aimsniff`; the database name is the second. Again, you may use whatever names you see fit.

3. Finally, load in the table structure for AimSniff. It's contained in the file `postgres.struct` from our patched distribution.

   ```
   psql -U user -d database -h server -f postgres.struct
   ```

   As always, susbstitute your values for name, database, and server.

4. You may need to modify your database server's `pg_hba.conf` file to allow database access from your collector machine. If you're on a local machine, this shouldn't be a problem, but remote machines may not be granted remote access over IP by default. Consult the PostgreSQL documentation for more information on granting host access.

At this point, you should be able to connect to the database server from the collector machine. If you have the `postgres-client` package installed on the collector, you should be able to connect to the database using the following:

```
psql -U user -d database -h server
```

Substituting the correct values as before. If it works, you're ready to move on to the next step!

1.4 Configuring AimSniff

At this point, you should be able to start AimSniff on the command line, and you should have your backend database set up correctly.

We're now ready to configure AimSniff, set it to start up on your collector, and configure your switching equipment to forward traffic to the collector.

1.4.1 AimSniff Configuration File

Find the file `aimsniff.config` (stock distribution) or `aimSniff.cfg` (Postgres distribution). Copy this file onto your collector, and store it with your other configuration files (`/etc/` might be a good spot for it).

Edit the file to include the correct parameters for your setup. Be sure to customize the database settings.

1.4.2 AimSniff Startup Script

Find the file `rc.aimsniff` (stock distribution) or `init.d.aimsniff` (Postgres distribution). Copy this file onto your collector, and store it with your other startup scripts (under Debian, `/etc/init.d/aimsniff` should be the final name of the file).

If you wish, you may set up AimSniff to lauch when the computer starts up. On Debian, you can do this by executing the following:

```
update-rc.d aimsniff defaults
```

1.4.3 Configuring Network Equipment

In order for your collector to sniff any traffic, it must be located on the network in such a way that all internet traffic passes to it.

The simplest way to do this is to attache the collector to a hub (or other repeater) on the segment with the traffic you wish to monitor.

For switched environments, however, you may need to configure your switching equipment to forward packets to your collector port in order to analyze them.

For example, the following commands are the ones we use on our Cisco 4507R core switch:

```
monitor session 1 source interface Gi6/1 both
monitor session 1 destination interface Gi6/2
```

That tells the switch to mirror all traffic **from** port Gi6/1 **to** port Gi6/2 (where our AimSniffer sits).

At this point, you should start seeing traffic on the interface. Use a utility like `tcpdump` to confirm that you're seeing the mirrored traffic.

1.4.4 Testing

You should now be ready to monitor traffic. Start the collector service, and check to see if any messages show up in the database (select from the `logs` or `handles` tables for quickest results).