

SSNiper SSN Scanner

Introduction

The SSNiper SSN scanner has been developed with Linux/UNIX systems in mind. Its purpose is to scan filesystem hierarchies for files that "may" have SSNs in them. As might be expected, this is a very false-positive-laden process. I have done what I can think of to limit these false positives without throwing too much data away. If there are any suggestions for other means, please feel free to [contact me](#).

- [Introduction](#)
- [Getting it](#)
 - [Beta Release](#)
 - [Changes](#)
- [Installation](#)
 - [RPM Systems](#)
 - [Binary tarball](#)
 - [Source tarball](#)
 - [Linux](#)
 - [Solaris \(probably other UNIXes too\)](#)
 - [IRIX](#)
 - [Post Install](#)
 - [Install / Use as non-root user](#)
- [Running SSNiper](#)
 - [Usage Statement](#)
 - [Sample Usage](#)
- [Report Review](#)
 - [Use of `ssnipер-report.pl`](#)
 - [Database Scanning](#)



SSNiper is brand new software, so expect to find some bugs or rough corners. Please contact [the maintainer](#) with bugs, suggestions, improvements, criticism, preferences, etc.

Getting it

The current version of SSNiper is 0.9.7, Release 2:

Distribution type	URL	Comments
Source tarball	ssnipер-0.9.7-2.tar.gz	experimental db-review, fixed <code>ssnipер-report.pl</code>
Linux i386 RPM	ssnipер-0.9.7-2.i386.rpm	experimental db-review, fixed <code>ssnipер-report.pl</code>
Linux i386 static binary tarball	ssnipер-0.9.7-2.i386.linux.bin.tar.gz	experimental db-review
Solaris SPARC generic static binary	ssnipер-0.9.4-2.sparc-generic.bin.tar.gz	better tested
Mac OS X x86 binary	ssnipер-0.9.5-1.i386.mac.bin.tar.gz	<ul style="list-style-type: none">• no libmagic (not all Mac OS Xs have the shared lib)• dynamically linked (Apple doesn't allow static link)• installs in <code>/usr/local</code>

Beta Release

Here are release versions that represent changes that need to be tested and confirmed before inclusion in the list above.

Distribution type	URL	Comments
Source Tarball	ssnipер-0.9.4-4.tar.gz	New state machine for identifying SSNs (thanks to Mike Hallock for the patch!)

Changes

Version	Date	Comments
0.9.7-2	Dec 3, 2009	<ul style="list-style-type: none">• Fixed bug in ssniper-report.pl handling filenames with parentheses• Incorporated experimental database (SQLite-based) support for results• Incorporated experimental database review functionality ('-p' flag)
0.9.5-1	Jan 10, 2008	<ul style="list-style-type: none">• moved Mike Hallock's modified FSM to production• Fixed SSN validation bug reported by Nathan VanHoudnos (some SSN ranges were not being validated properly)
0.9.4-2	Nov 28, 2007	<ul style="list-style-type: none">• added "skip list" of files/directories to skip from text file• fixed bunch of compile problems on Solaris• fixed timestamp output error on Solaris (older strftime)• fixed install script to be more modular
0.9.3-4	Nov 16, 2007	Fixed error recovery on broken gzip/bzip files / added more debug output (thanks Andy Wettstein)
0.9.3-3	Nov 14, 2007	Added checking for config file in CWD, added error messages for missing config files (thanks Allan Tuchman)
0.9.3-2	Nov 13, 2007	Fixed problem in ssniper-reports.pl with spaces in filenames

Binary distributions are statically compiled, so as long as you have a good magic file somewhere, it should pretty much just work.

In the version numbers, the release (the #-# at the end) indicates patches and bug fixes. Changes in the other digits indicate addition of features in order of magnitude.

Installation

RPM Systems

First, obtain the RPM above. Then install it as follows:

```
$ rpm -ivh ssniper-0.9.3-2.i386.rpm
```

Binary tarball

First, obtain the tarball above. Then install it as follows:

```
$ tar xzvf ssniper-0.9.3-2.i386.linux.bin.tar.gz
$ cd ssniper-0.9.3-2
$ ./install.sh
```

Source tarball

Linux

SSNiper is packaged with GNU autotools, so you can use the standard configure salute:

```
$ ./configure
..
$ make && make install
```

In case you're on a strange system (for now, such as Mac OS X), you may have to disable some of the features. So far, the following modules can be disabled with parameters to configure:

--without-magic	Don't use libmagic to vet filetypes
--without-zlib	Don't bind to zlib for looking through gzips
--without-bzip	Don't bind to libbz2 for bziped files
--with-static	Link executable statically (doesn't work on Macs)

Solaris (probably other UNIXes too)

You will probably need to disable the bzip and magic packages to compile on Solaris. At some point, I may try to get more detailed checking into the `configure` process so this will be automated:

```
$ ./configure --without-magic --without-bzip
```

For the generic SPARC binary tarball above, I use this line:

```
$ CC="/opt/SUNWsprow/bin/cc" CFLAGS=" -xarch=generic -dn -xCC " \
LDLFLAGS=" <location of static libararies> " \
./configure --without-magic --without-bzip \
--sysconfdir=/etc/ssniper --prefix=/usr
```

IRIX

I have received a report from Mike Hallock that he was able to compile SSNIper on IRIX with the MIPSpro compiler.

Post Install

Be sure to review the configuration file `/etc/ssniper/ssniper.conf`. Of significant importance is to confirm the location of your `magic number file`. This will be used by SSNIper to determine filetypes (to eliminate false positives). On Red Hat systems, this should be set to `/usr/share/file/magic`. Other distributions may vary.

Install / Use as non-root user

You may not wish to install SSNIper for the whole system. Or, perhaps you do not have root access, but want to scan some filesystem locations anyway. You can use the binary distribution tarballs linked above. SSNIper, as of 0.9.3-3, will look for a config file in either the pre-compiled location (usually `/usr/local/etc/ssniper.conf` or something similar) and in the current working directory. So, you should just be able to untar the binary distribution and run it in the directory where it was unarchived.

Alternatively, if for some reason SSNIper can't find the config file, you can override the config file location with the `-c` flag. I.e.:

```
user@host /tmp/ssniper-0.9.3-1 $ ./ssniper -c ./ssniper.conf /path/to/filesystem
```

When cleaning the report output, you will also need to specify the path to the `ssniper-report.pl` script. E.g.:

```
user@host /tmp/ssniper-0.9.3-1 $ cat ssniper_results.log | perl ./ssniper-report.pl > report_file.txt
```

Running SSNIper

Usage Statement

Here is the usage statement for SSNIper:

```
SSNIper 0.9.3: SSN scanner (maintainer: Joshua Stone, josh@uiuc.edu)
Copyright (C) 2007 The Board of Trustees of the University of Illinois
-----
(License: University of Illinois/NCSA Open Source License)

usage: ssniper [-hmnclzb] <dir> [<dir>*]

<dir>          Directory to scan
-h            This help output
-m <magic>    Override location of magic file
-n <name>     Name prefix on output files*
-c <conf>    Configuration file (default ./ssniper.conf)
-l <bytes>   Limit file scanning cap to first <bytes> bytes
-z          Skip gzip files (if compiled in)
-b          Skip bzip files (if compiled in)
```

Sample Usage

Sample usage would be:

```
ssniper /home /opt /usr/local /scratch
```

This will cause SSNIper to scan the provided directories for files with possible SSNs. The results will be output to three files:

output file	purpose
ssniper_results.log	SSN scan results – a list of files that may contain SSNs
ssniper_info.log	Scan information, such as configuration info, progress / timing, etc.
ssniper_debug.log	Debug information, such as the file skip list and errors encountered

You can change this output file name with the `-n` switch. E.g.: `ssniper -n foo /path/to/files` will name the output files `foo_results.log`, `foo_info.log`, `foo_debug.log`. This is useful for running multiple scans without overwriting the scan results (or having to move them when you're done).

Report Review

Use of `ssniper-report.pl`

The `ssniper_results.log` file is fairly dense. I have provided a followup Perl script that will take this file and condense it to a more human-readable report. This can be done as follows:

```
$ cat ssniper_results.log | ssniper-report.pl > report_file.txt
```

Note: if you have not "installed" SSNIper, and are instead running out of the untarred directory from a binary distribution (or have not done a `make install` when compiling from source), you will need to adjust the above line accordingly (i.e., pipe it through `perl ./ssniper-report.pl` instead).

The resulting report will categorize the results so that they will be easier to review by hand. The results will be categorized as follows:

Category	Meaning	Interpretation
High Risk, non-delimited and delimited	Files with many hits of both delimited and non-delimited 9-digit numbers	Review manually

High Risk, delimited	Files with many hits of delimited SSNs /only/	Review manually
High Risk, non-delimited	Files with many hits of non-delimited SSNs /only/	Review manually
Low Risk, non-delimited and delimited	Files with few hits of both delimited and non-delimited SSNs	Cursory review
Low Risk, delimited	Files with few hits of delimited SSNs /only/	Cursory review
Low Risk, non-delimited	Files with few hits of non-delimited SSNs /only/	Cursory review

Database Scanning

There is currently experimental support for storing SSNIper results in a database. The intent is that this should support running SSNIper "regularly", as with a `cron` job or similar, and performing a regular "review" of hits. The features for this mode are:

- Unattended scanning (quiet output suitable for `cron`)
- Hit tracking
- Marking of false positives
- Interactive review with '-p' flag
- Report emailing when new hits are identified

To configure this form of scanning, add one or more `root` clauses to your config file (probably `/usr/local/etc/ssnipiper/ssnipiper.conf` or `/etc/ssnipiper/ssnipiper.conf`) for what roots you wish to scan. E.g.:

```
root "/var/www"  
root "/home"
```

You may also configure an `email` clause in the config file to designate the recipient of hit reports:

```
email "josh@example.com"
```

Once thusly configured, you may run SSNIper with the '-d' flag to enable quiet database mode. This flag requires an argument to designate the location of the database file. E.g.:

```
$ ssnipiper -d /var/lib/ssnipiper.db
```

SSNIper will then scan the roots indicated in the config file and store its results in the indicated database file.

To review these hits, run SSNIper in database review mode with the '-p' flag. This flag also takes a filepath argument to designate the result database. E.g.:

```
$ ssnipiper -d /var/lib/ssnipiper.db
```

The user will be prompted interactively for each unknown or true-positive entry in the results database.