ITSPA SNITCH

Setting up your own Honeypot

Document Version: V1.0
Date: August 2016

ITSPA SNITCH

ITSPA SNITCH (Suspicious Numbering In Telephone Call Handling) is a database designed to handle information gathered from suspicious activity on VoIP systems. The database currently holds source IP addresses of suspicious SIP requests and dialled numbers (B numbers) from some known fraudulent calls.

The data in SNITCH is gathered from a number of sources including information collected from operational systems run by ITSPA members and a number of Honeypots. Honeypots are Internet connected systems with minimal security controls, designed specifically to attract attacks. Data collected from Honeypots can be fed directly into SNITCH.

Roll-your-own SNITCH Honeypot

The SNITCH project includes a simple application which will turn a basic Asterisk installation into a Honeypot. This application is available as source code and should build on any Linux system. It should also build on other Unix systems such as FreeBSD and MacOS although these have not been tested.

The application, named *honeypot*, works by passively listening to all SIP traffic sent to an Asterisk PBX and reporting the source IP address and other information of each new request in real-time to SNITCH. This approach is taken because the logging provided by Asterisk does not capture all of the information needed by SNITCH. The *honeypot* application currently reports the following SIP methods,

- OPTIONS
- REGISTER
- INVITE
- CANCEL
- SUBSCRIBE
- NOTIFY
- INFO
- MESSAGE

Other SIP methods could be added but tests have shown that the first 3 methods account for virtually all attacks.

Note that dialled numbers (B numbers) are included in the data sent to SNITCH, but SNITCH does not add these numbers to the database as many of the B numbers used in fraudulent calls are legitimate numbers.

The honeypot application will normally run on the same system as the Asterisk PBX, however it can also run on a separate system. To run *honeypot* on a separate system, that system must be able to monitor all traffic sent to the Asterisk PBX. For example it could be connected to a switch span port.

The Asterisk PBX must be dedicated to operating as a honeypot target as the *honeypot* application assumes that all SIP requests sent to the target are fraudulent and will report them

to SNITCH. The *honeypot* source distribution includes two asterisk configuration files which will enable any newly installed Asterisk system to run as a honeypot target. It goes without saying that there should be no SIP trunk or other PSTN connections on the Asterisk system.

Downloading the *honeypot* Souce

The source for the *honeypot* application is located at:

http://software.um-labs.net/snitch-honeypot.tar

For validation, the sha1 checksum is:

```
fce58e105f99b14498774b5ecf3cd327fe0b55ec  snitch-honeypot.tar
```

Download the source on to a suitable Linux platform with a full development system and untar the file with:

```
tar xvf snitch-honeypot.tar
```

Build Prerequisites

The *honeypot* application depends on the following libraries which must be installed prior to building the application.

- libpcap, packet capture library
- libosip, SIP parser
- libcurl, web server interaction

On CentOS systems you can install these libraries with the following command:

```
yum install libpcap-devel libosip-devel libcurl-devel
```

Other Linux distributions may differ.

Building and Installing *honeypot*

To build *honeypot*, change directory into the honeypot directory which was created when the source was downloaded and un-tar'ed and type:

```
make
```

If the application builds successfully switch to a privileged user identity with su or sudo and type:

```
make install
```

This will install *honeypot* in /usr/local/bin.

Running *honeypot*

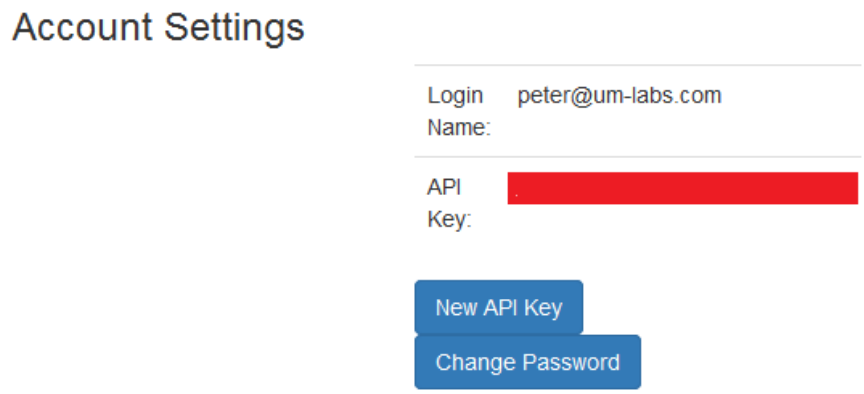
The *honeypot* application takes a number of argument, running ./honeypot in the build directory gives a complete list.

```
~/src/honeypot $ ./honeypot
Usage: honeypot options...
```

```
where options are
--apikey    key          user's APIKEY (mandatory)
--rblrhost  hostname     RBL host (defaults to rblportal.um-labs.net)
--monitorip ipaddr       IP address to monitor (mandatory)
--interval  minutes      Max interval between reports
--reportmax count        Max number of incidents in a single report
--interface iface-name   network interface to monitor
--listif                 list available interfaces and exit
--foreground             run in foreground
--apilog                 enable API logging when running in foreground
--log       filename     log to file (default is syslog)
```

Option	Status	Description
--apikey *key*	Mandatory	Your SNITCH API key which is used to identity and authenticate each incident report made by *honeypot*
--rblhost *hostname*	Mandatory	The hostname of the SNITCH server, currently snitch.um-labs.com
--monitorip *ipaddress*	Mandatory	The IP address of the Asterisk PBX used as a honeypot target. Correctly specifying this address avoids reporting any legitimate traffic on your network.
--interval *minutes*	Not yet implemented	This parameter batches reports submitting any batched reports after the specified period. Early tests have shown that this is probably not necessary, this option may be removed.
--reportmax *count*	Not yet implemented	This parameter batches reports submitting all batched reports when count is reached. Early tests have shown that this is probably not necessary, this option may be removed.
--interface *name*	Mandatory	The local network interface name to be use for monitoring SIP traffic (see below).
--listif	Optional	List available network interfaces and exit.
--foreground	Optional	Force *honeypot* to run in foreground.
--log *filename*	Optional	Output diagnostic log to a named file, default is to use syslog
--apilog	Optional	Display the responses from SNITCH each time a report is submitted. This option generates a lot of output and takes effect only if --foreground is specified.

Your SNITCH API key is shown on the My Account page of the SNITCH web interface, for example:



When running *honeypot* it is important to ensure that you specify the correct network interface with the --interface option, particularly if the host system has multiple interfaces. If *honeypot* is run on the same system as the Asterisk PBX, then the network interface used to receive external SIP traffic must be used. If *honeypot* is run on a separate system, then the interface connected to the same network as the Asterisk server must be used. You can get a list of local interfaces and their IP addresses by running *honeypot --listif*.

```
[root@honeypot honeypot]# ./honeypot --listif
[0]     eth0 (145.128.20.133)
[1]     eth1 (192.168.145.9)
[2]     lo (127.0.0.1)
```

You can use either the interface name (eth0) or the index in the list (0) with the --interface option. As *honeypot* is based on a packet sniffer, it must be run from a privileged account, normally root.

The *honeypot* source distribution includes a sample script, *run_honeypot*, which can be used to start *honeypot* with the correct arguments. This script must be modified before use. Parameters that must be changed are highlighted in red below.

```
#!/bin/sh
#
# Run the honeypot
#
# Usage: run_honeypot start | stop
#
# This script should be copied to /etc/init.d, /etc/local.d or equivalent
# location to ensure that it is autostarted  during system boot.
# Some changes may be needed to enable this script on your system.
# Refer to local documentation
#
APIKEY='your-api-key'           # Add your API key here (use single quotes)
APIHOST=snitch.um-labs.net      # SNITCH hostname
MONITORIP="192.168.19.71"       # IP Address of target PBX
INTERVAL=10                     # Not yet implemented
MAXREPORTS=100                  # Not yet implemented
```

```
INTERFACE=eth0                  # Set your own interface name
#DEBUG_OPTIONS="--foreground --apilog"   # Run in foreground with debugging
```

Once this script has been customised, start *honeypot* with:

```
./run_honeypot start
```

As *honeypot* defaults to running in background, *run_honeypot* will return to the command shell.

Diagnostics and Logging

When *honeypot* is first installed it is recommended that it is run with the --foreground option. This will force the application to run in the foreground and log to standard output. The default is for *honeypot* to run in background and log via syslog.

When *honeypot* is running it will log each SIP request received and report it to SNITCH. Each log entry captured by syslog will be similar to:

```
Aug 14 10:45:11 honeypot honeypot: 1471164311 Received INVITE,
      IP 62.210.26.82, From sip:7010@145.128.20.133,
      To sip:910972597657476@145.128.20.133, UA sipcli/v1.8
```

Following the honeypot: field which identifies the reporting application, the log entry includes the following:

- 1471163944, the event timestamp in seconds since 00:00:00 on the 1st January 1970.
- Received INVITE, the SIP method in this request
- IP 62.120.26.82, the source IP of this request
- From sip:7010@145.128.20.133, the From URI used in the request
- To sip:910972597657476@145.128.20.133, the To URI used in the request
- UA sipcli/v1.8, the user agent header used in the request

This SIP INVITE is shown on the Recently Reported IPs page of the SNITCH GUI. The GUI shows all times as UTC. The honeypot which captured this SIP INVITE request is running in the CEST time zone, hence the two hour difference.

Recently Reported IPs

IP Address	Location	Reason	Reported by	Incident Date
62.210.26.82	France, , Paris (8th arrondissement of Paris)	SIP INVITE	UMLABS	14 Aug 2016 08:44
62.210.26.82	France, , Paris (8th arrondissement of Paris)	SIP INVITE	UMLABS	14 Aug 2016 08:44
62.210.26.82	France, , Paris (8th arrondissement of Paris)	SIP INVITE	UMLABS	14 Aug 2016 08:44

If *honeypot* fails to report an event to SNITCH, then an error will be logged. The most likely error is an invalid API key which is logged as follows:

```
Aug 14 10:03:42 takeback honeypot[3824]: ERROR, API Call returns 403
(Forbidden), REGISTER from 89.163.204.2
```

This log message shows that a SIP REGISTER report from 89.163.204.2 was rejected by SNITCH.

Verbose logs which include the response returned from SNITCH for each report can be enabled by adding the --apikey to the command line.

```
<?xml version="1.0"?>
<addresslist>
  <address>
    <IPAddress>89.163.204.2</IPAddress>
    <reason>SIP REGISTER</reason>
    <source>UMLABS</source>
    <incidentdate>1471168780</incidentdate>
    <country>Germany</country>
    <region>BY</region>
    <regionname/>
    <city>Coburg</city>
  </address>
</addresslist>
Aug 14 10:59:43: 1471168780 Received REGISTER, IP 89.163.204.2,
   From sip:0007@80.195.188.117, To sip:0007@80.195.188.117, UA –
```

Asterisk Dialplan

The *honeypot* distribution includes a simple Asterisk dialplan which will respond to all SIP requests and answer any INVITE request. This dialplan is implemented in sip.conf and extensions.conf. To use this dialplan simply copy these files into your Asterisk configuration directory, normally /etc/asterisk. The sip.conf configuration file sets the *domain* parameter to um-labs.conf. It is recommended but not essential to change this to some other value. This can be your own domain or a made up domain. Sip.conf sets *allowguest* to yes which means requests to any SIP URL will be accepted.

This dialplan has been tested on Asterisk 11.21.1 an Asterisk 13.9.1, but should work on any other version.