rtpbreak

February 18, 2014 Sniffing/Spoofing

rtpbreak Package Description

With rtpbreak you can detect, reconstruct and analyze any RTP session. It doesn't require the presence of RTCP packets and works independently form the used signaling protocol (SIP, H.323, SCCP, ...). The input is a sequence of packets, the output is a set of files you can use as input for other tools (wireshark/tshark, sox, grep/awk/cut/ cat/sed, ...). It supports also wireless (AP_DLT_IEEE802_11) networks.

- reconstruct any RTP stream with an unknown or unsupported signaling protocol
- reconstruct any RTP stream in wireless networks, while doing channel hopping (VoIP activity detector)
- reconstruct and decode any RTP stream in batch mode (with sox, asterisk, ...)
- reconstruct any already existing RTP stream
- reorder the packets of any RTP stream for later analysis (with tshark, wireshark, ...)
- build a tiny wireless VoIP tapping system in a single chip Linux unit
- build a complete VoIP tapping system (rtpbreak would be just the RTP dissector module!)

Source: rtpbreak Documentation
rtpbreak Homepage | Kali rtpbreak Repo

- Author: Dallachiesa Michele
- License: GPLv2

Tools included in the rtpbreak package

rtpbreak – Detects, reconstructs, and analyzes RTP sessions

root@kali:~# rtpbreak -h
Copyright (c) 2007-2008 Dallachiesa Michele <micheleDOTdallachiesaATposteDOTit>
rtpbreak v1.3a is free software, covered by the GNU General Public License.

USAGE: rtpbreak (-r|-i) <source> [options]

INPUT

-r <str> Read packets from pcap file <str>
-i <str> Read packets from network interface <str>
-L <int> Force datalink header length == <int> bytes

```
OUTPUT

 -d <str>      Set output directory to <str> (def:.)
 -w            Disable RTP raw dumps
 -W            Disable RTP pcap dumps
 -g            Fill gaps in RTP raw dumps (caused by lost packets)
 -n            Dump noise packets
 -f            Disable stdout logging
 -F            Enable syslog logging
 -v            Be verbose

SELECT

 -m            Sniff packets in promisc mode
 -p <str>      Add pcap filter <str>
 -e            Expect even destination UDP port
 -u            Expect unprivileged source/destination UDP ports (>1024)
 -y <int>      Expect RTP payload type == <int>
 -l <int>      Expect RTP payload length == <int> bytes
 -t <float>    Set packet timeout to <float> seconds (def:10.00)
 -T <float>    Set pattern timeout to <float> seconds (def:0.25)
 -P <int>      Set pattern packets count to <int> (def:5)

EXECUTION

 -Z <str>      Run as user <str>
 -D            Run in background (option -f implicit)

MISC

 -k            List known RTP payload types
 -h            This
```

rtpbreak Usage Example

Analyze RTP traffic using interface eth0 ***(-i eth0)***, fill in gaps ***(-g)***, sniff in promiscuous mode ***(-m)***, and save to the given directory ***(-d rtplog)***:

```
root@kali:~# rtpbreak -i eth0 -g -m -d rtplog
 + rtpbreak v1.3a running here!
 + pid: 10951, date/time: 17/05/2014#13:40:02
 + Configuration
   + INPUT
     Packet source: iface 'eth0'
```

```
    Force datalink header length: disabled
  + OUTPUT
    Output directory: 'rtplog'
    RTP raw dumps: enabled
    RTP pcap dumps: enabled
    Fill gaps: enabled
    Dump noise: disabled
    Logfile: 'rtplog/rtp.0.txt'
    Logging to stdout: enabled
    Logging to syslog: disabled
    Be verbose: disabled
  + SELECT
    Sniff packets in promisc mode: enabled
    Add pcap filter: disabled
    Expecting even destination UDP port: disabled
    Expecting unprivileged source/destination UDP ports: disabled
    Expecting RTP payload type: any
    Expecting RTP payload length: any
    Packet timeout: 10.00 seconds
    Pattern timeout: 0.25 seconds
    Pattern packets: 5
  + EXECUTION
    Running as user/group: root/root
    Running daemonized: disabled
 * You can dump stats sending me a SIGUSR2 signal
 * Reading packets...
```

spoofing, voip

Related Posts

Cookie Cadger February 16, 2014

hamster-sidejack February 18, 2014

Spooftooph February 18, 2014