If getting the error “config: error: cannot find a suitable libfixbuf (>= 2.3.0)”

1. install and configure Ubuntu (or similiar VMWare app)
2. Download SiLK from instructor provided link
3. Open command prompt
4. $ sudo apt-get install libpcap-dev
5. Change to directory where SiLK was downloaded in Step 1.
6. $ ./configure --prefix=/usr --sysconfdir=/etc/silk --enable-data-rootdir=/netflow --enable-ipv6 --enable-output-compression
7. $ make
8. $ sudo make install

Additional link: https://github.com/bbayles/netsa-pkg/tree/master/yaf-src

Use this link to help with the SiLK command syntax, https://tools.netsa.cert.org/silk/silk-reference-guide.html Gives you each command and the syntax breakdown

Building
========
YAF requires glib 2.4.7 or later; glib is available at
http://www.gtk.org. Build and install glib before building YAF. Note that
glib is also included in many operating environments or ports collections.

YAF requires libfixbuf version 2.3.0 or later; libfixbuf is available at
http://tools.netsa.cert.org/fixbuf. Build and install libfixbuf before
building YAF.

Common Issues when Installing or Running YAF
==========
Configure Error: configure: error: Cannot find a suitable libfixbuf (>= 1.0.0)
(Try setting PKG_CONFIG_PATH): No package 'libfixbuf' found

Solution: export PKG_CONFIG_PATH=/usr/local/lib/pkgconfig
if libfixbuf was installed in the default location (Otherwise /$prefix/lib/pkgconfig)

Runtime Error: yaf: error while loading libraries: libairframe-2.3.0.so.4:
cannot open share object file: No such file or directory

Solution: Most likely yaf libraries were installed in a nonstandard location.
Try running `ldconfig` or setting LD_LIBRARY_PATH to the location of libairframe.

Runtime Error: "couldn't open library "dnsplugin": file not found"

Solution: Most likely yaf application labeling libraries were installed
in a nonstandard location (default: /usr/local/lib/yaf).
Set LTDL_LIBRARY_PATH to the location of those libraries ($prefix/lib/yaf).
If you are starting yaf via a startup script, it may be necessary to
export this environment variable from the startup script.

Error: yaf terminating on error: Failed to load certificate file: error:0906D06C:PEM routines:PEM_read_bio:no start line

Solution: When running yaf exporting via TLS, the certificate files given to
--tls-ca and --tls-cert must be in PEM format. DER format is not accepted.

Error: yaf terminating on error: Failed to load private key file: error:0906A068:PEM routines:PEM_do_header:bad password read

Solution: Most likely the key file given to --tls-key requires a password and
the YAF_TLS_PASS environment variable was not set. Set the YAF_TLS_PASS
environment variable to the correct password for the --tls-key or remove the
password from the key file (openssl rsa -in key.key -out key.key).

Error: yaf terminating on error: Failed to load private key file: error:06065064:digital envelope routines:EVP_DecryptFinal_ex:bad decrypt

Solution: Most likely the key file given to --tls-key requires a password and
the YAF_TLS_PASS environment variable was set to the incorrect password. Set
the YAF_TLS_PASS environment variable to the correct password or remove the
password from the key file (openssl rsa -in key.key -out key.key)

Additional links:

Cofiguring YAF with SiLK, https://tools.netsa.cert.org/yaf/libyaf/yaf_silk.html
SiLK Configuration sensor.conf, https://tools.netsa.cert.org/silk/sensor.conf.html
https://www.reddit.com/r/networking/comments/4ga817/using_yaf_to_dump_to_a_netflow_probe/

For the SiLK assignment you will mostly get TCP traffic since you are doing pings and traceroutes to url/websites (HTTP). Port udp is normally associated with ports 113, 135, 160 and 161. the website will be TCP since it is connection oriented protocol with three-way handshake.
normal commands to use would be :

rwfilter -aport=80 -proto=6 -type=all -pass=stdout | rwuniq - fields=sIP -values=bytes
sourceIP.txt
rwfliter -aport=80 -proto=6 -type=all -pass=stdout | rwuniq -fields=dIP - values=bytes
destIP.txt

rwfilter -aport=80 --proto=6 -type=all -pass=stdout | rwcut
all.txt
rwfliter -aport=80 --proto=6 -type=all -pass=stdout | rwuniq -fields=sIP -values=flows
uniqueSource.txt
rwfilter -aport=80 -proto=6 -type=all -pass=stdout | rwuniq -fields=dIP -values= flows
uniqueDest.txt
rwfilter -aport=80 -proto=6 -type=all -pass=stdout | rwuniq -fields=protocol- values=Distinct:dIP
uniqueProtocolsDistinctValues.txt

To get UDP you would have to change "80" to "160 or 161". If all you see is TCP traffic that is fine, no worries.

Example of output you should get from pings and traceroutes.

sIP| dIP|sPort|dPort|pkts| bytes|flags| sTime| 88.187.13.78|71.55.40.204|40936| 80| 83| 3512|FS PA|2010/12/08T11:00:01| 71.55.40.204|88.187.13.78| 80|40936| 84|104630|FS PA|2010/12/08T11:00:01| 88.187.13.78|71.55.40.204|40938| 80| 120| 4973|FS PA|2010/12/08T11:00:04| 71.55.40.204|88.187.13.78| 80|40938| 123|155795|FS PA|2010/12/08T11:00:05| 88.187.13.78|71.55.40.204|56172| 80| 84| 3553|FS PA|2010/12/08T12:00:02| 71.55.40.204|88.187.13.78| 80|56172| 83|103309|FS PA|2010/12/08T12:00:02| 88.187.13.78|71.55.40.204|56177| 80| 123| 5093|FS PA|2010/12/08T12:00:05| 71.55.40.204|88.187.13.78| 80|56177| 124|157116|FS PA|2010/12/08T12:00:05|

Install SiLK on an ubuntu 16.04 virtual machine running in VMWare Workstation Pro. Start with a clean install and then ensure that the following tools are installed and up to date:

- gcc
- gcc-c++
- glib2
- libpcap
- python

After these packages are installed then download and un-package the following files from https://tools.netsa.cert.org/index.html:

1. libfixbuf
2. YAF 2.11.0
3. SiLK 3.19.1

When SiLK is being configured, use the parameter "--with-libfixbuf=/usr/local/lib/pkgconfig/" to tell it where to find the libfixbuf package.

Create a silk.conf file and moved it to /etc/ld.so.conf.d/ with the following data:
/usr/local/lib
/usr/local/lib/silk

Create a sensors.conf file to give yaf the information on your network interface:
probe S0
ipfix listen-on-port 18001
protocol tcp
listen-as-host 127.0.0.1end probe
group my-network ipblocks 192.168.23.0/24
end group
sensor S0
ipfix-probes S0
internal-ipblocks @my-network
external-ipblocks remainder
end sensor

Information regarding a network interface can be found using the ifconfig tool: /var/log$
ifconfig result ens33

After getting the network interface information make some edits to yaf.conf: (below)

The highlighted parts are changed. The port number was defined above in the sensors.conf file and the name of the network interface was given with the ifconfig tool.

Then start the rwflowpack service with "sudo service rwflowpack start" and start the netflow capture with the following command:

sudo /usr/local/bin/yaf --silk --ipfix=tcp --live=pcap --out=127.0.0.1 --ipfix-po=18001 --in=ens33

Then check to make sure netflow entries were being captured

some more helpful hints:

If getting a "failed to start rwflowpack.service error"
mitigation and cause..

Short answer:

There is a version mismatch between SiLK and libfixbuf. Recompiling and reinstalling SiLK should fix the issue.

Long answer:

It appears that your SiLK installation is finding a different version of the libfixbuf library than the one it found when the source code was compiled. When SiLK was compiled, it found a 1.x version of libfixbuf that includes the function named fbSessionAddTemplateCtxCallback2. When you attempt to start rwflowpack now, it is finding a 2.x version of libfixbuf that does not have that function.

SiLK may be compiled against either version of libfixbuf, but the version of libfixbuf should not be changed once SiLK has been compiled.

If you have multiple versions of libfixbuf on your system, it could be that SiLK finds one the 1.x one when it is compiled and the 2.x one when it is invoked.

Troubles with rwflowpack starting the service. If it tells you that service doesn't exist- what will work is to add the permissions after moving it into ***init.d*** and change directory (cd) into their to add the executable permissions. From there, you can run the below commands to get the unique dip, sip, port, etc.

commands to run:

Syntax:
SIP; rwfilter -aport=80 -proto=6 -type=all -pass=stdout | rwuniq - fields=sIP -values=bytes
DIP; rwfliter -aport=80 -proto=6 -type=all -pass=stdout | rwuniq -fields=dIP - values=bytes
All traffic; rwfilter -aport=80 --proto=6 -type=all -pass=stdout | rwcut
unqiueSource: rwfliter -aport=80 --proto=6 -type=all -pass=stdout | rwuniq -fields=sIP -values=flows
unqiueDest. rwfilter -aport=80 -proto=6 -type=all -pass=stdout | rwuniq -fields=dIP -values=flows

Uniqueprotocolsdistinctvalues. rwfilter -aport=80 -proto=6 -type=all -pass=stdout | rwuniq -fields=protocol- values=Distinct:dIP

Error: rwflowpack: site configuration file not found

Solution: https://tools.netsa.cert.org/silk/rwflowpack.html
file not found error received, is most likely from the sensors.conf file, due to missed step of copying the file over properly

To fix a Yaf terminating connection refused error: try the following

Use the following link for building and configuring YAF+SiLk
https://tools.netsa.cert.org/yaf/libyaf/yaf_silk.html
The below parameters are derived from that tutorial.

YAF take a port in **YAF_IPFIX_PORT** to connect to the IPFIX collector on the specified port. So, YAF does not open any port with that number and does not listen to that port

So you can change the value of YAF_IPFIX_PORT= in **yaf.conf**, from **18000** to **18001** (the port which is defined for *listen-on-port* in *sensor.conf*)

-best to use a VM environment (e.g., VMWare Workstation Pro, or Ubuntu).
-Biggest issue is the Yaf install

- Watch for formatting as it relates to command blocks, and paragraph tags.
- Missing dependencies can be solved by doing a sudo apt install.

Difficulties in SILK collector setup

The YouTube video provided in the assignment is helpful. Follow those instructions carefully to set up Ubuntu VM correctly with all the NetSA tools [1]. Install SiLK 3.19.1 on an Ubuntu 18 VM. if running rwflowpack but receive some errors. Use link to walk through how to configure rwflowpack. Follow these instructions to change the rwflowpack.conf file so that it looks for configuration files in the /data directory [2].

1] "How to Install NetSA Tools on Ubuntu", YouTube, 2016. [Online]. Available: https://www.youtube.com/watch?v=5fLhb6EHvyw&list=PLSNlEg26NNpyBCKGeWaYWyuZPGfkaPWAM&index=1. [Accessed: 12- Sep- 2020].

[2] "Netflow on Nexus 1000v", SANS Internet Storm Center, 2020. [Online]. Available: https://isc.sans.edu/forums/diary/Netflow+on+Nexus+1000v/16865/. [Accessed: 12- Sep- 2020].

After that, create a sensors.conf file and a silk.conf file and place both of them in the /data directory. The sensors.conf file contains all the fields. The silk.conf file can be copied directly from the SiLK 3.19.1 package (in /site/generic/silk.conf).

The final thing to do before collecting data is to set the environment variable SILK_DATA_ROOTDIR=/data. Do this with the command 'export SILK_DATA_ROOTDIR=/data' [3].

From there, run the following commands to start rwflowpack and yaf [3,4].

[3] "YAF - Documentation", Tools.netsa.cert.org, 2020. [Online]. Available: https://tools.netsa.cert.org/yaf/libyaf/yaf_silk.html. [Accessed: 12- Sep- 2020].

[4] "SiLK — rwflowpack", Tools.netsa.cert.org, 2020. [Online]. Available: https://tools.netsa.cert.org/silk/rwflowpack.html. [Accessed: 12- Sep- 2020].

<u>More thoughts to help with Module 2's SiLK assignment:</u>

Follow the steps outlined in this link; https://tools.netsa.cert.org/silk/silk-on-box-deb.html

---

Issue:

If you get an error "job for rwflowpack.service failed because the control process exited with error code"

Answer:

Look at https://tools.netsa.cert.org/silk/silk-on-box-deb.html under "Setup SiLK". There is a section on running rwflowpack as a service.

Issue:

The first issue was with the data directory path and the second was an issue with it not being able to find the site config file.

Answer: Can do syntax of journalctl -xe | grep “rwflowpack”.

Issue:

When running "rwfilter --proto=0- --type=all --pass=stdout | rwcut | head", you will see data returned. However, Section 8 of the install guide can be confusing, and most of the IP commands say the option is unknown (route-cache, flow-cache, etc.).

Answer:

Can use the following script from this website here: https://gist.github.com/maka-io/e23c1b5f32d3a1ca84c10f21753ffe70 . Can help with running rwflowpack and finalizinig libfixbuf configurations.

You can pass the --with- libfixbuf=/usr/local/lib/pkgconfig/ argument to /configure for both rwflowpack and yaf. This will tell each of these services where the libfixbuf files are.

In terms of collecting the data, you are doing pings and traceroute, then navigating to the webpages to generate ingress and egress flows. Then taking screen shots to discuss findings & answer questions.

Example screen shots below.

Ping google.com

```
algorist@ubuntu:~$ date -u
Wed Sep 16 05:59:17 UTC 2020
algorist@ubuntu:~$ ping -c 12 google.com
PING google.com (216.58.194.174) 56(84) bytes of data.
64 bytes from sfo07s13-in-f14.1e100.net (216.58.194.174): icmp_seq=1 ttl=128 time=13.0 ms
64 bytes from sfo07s13-in-f14.1e100.net (216.58.194.174): icmp_seq=2 ttl=128 time=18.4 ms
64 bytes from sfo07s13-in-f14.1e100.net (216.58.194.174): icmp_seq=3 ttl=128 time=18.9 ms
64 bytes from sfo07s13-in-f14.1e100.net (216.58.194.174): icmp_seq=4 ttl=128 time=21.9 ms
64 bytes from sfo07s13-in-f14.1e100.net (216.58.194.174): icmp_seq=5 ttl=128 time=12.3 ms
64 bytes from sfo07s13-in-f14.1e100.net (216.58.194.174): icmp_seq=6 ttl=128 time=20.2 ms
64 bytes from sfo07s13-in-f14.1e100.net (216.58.194.174): icmp_seq=7 ttl=128 time=20.3 ms
64 bytes from sfo07s13-in-f14.1e100.net (216.58.194.174): icmp_seq=8 ttl=128 time=13.0 ms
64 bytes from sfo07s13-in-f14.1e100.net (216.58.194.174): icmp_seq=9 ttl=128 time=12.4 ms
64 bytes from sfo07s13-in-f14.1e100.net (216.58.194.174): icmp_seq=10 ttl=128 time=13.0 ms
64 bytes from sfo07s13-in-f14.1e100.net (216.58.194.174): icmp_seq=11 ttl=128 time=12.0 ms
64 bytes from sfo07s13-in-f14.1e100.net (216.58.194.174): icmp_seq=12 ttl=128 time=13.1 ms

--- google.com ping statistics ---
12 packets transmitted, 12 received, 0% packet loss, time 11020ms
rtt min/avg/max/mdev = 12.039/15.744/21.921/3.674 ms
algorist@ubuntu:~$ rwfilter --proto=1 --type=all --pass=stdout --stime=2020/09/16T05:59:00-2020/09/16T06:00:00 | rwuniq --fields=dIP,sTime,eTime,sport,proto --value
s=bytes,flows --sort-output | nl
     1                                  dIP|             sTime|             eTime|sPort|pro|              Bytes|   Records|
algorist@ubuntu:~$ rwfilter --proto=1 --type=all --pass=stdout --stime=2020/09/16T05:59:00-2020/09/16T06:00:00 | rwuniq --fields=dIP,sTime,eTime,sport,proto --value
s=bytes,flows --sort-output | nl
     1                                  dIP|             sTime|             eTime|sPort|pro|              Bytes|   Records|
     2                      192.168.180.138|2020/09/16T05:59:23|2020/09/16T05:59:34|    0|  1|                1008|         1|
     3                       216.58.194.174|2020/09/16T05:59:23|2020/09/16T05:59:34|    0|  1|                1008|         1|
```

Ingressing flow counts:

```
algorist@ubuntu:~$ rwfilter --sport=0-1024 --proto=6,17 --type=all --pass=stdout --stime=2020/09/16T06:32:00-2020/09/16T06:32:55 | r
wtotal --proto --skip-zero --summation
   protocol|        Records|              Bytes|          Packets|
           6|            169|            2838975|             5436|
          17|             31|               5208|               31|
      TOTALS|            200|            2844183|             5467|
```

Egressing flows:

```
algorist@ubuntu:~$ rwfilter --dport=0-1024 --proto=6,17 --type=all --pass=stdout --stime=2020/09/16T06:32:55-2020/09/16T06:33:40 | r
wuniq --fields=sIP,sTime,eTime,dport,proto --values=bytes,flows --bin-time=60 --sort-output | nl
     1                                  sIP|              sTime|             eTime|dPort|pro|              Bytes|    Records|
     2                      192.168.180.138|2020/09/16T06:33:00|2020/09/16T06:33:00|   53| 17|               1388|         18|
     3                      192.168.180.138|2020/09/16T06:33:00|2020/09/16T06:33:00|  443|  6|              75946|         55|
     4                      192.168.180.138|2020/09/16T06:33:00|2020/09/16T06:34:00|  443|  6|             196115|         12|
     5                      192.168.180.138|2020/09/16T06:33:00|2020/09/16T06:35:00|   80|  6|               2279|          2|
     6                      192.168.180.138|2020/09/16T06:33:00|2020/09/16T06:35:00|  443|  6|              80668|         17|
     7                      192.168.180.138|2020/09/16T06:33:00|2020/09/16T06:36:00|  443|  6|             118680|         28|
```