Knowm Inc. | Santa Fe, New Mexico, USA | Founded in 2015 | www.knowm.org | contact@knowm.org

The **Plug and Play Network Monitor** contains all the components you need to monitor network traffic on your local network. Simply tap into the network branch you wish to monitor and turn on the *Intel Celeron Fanless Mini Computer*. The installed open-source software is configured to automatically start up and collect data. Visit http://localhost:5601 to view the Kibana web interface and start exploring the data and creating a custom dashboard.

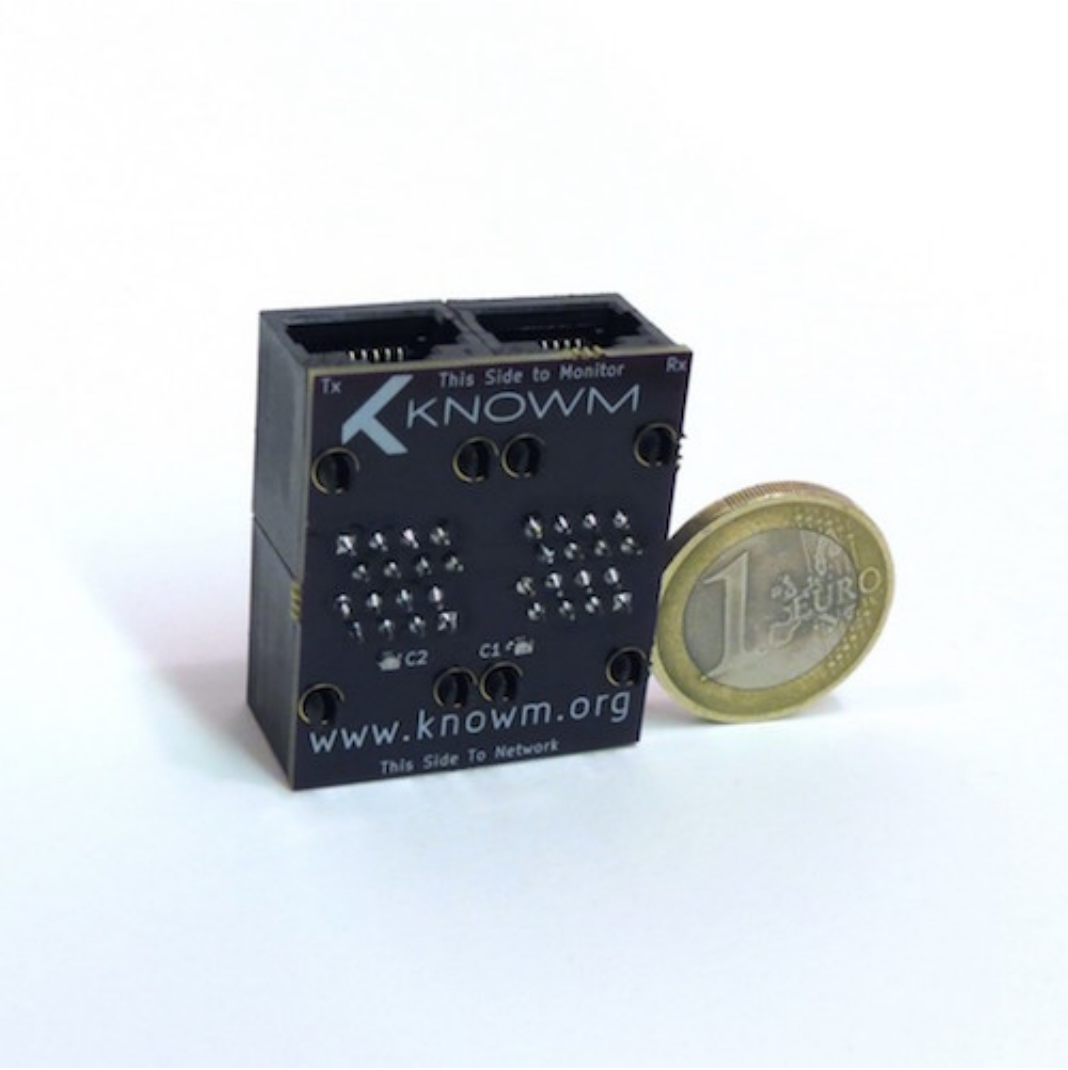
Knowm Fanless Computer

The *Intel Celeron Fanless Mini Computer* is an elegant and efficient fan-less mini-computer running Ubuntu desktop.

- Intel Dual Core 1.8Ghz Processor
- 8GB DDR3L RAM
- Dual Intel Gigabit Network Card
- Dual Band 300Mbps Wireless
- HDMI, VGA, LAN X 2, USB 2.0, USB 3.0 X 4, MIC Input, SPK Output
- VESA Mount (enables you to mount behind your monitor)
- PC Stand Included
- 12V external DC power supply
- Ubuntu 16.04 LTS Desktop

Passive LAN Tap

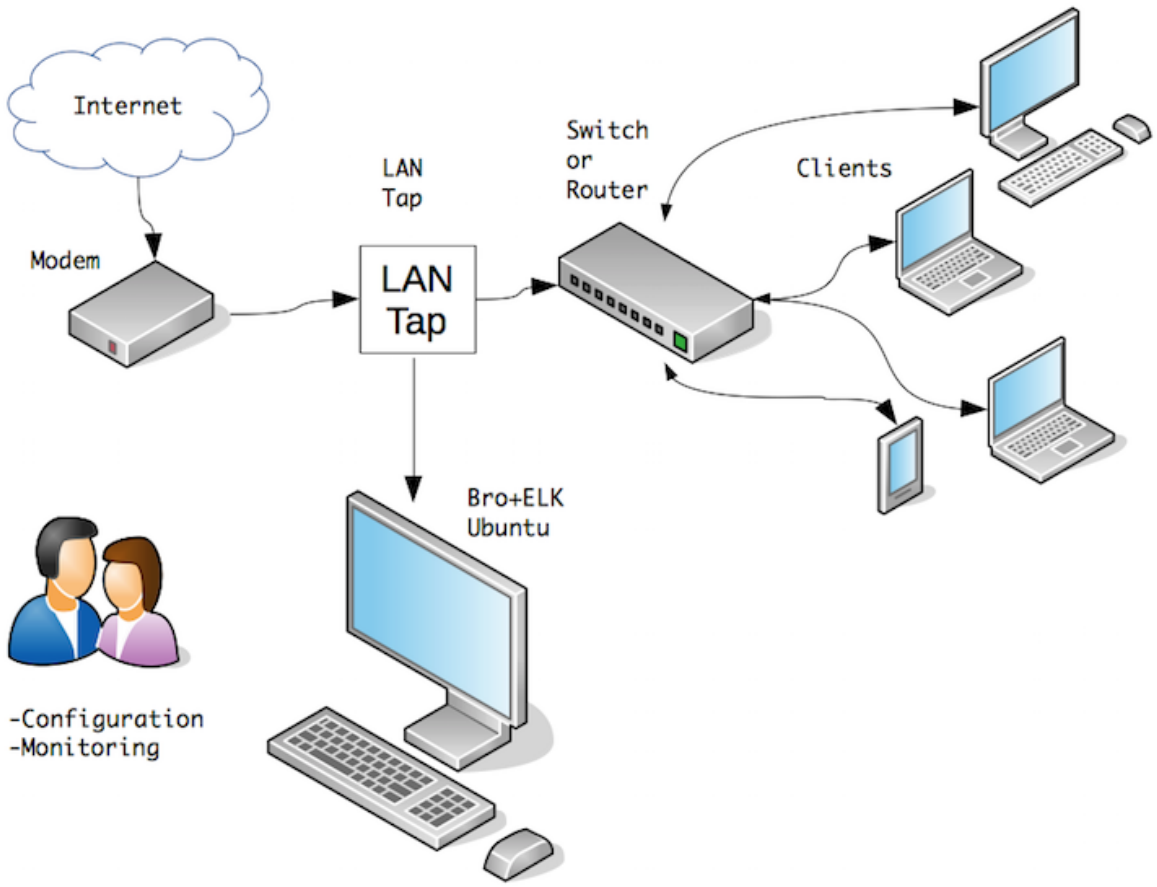
The Knowm Passive LAN Tap is a passive device for monitoring packet flow on an Ethernet connection. The Rx(receiving) and Tx(transmitting) ports are located on the same side of the LAN tap while the traffic ports are located on the opposite side. This configuration allows for a more compact and manageable component for your network. **The Knowm passive LAN Tap is designed to monitor 10 Base-T and 100 Base-TX networks. If installed on 1000 Base-T networks, the device will force switches and routers to transmit at 100Mbs.** If you must monitor networks that exceed 100Mbs, active (and much more expensive) equipment must be used.



Bro + ELK Stack

In general, there are two approaches to detecting cyber security attacks. In many cases, both techniques are applied concurrently. The first approach is a rules matching technique where individual network packets or session data are parsed and matched against a set of rules. These systems, while effective, require lots of manual tuning to match a particular network's character, i.e. weeding out rules that produce false positives. The second approach is a human-intuition-driven hands-on technique where a person is given a dashboard or dashboards filled with raw and/or processed data displaying an overview of what is currently happening on the network. The software usually provides tools for "digging down" into the data so that the person can carry out an iterative discovery process of suspicious or "anomalous" activity. This technique works well because the human brain is adept at searching, pattern recognition, discovery, and anomaly detection. The Bro+ELK software stack is emerging as the best open-source package for recording, searching and monitoring real-time network information.

The Bro network security monitor provides a comprehensive platform for general network traffic analysis. The result of 15 years of research, Bro is relied upon by thousands around the world for securing their cyber infrastructure. Bro's user community includes major universities, research labs, supercomputing centers, and open-science communities.



Cyber Security Edition

Elastic Search, Logstash and Kibana – together the ELK Stack – is emerging as the best software stack to collect, manage and visualize big data. The ELK stack is a flexible tool for searching, analyzing and monitoring data. When combined with Bro as a data source, the ELK stack provides flexible real-time (and beautiful!) views on network data.

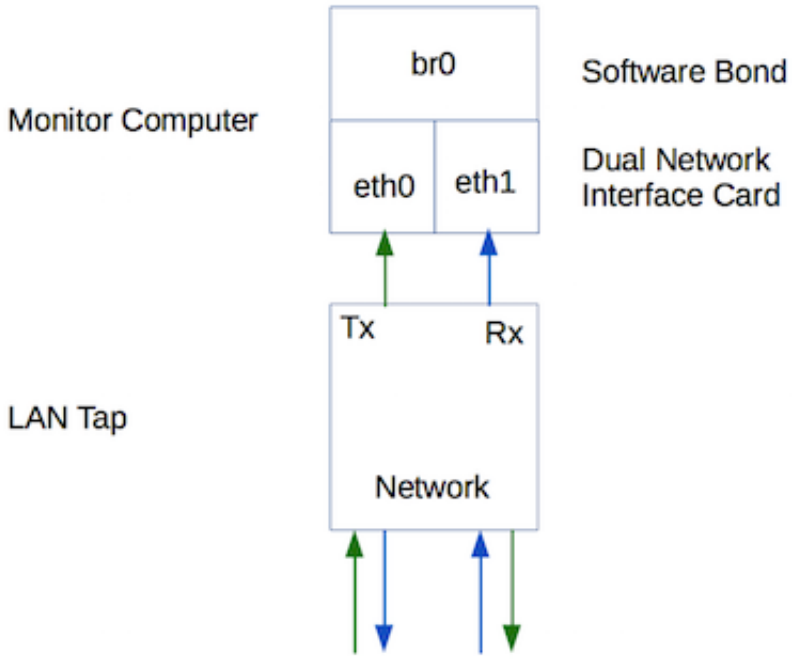
Technical Details

This section provides information about all the key technical parts, mostly software, which work together to turn raw packets going across the network into real-time plots in Kibana. All of the information is publically available on knowm.org and it referenced in the sections below.

Bonded Network Interface

source: How To Create a Bonded Network Interface

When using a passive or active LAN tap and not an “aggregator” or “mirror port” for monitoring network traffic, both the ‘incoming’ and ‘outgoing’ channels need to be bonded together for IDS systems such as Bro to process as a single network flow channel. This setup involves using a technique called *bonding* to take two physical interfaces and bond them together, creating a logical interface.



A network monitor machine has a minimum of two network interfaces which will be bonded into a single logical interface by software. Many times, a third NIC interface will be present on a monitor machine, which can be used for remote access (management port). Normally, we will use an integrated NIC port as the management port and a 3rd-party NIC with dual ports as the monitoring ports. The monitoring ports are connected to the LAN Tap and the packet flow is rejoined internally via port bonding (software-based). In the case of this **Plug and Play Network Monitor**, it has two integrated NICs which are bonded together and no monitoring port. The monitoring interface however still exists over the Wifi interface for remote access. Alternatively, the user can hook a monitor, keyboard and mouse directly up the the *Intel Celeron Fanless Mini Computer* and interact with the Network Monitor directly.

The bonded interface has the name `br0`. Using the command `bmon` in the terminal, you should be able to observe data flowing on `br0` which is the combination of the two physical monitor interfaces.

Bro

source: How to Install Bro Network Security Monitor on Ubuntu

Bro Network Security Monitor is an open source network monitoring framework. In a nutshell, Bro monitors packet flows over a network with a network tap installed with optional bonded network interfaces, and creates high-level “flow” events from them and stores the events as single tab-separated lines in a log file. You can then parse these log files to data mine for information about the network traffic on the network you are monitoring. These logs include not only a comprehensive record of every connection seen on the wire, but also application-layer transcripts such as all HTTP sessions with their requested URIs, key headers, MIME types, server responses, DNS requests with replies, SSL certificates, key content of SMTP sessions, and much more. For more information about Bro itself, read their extensive documentation.

Bro is installed at `/nsm/bro`. Its main configuration file is at `/nsm/bro/etc/node.cfg`, which is where Bro is told which network interface to monitor, i.e. `br0`. BroControl is used to start Bro and it is triggered on systemn startup because of the additional command found in `/etc/rc.local`. There is also a cron job configured to do standard bro maintenance once a day, which can be found by running `crontab -e`. Bro listens on the `br0` bonded interface and writes events in `/nsm/bro/logs/current/`. To start and stop bro manually, you use the `broctl` command such as `sudo /nsm/bro/bin/broctl stop`. More information about BroControl can be found here.

Logstash

source: How to Set Up the ELK Stack- Elasticsearch, Logstash and Kibana
source: Integrate Bro IDS with ELK Stack

Logstash is the component that parses the bro logs and pushes the data into Elasticsearch.

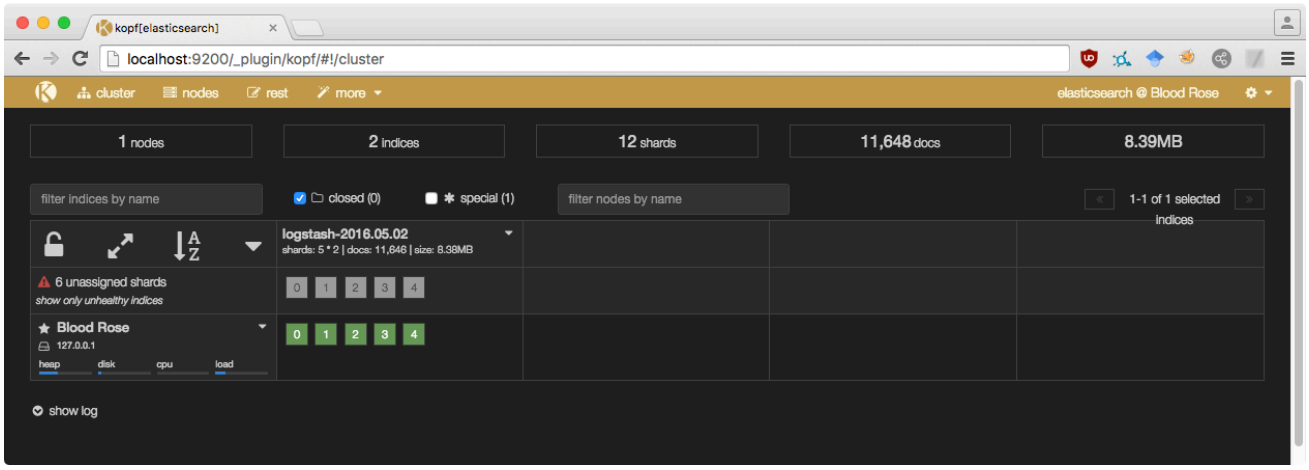
Logstash will load all the config files it finds in `/etc/logstash/conf.d` at start up. The files are available on github at https://github.com/timmolter/logstash-dfir/tree/master/conf_files/bro. Logstash is configured to start automatically at start up and can be controlled manually with: `sudo systemctl restart|start|stop logstash`.

Elasticsearch

source: How to Set Up the ELK Stack- Elasticsearch, Logstash and Kibana

Elasticsearch stores all the events generated by Bro and pushed to Elasticsearch via Logstash. Elasticsearch is also configured to start up automatically and can be controlled manually with: `sudo systemctl restart|start|stop elasticsearch`.

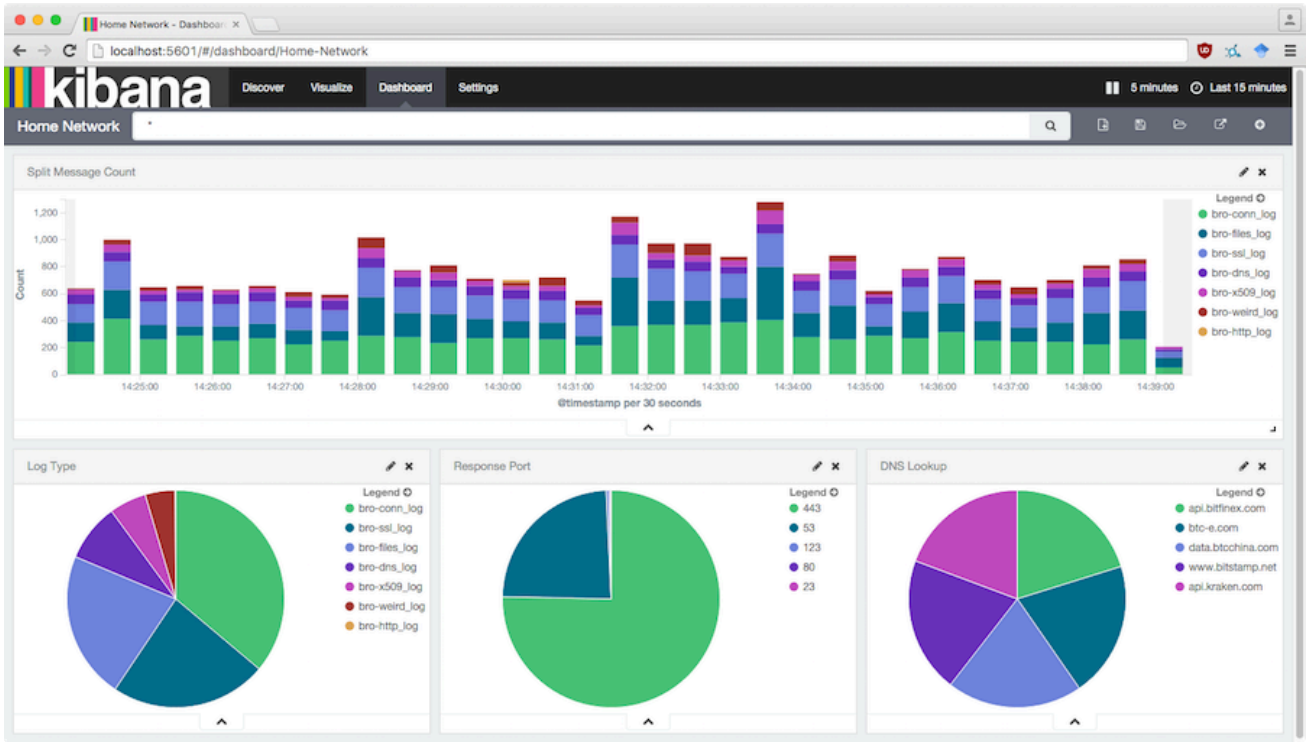
At tool called `kopf` with a web interface is also installed. It can be used to view and perform maintenance and cleanup on the data store. It can be accessed in the browser via: http://localhost:9200/_plugin/kopf/#!/cluster.



Kibana

source: How to Set Up the ELK Stack- Elasticsearch, Logstash and Kibana

Kibana is the web interface which provides a wonderful visual data analysis and discovery interface. Kibana is also configured to start up automatically and can be controlled manually with: `sudo systemctl restart|start|stop kibana`. Creating visualizations and dashboards is beyond the scope of this document, but there are extensive resources on the web to be found. The **Plug and Play Network Monitor** does however come preconfigured with a simple dashboard with a handful of visualizations and can be accessed in the browser via: http://localhost:5601/#/dashboard/Sample-Dashboard.



The sample Sample-Dashboard dashboard JSON file can be found on the Desktop called `Sample-Dashboard.json`. You can import this or any other pre-configured dashboard in Kibana under Settings ==> Objects.

General Performance Tuning

For a low-volume ELK+Bro installation, it's probably not necessary to tune the system for performance, it should run smoothly out of the box. However, as the amount of data scales up all the components are going to become constrained and will compete for hardware resources. The **Plug and Play Network Monitor** ships with a few general performance tweaks, which are outlined below for reference. Depending on your specific network characteristics and on how you are using the system, you may need to investigate addition tweaks to optimize your particular installation.

elasticsearch.yml

```
sudo nano /etc/elasticsearch/elasticsearch.yml
```

Add:

```
indices.fielddata.cache.size: .75%
indices.breaker.fielddata.limit: .85%
```

and uncomment:

```
bootstrap.mlockall: true
```

Important note: do NOT leave any spaces before the config lines in `elasticsearch.yml`!

elasticsearch

> Lucene's performance relies on this interaction with the OS. But if you give all available memory to Elasticsearch's heap, there won't be any left over for Lucene. This can seriously impact the performance of full-text search. The standard recommendation is to give 50% of the available memory to Elasticsearch heap, while leaving the other 50% free. It won't go unused; Lucene will happily gobble up whatever is left over. – http://www.elastic.co/guide/en/elasticsearch/guide/master/heap-sizing.html

Check your system's memory

```
free -m
```

Edit `elasticsearch`

```
sudo nano /etc/default/elasticsearch
```

Uncomment as shown:

```
# Heap Size (defaults to 256m min, 1g max)
ES_HEAP_SIZE=4g

# Maximum number of open files, defaults to 65535.
MAX_OPEN_FILES=65535

# Maximum locked memory size. Set to "unlimited" if you use the
# bootstrap.mlockall option in elasticsearch.yml. You must also set
# ES_HEAP_SIZE.
MAX_LOCKED_MEMORY=unlimited
```

limits.conf

```
sudo nano /etc/security/limits.conf
```

Add the two lines as shown:

```
#<domain>      <type>  <item>         <value>
#
```

```
#*               soft    core            0
#root            hard    core            100000
#*               hard    rss             10000
#@student        hard    nproc           20
#@faculty        soft    nproc           20
#@faculty        hard    nproc           50
#ftp             hard    nproc           0
#ftp             -       chroot          /ftp
#@student        -       maxlogins       4

elasticsearch - nofile 65535
elasticsearch - memlock unlimited

# End of file
```

Delete Old Index Data with Curator

```
sudo apt-get -y install python-pip
sudo pip install elasticsearch-curator
which curator
sudo crontab -e
```

Add the following:

```
30 0 * * * /usr/local/bin/curator --host 127.0.0.1 delete indices --older-than 7 --timestring \%Y.\%m.\%d --tim
```

Happy Monitoring!

This description can be found pre-installed on the desktop and can be downloaded in pdf form at:
http://knowm.org/downloads/Plug_and_Play_Network_Monitor.pdf

If you have any questions, please don't hesitate to contact us via email at `contact@knowm.org` !

Knowm Inc. | Santa Fe, New Mexico, USA | Founded in 2015 | www.knowm.org | contact@knowm.org