NETCAP

Please follow the instructions at https://docs.netcap.io/v/v0.5/ for general setup and operation. The latest version is on the software-audit-records branch and is not yet merged into master, and adds new custom audit records and advanced data correlation features to the engine core.

We supply a linux binary for testing without needing to compile: **netcap_linux_amd64_libc/net**

You can find example usage videos for using the machine learning capabilities with tensorflow and keras here:

- https://youtu.be/6cIUXHXEh-4
- https://youtu.be/kvjfytVFSnU

The source code for the Tensorflow Deep Neural Network is here: https://github.com/dreadl0ck/netcap-tf-dnn

> Caution: Syntax has changed slightly for v0.5!

e.g:

```
net.capture -r file.pcap
```

is now

```
net capture -read file.pcap
```

Operation of ELK stack

Configuration files:

- kibana.yml
- elasticsearch.yml

Configure indices:

```
export ELASTIC_PASS="password here"
/root/net capture -elastic-user elastic -elastic-pass "\$ELASTIC_PASS" -kibana "https://yourkibana.net:5443" -gen-elastic-indices
```

Manually increase limit for selected audit record field count:

```
PUT netcap-v2-http/_settings
{
  "index.mapping.total_fields.limit": 10000000
}
```

> HTTP audit records usually have a high number of fields because parameter and header names are stored as a unique fields.

Ingest data from PCAP directory:

```
screen -L time ./analyze.sh FIRST-2015_Hands-on_Network_Forensics_PCAP
```

analyze.sh:

```
#!/bin/bash

for f in $1/*/*.pcap
do
    filename=$(basename -- "$f")
    file=${filename%.pcap}
    /root/net capture -read "$f" \
        -out "${file}.net" \
        -opts datagrams \
        -local-dns \
```

```
        -geoDB \
        -elastic \
        -fileStorage files \
        -elastic-user elastic \
        -elastic-pass "$ELASTIC_PASS" \
        -kibana "https://dreadl0ck.net:5443"
done
```

> Important: this invocation uses the resolvers databases, which you will need to download and add from here: https://docs.netcap.io/v/v0.5/resolvers

> The documentation for the generation and indexing vulnerability databases is not finished yet.

Download dataset files: https://download.netresec.com/pcap/FIRST-2015/FIRST-2015_Hands-on_Network_Forensics_PCAP.zip

Add this script to the downloaded folder and execute to add the pcap file extension to all files:

```
#!/bin/bash

# tcpdump and also netcap require the pcap files to have the .pcap file extension
# this script makes sure all files in the current dir have the .pcap extension

for f in */*; do
    echo "$f"

    filename=$(basename -- "$f")
    if [[ "$filename" == "rename_pcaps.sh" ]]; then
        continue
    fi

    extension="${filename##*.}"
    filename="${filename%.*}"

# echo $extension

    if [[ "$extension" != "pcap" ]]; then
        echo "mv $f $f.pcap"
        mv "$f" "${f}.pcap"
    fi
done
```

Setup: Elastic and Kibana

Filebeat installation: https://www.elastic.co/downloads/beats/filebeat

Elastic errors:

- https://stackoverflow.com/questions/50609417/elasticsearch-error-cluster-block-exception-forbidden-12-index-read-only-all
- https://kb.objectrocket.com/elasticsearch/how-to-fix-the-forbidden-12-read-only-api-error-in-elasticsearch-282

  PUT _cluster/settings { "transient": { "cluster.routing.allocation.disk.watermark.low": "10gb", "cluster.routing.allocation.disk.watermark.high": "5gb", "cluster.routing.allocation.disk.watermark.flood_stage": "2gb", "cluster.info.update.interval": "1m" } }

via curl:

```
curl --header 'Content-Type: application/json' -XPUT http://localhost:9200/_cluster/settings -d '{
 "transient": {
   "cluster.routing.allocation.disk.watermark.low": "10gb",
   "cluster.routing.allocation.disk.watermark.high": "5gb",
   "cluster.routing.allocation.disk.watermark.flood_stage": "2gb",
   "cluster.info.update.interval": "1m"
 }
}'
```

Elastic Utils

Delete index:

```
curl -XDELETE localhost:9200/indexName
```

Configure mapping:

```
DELETE netcap-audit-records
PUT netcap-audit-records
PUT /netcap-audit-records/_mapping
{
    "properties": {
    "Timestamp": {
        "type": "date"
    },
    "Version": {
        "type": "text"
    },
    "ID": {
        "type": "text"
    },
    "Protocol": {
        "type": "text"
    }
}
}
```

Create index pattern:

```
    curl -X POST "http://localhost:5601/api/saved_objects/index-pattern" -H 'kbn-xsrf: true' -H 'Content-Type:
application/json' -d'
    {
    "attributes": {
    "title": "netcap-ethernet\*",
    "timeFieldName": "time"
    }
    }'
```

```
curl -X GET "\${KIBANA_ENDPOINT}/api/saved_objects/\_find?type=index-pattern&search_fields=title&search=netcap\*" -H
'kbn-xsrf: true' --user "elastic:\$ELASTIC_PASS"

curl -X DELETE "${KIBANA_ENDPOINT}/api/saved_objects/index_pattern/f245ba40-e1a9-11ea-a16f-af07127330c7" -H 'kbn-xsrf:
true' --user "elastic:$ELASTIC_PASS"
```

Increase field limit for selected audit records:

```
PUT netcap-http/\_settings
{
"index.mapping.total_fields.limit": 10000
}
```

TCP anomalies query:

```
{
  "description": "TCP Anomalies",
  "source": {
    "index": "netcap-tcp*"
  },
  "dest": {
    "index": "netcap-outliers-tcp"
  },
  "analyzed_fields": {
    "includes": [
                "SrcPort",
                "DstPort",
                "SYN",
                "ACK",
                "RST",
                "FIN",
                "PayloadSize"]
  },
  "analysis": {
    "outlier_detection": {}
  },
  "model_memory_limit": "5000mb"
```

```
}
```

TCP Flag anomalies:

```
{
    "description": "TCP Anomalies",
    "source": {
        "index": "netcap-tcp\*"
    },
    "dest": {
        "index": "netcap-outliers-tcp"
    },
    "analyzed_fields": {
        "includes": [
        "SYN",
        "ACK",
        "RST",
        "FIN"]
    },
    "analysis": {
        "outlier_detection": {}
    },
    "model_memory_limit": "5000mb"
}
```