Configuración Cowrie

2018-10-29

Requisitos

Tenemos que tener instalado docker

Instalación de la imagen y creación del contenedor

Para instalar la imagen y crear el contenedor usaremos:

```
docker pull ouspg/cowrie
docker run -dit -p 2222:2222 -p 2223:2223 -v $(pwd)/dl:/home/cowrie/cowrie/dl -v $(pwd)/log:/home/cowrie/cowr
```

Para obtener una consola del contenedor en ejecución usaremos:

```
docker exec -it [CONTAINER_ID] sh
```

vim no esta instalado, solo esta vi, todos los ficheros están en el directorio local

Si tenemos parada la maquina y queremos volver a ejecutarla

```
docker start [CONTAINER_ID]
```

Configuración de log en la base de datos sqlite

Para guardar los log en una base de datos sqlite necesitamos tenemos que conectarnos al contenedor y descomentar las lineas 348 y 348 aproximadamente del fichero *cowrie.cfg*. Hay que acceder el contendedor para hacer este paso

```
[output_sqlite]
db_file = log/cowrie.db # Cuidado, por defecto lo guarda en el directorio local no el log
```

Una vez hecho eso, se te creara la base de datos vacía en el directorio que le hayas indicado y tendrás que cargar las tablas que necesitaremos, esto lo haremos desde nuestra maquina física ya que tenemos linkeado el directorio con el volumen del contenedor, ejecutando el siguiente comando:

```
sqlite3 cowrie.db < sqlite3.sql
```

El fichero sqlite3.sql será:

```
CREATE TABLE IF NOT EXISTS `auth` (
  `id` INTEGER PRIMARY KEY,
  `session` char(32) NOT NULL,
  `success` tinyint(1) NOT NULL,
  `username` varchar(100) NOT NULL,
  `password` varchar(100) NOT NULL,
  `timestamp` datetime NOT NULL
) ;

CREATE TABLE IF NOT EXISTS `clients` (
  `id` INTEGER PRIMARY KEY,
  `version` varchar(50) NOT NULL
) ;
```

```
CREATE TABLE IF NOT EXISTS `input` (
  `id` INTEGER PRIMARY KEY,
  `session` char(32) NOT NULL,
  `timestamp` datetime NOT NULL,
  `realm` varchar(50) default NULL,
  `success` tinyint(1) default NULL,
  `input` text NOT NULL
) ;
CREATE INDEX input_index ON input(session, timestamp, realm);

CREATE TABLE IF NOT EXISTS `sensors` (
  `id` INTEGER PRIMARY KEY,
  `ip` varchar(15) NOT NULL
) ;

CREATE TABLE IF NOT EXISTS `sessions` (
  `id` char(32) NOT NULL PRIMARY KEY,
  `starttime` datetime NOT NULL,
  `endtime` datetime default NULL,
  `sensor` int(4) NOT NULL,
  `ip` varchar(15) NOT NULL default '',
  `termsize` varchar(7) default NULL,
  `client` int(4) default NULL
) ;
CREATE INDEX sessions_index ON sessions(starttime, sensor);

CREATE TABLE IF NOT EXISTS `ttylog` (
  `id` INTEGER PRIMARY KEY,
  `session` char(32) NOT NULL,
  `ttylog` varchar(100) NOT NULL,
  `size` int(11) NOT NULL
) ;

CREATE TABLE IF NOT EXISTS `downloads` (
  `id` INTEGER PRIMARY KEY,
  `session` CHAR( 32 ) NOT NULL,
  `timestamp` datetime NOT NULL,
  `url` text NOT NULL,
  `outfile` text NOT NULL,
  `shasum` varchar(64) default NULL
) ;
CREATE INDEX downloads_index ON downloads(session, timestamp);

CREATE TABLE IF NOT EXISTS `keyfingerprints` (
  `id` INTEGER PRIMARY KEY,
  `session` CHAR( 32 ) NOT NULL,
  `username` varchar(100) NOT NULL,
  `fingerprint` varchar(100) NOT NULL
) ;
```