OSINT – Open Source Intelligence

The power of the Search Engine Attacks

El presente manual muestra desde un punto de vista práctico cómo emplear los diferentes buscadores de contenidos disponibles en internet para realizar labores de inteligencia de fuentes abiertas (OSINT) mediante taxonomías y ontologías.

La Inteligencia de Fuentes Abiertas (OSINT) es aquella recopilada a partir de fuentes disponibles al público. En la comunidad de inteligencia, el término "abierto" se refiere a las fuentes abiertas, disponibles al público (a diferencia de las fuentes privadas o semiprivadas). De esta forma, se podría definir OSINT como una forma de recolectar información y datos a los que aplicarle una cierta inteligencia lo cual implica encontrar, seleccionar y adquirir información desde fuentes públicas, como buscadores, para poder analizarla y obtener gran cantidad de información acerca de una determinada temática.

El alcance del manual se circunscribe a Google, Bing, Yahoo y, por supuesto, Shodan, centrándose principalmente en este último y Google

Para poder realizar búsquedas avanzadas que permitan obtener ciertos datos de valor e necesario definer primero los operadores/comandos de los diferentes buscadores.

Google

En el caso de Google los principales son los siguientes:

- **site:**
  - Permite definir una búsqueda que muestre los resultados de un sitio en concreto. Por ejemplo:

    site: highsec.es

    Mostrará resultados dentro del dominio de highsec.es
- **intitle:**
  - Buscará la cadena que le sea pasada como parámetro, pudiendo estar sólo parte de dicha cadena en el título. Por ejemplo.

    intitle:"index of /"

    Esto permitirá buscar servidores que tengan un listado de archivos mediante un index of.

- **allintitle:**
  - Similar al anterior pero indicando de forma expresa y obligatoria que toda la cadena que sea definida tiene que aparecer en el título. Por ejemplo:

    allintitle:"index of /admin/"

    Buscará todos los resultados que coincidan exactamente

- **inurl:**
    - Permitirá buscar una determinada cadena dentro de la URL. Por ejemplo:

      inurl:"c99.php"

      Lo cual nos mostrara los servidores que tengan instalada una shell c99 en php.
- **allinurl:**
    - Similar al anterior "all", pues la diferencia con inurl es que en este comando todo debe aparecer obligatoriamente en la URL. Por ejemplo:

      allinurl:"and+1=0+union+select+1"

      Esta query devolverá páginas que muy posiblemente sean vulnerables a un SQLi.

- **intext:**
    - Permite hacer una búsqueda de la cadena en el texto de los resultados, ni en el título, ni en URL, etc... Por ejemplo:

      intext:"@gmail.com password="

      Este resultado nos podría mostrar algunas cuentas de correos de Google con su usuario/email y contraseña.

- **allintext:**
    - Permitirá buscar los resultados que solo contengan de forma exacta lo que hemos buscado. Por ejemplo:

      allintext:"@yahoo.es:pass"

      Este comando nos podría permitir buscar algunas cuentas de yahoo.es.
- **cache:**
    - Permitirá ver una página en la cache de Google, por lo que no será necesario conectarse a la página real. Esto es muy práctico en la fase de fingerprinting en un test de intrusión. Por ejemplo:

      cache:highsec.es

      Esto mostraría la página de highsec.es que tiene Google en cache.
- **info:**
    - Proporcionará cierta información acerca de un dominio. Por ejemplo:

      info:highsec.es

      Donde solicitará si queremos mostrar la página en cache, si queremos ver webs similares, si queremos ver aquellas páginas que tengan un enlace a highsec.es, si queremos buscar las diferentes páginas de HighSec o si queremos buscar las web donde aparezca la frase highsec.es.

- **link:**
  - Permite buscar las páginas que tienen un determinado link. Por ejemplo:

    link: www.highsec.es

    Es importante recalcar que buscara SOLO ese link, que sería distinto de highsec.es

- **ext:**
  - Buscará la cadena introducida como la extensión del archivo. Por ejemplo:

    ext:php3

    Esto nos permitirá hacer búsquedas donde los resultados sean archivos .php
- **filetype:**
  - Similar al anterior, pero Google encontrará unos determinados tipos bien conocidos como: pdf, doc, docx, xml, txt...

    filetype:txt inurl:robots

    Esta búsqueda sacará los archivos robots.txt
- **'-' (El menos):**
  - Permitirá negar un determinado operando. Por ejemplo:

    -filetype:pdf

    Buscará cualquier resultado menos archivos pdf.
- **'||':**
  - Permitirá añadir varias condiciones (Es un OR). Por ejemplo:

    ext:txt || ext:sql intext:password intext:username

    Esto sería un comando que permitiría obtener algunos archivos con usuarios y contraseñas que tengan extensión txt o sql.

- **'&&':**
  - Permite hacer un AND de dos condiciones, es decir que tiene que cumplirse las dos en los resultados a buscar. Por ejemplo:

    ext:txt || ext:sql intext:password && intext:username

    o

    (ext:txt || ext:sql) (intext:password && intext:username)

    Esta búsqueda sería similar a la anterior pero se impone como condición que tiene que aparecer tanto password como username en el texto.

La Google Hacking DataBase (GHDB) es un proyecto que surgió hace algunos años donde la gente comparte sus *dorks* personalizados, nombre que se le da a las búsquedas avanzas que hacen uso de operadores para conseguir una cierta información.

http://www.exploit-db.com/google-dorks/

http://www.hackersforcharity.org/ghdb/

Existen gran variedad de *dorks*, pero suelen ser *deprecated* porque, aunque sin duda supone un buen punto de partida para obtener una visión general de cómo se podría plasmar en un comando de Google una búsqueda completa en internet de una forma precisa, aunque como se

verá más adelante, es cuestión de definir bien los criterios y parámetros de búsqueda.

Antes de profundizar en el uso de los *dorks* y mostrar el proceso de definición que un analista debería seguir, se definirá el término "Google Hacking", siendo uno de los métodos principales que utilizan/utilizaban algunos grupos como Lulz o Anonymous para llevar a cabo sus ataques. Esto se debe a que pueden llegar recopilar gran cantidad de información de la víctima como pueden ser cuentas de redes sociales, información confidencial y un sin fin de datos de utilidad.

En estas circunstancias, es realmente práctico usar la caché de Google para visitar una página, ya que de esta forma no se realizará una conexión directa con el objetivo y por lo tanto no se dejará ningún rastro.

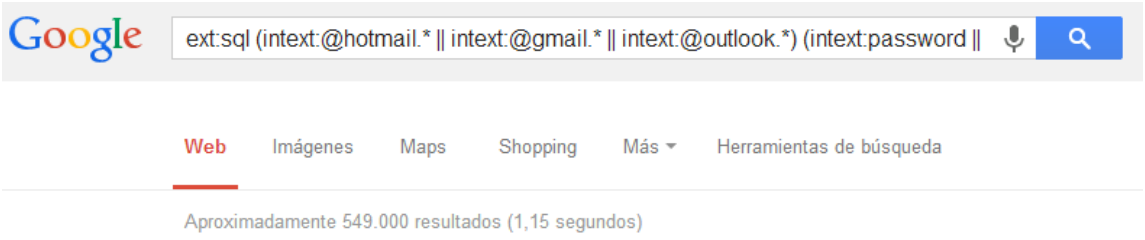
En este punto se muestra los distintos vectores de información accesible a un potencial atacante como cuentas de correo, servidores mal configurados, documentos confidenciales, páginas que han sido *hackeadas*, localizar vulnerabilidades web, cámaras IP, etc.

En el caso de las cuenta de correos, cabría preguntarse qué tipo de archivos podrían estar almacenados en unas cuentas en un servidor de correo, como podrían ser archivos sql, log y xls. Tras este punto, se debería pensar cómo buscar un correo y después cómo localiza una contraseña dentro de un fichero. Esta podría ser una de las cadenas de búsqueda:

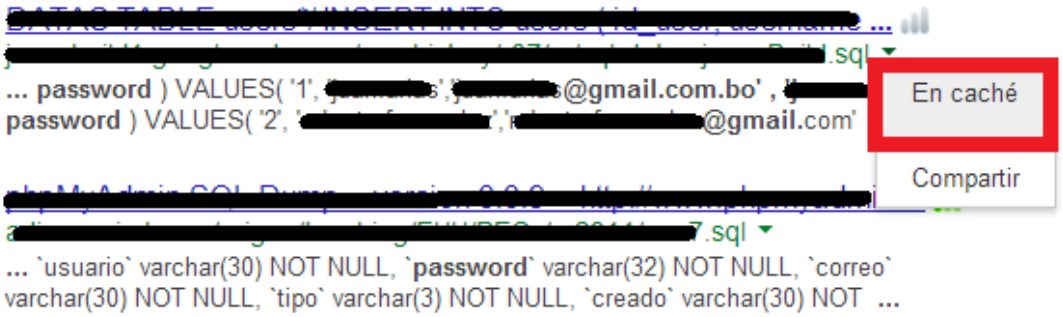
ext:sql (intext:@hotmail.* || intext:@gmail.* || intext:@outlook.*) (intext:password || pass)

Donde se estaría buscando en un archivo sql las cadenas “@hotmail.”, “@gmail.” y “@outlook.”, lo cual permitirá encontrar cuentas de cualquier país ya sean @gmail.es o @gmail.com por ejemplo. Y por último se procede a realizar la búsqueda en el texto que aparezca ‘password’ o ‘pass’.

Como se puede comprobar, aparecen algunos resultados:



Ahora se selecciona el enlace deseado y se visualiza a través de la caché del propio navegador, de tal forma que únicamente se quedara registrado en Google:



Este extremo podría seguir desarrollándose para dar lugar a otros *dorks* como por ejemplo:

Cuentas de correo:
ext:sql (intext:@hotmail.* || intext:@gmail.* || intext:@outlook.*) (intext:password || pass)
ext:xls (intext:@hotmail.* || intext:@gmail.* || intext:@outlook.*) (intext:password || pass)

Cuentas de correo de gobiernos:
ext:sql (intext:@gov.* || intext:@gob.*) (intext:password || pass)
ext:xls (intext:@gov.* || intext:@gob.*) (intext:password || pass)
(ext:sql ||ext:xls) (intext:@gov.* || intext:@gob.*) (intext:password || pass)

Cuentas de correo generales:
(ext:sql ||ext:xls) (intext:@hotmail.* || intext:@gmail.* || intext:@outlook.* || intext:@gov.* || intext:@gob.*) intext:password

ext:log (intext:@hotmail.* || intext:@gmail.* || intext:@outlook.* || intext:@gov.* || intext:@gob.*) intext:password

Cuentas de correo en los logs:
ext:log (intext:@hotmail.* || intext:@gmail.* || intext:@outlook.* || intext:@gov.* || intext:@gob.*) intext:password

A continuación se muestran algunos ejemplos de cómo se podría encontrar servidores vulnerables. Es importante especificar que el resultado depende de lo que se desee buscar, ya sea una vulnerabilidad, configuraciones por defecto u otro valor. En este caso se muestra cómo sería posible obtener las configuraciones de un servidor para capturar las credenciales de acceso vía web y ftp.

filetype:config inurl:web.config inurl:ftp

Otros posibles ejemplos podrían ser los siguientes:
Index of de sitios gubernamentales:

(site:gob.* || site:gov.*) intitle:"index of /" intext:"Last modified Size Description"

Servidores ftp gubernamentales:

(site:gob.* || site:gov.*) inurl:"ftp://ftp" site:ftp.*
intext:warning intext:/var/www/ ext:php (site:gob.* || site:gov.*)

Para localizar documentos de carácter confidencial, lo primero que se podría hacer es filtrar la búsqueda únicamente a dominio de cualquier gobierno, después buscar de diversas formas la palabra 'confidencial' o 'confidential'. Algunos ejemplos podrían ser los siguientes en donde se buscan documentos que no sean públicos, documentos que sean algún tipo de reporte y datos confidenciales referentes a contabilidad y datos bancarios.

Archivos confidenciales de sitios gubernamentales:

(site:gob.* || site:gov.*) && (intitle:confidential || intitle:confidencial) ext:pdf future
(site:gob.* || site:gov.*) intext:"confidential reports" ext:pdf future

Que no sean públicos:

(site:gob.* || site:gov.*) intext:"confidential reports" ext:pdf -"public"

Contabilidad y bancarios:

(site:gob.* || site:gov.*) intext:"confidential" ext:xls -"public" (site:gob.* || site:gov.*) intext:"confidential" ext:xls -"public" "financial" intext:"confidential" ext:xls -"public" "financial" "bank" "account"

Ahora que ya se vislumbra la potencia que realmente tienen los operadores avanzados de Google para obtener información, se procede a analizar cómo se podría localizar una página gubernamental que haya sido hackeadas. Para ello un ejemplo podría ser el siguiente:

(site:gob.* || site:gov.*) intitle:"hacked by"

O buscando por ejemplo archivos /etc/passwd públicos en sitios gubernamentales que potencialmente podrían permitir acceso al servidor. Un par de ejemplos serian:

inurl:"/etc/passwd" ext:php (site:gob.* || site:gov.*)
intext:root:x:0:0:root:/root:/bin/bash ext:php (site:gob.* || site:gov.*)

A fin de localizar vulnerabilidades web, se podría por ejemplo buscar por un error muy típico de MySQL que es "Warning: mysql_fetch_array", indicando que hay un error en la consulta que está realizando la aplicación web a la base de datos y que muy posiblemente sea susceptible a un SQL Injection. Para buscar esto podríamos hacer una consulta como:

intext:"warning mysql_fetch_array" inurl:.php?*= ext:php

También se podría buscar páginas susceptibles a SQL Injection donde algún usuario ya ha intentado realizar el ataque lo cual podría significar que son vulnerables como por ejemplo:

(site:gob.* || site:gov.*) inurl:"union+select+1" filetyp:php

Para terminar, se analiza cómo se podría localizar algunas cámaras IP, aunque para esto se verá más adelante que el buscador Shodan es más eficaz. En este caso el dork empleado es eficaz en tanto en cuanto se ha identificado que sélo ese tipo de modelo de cámaras IP tiene esa URL, por lo tanto es algo identificativo que al buscar en Google permite encontrar dichas cámaras. El ejemplo sería el siguiente:
inurl:/control/userimage.html

El ejemplo visto es un caso puntual, pero si se desease buscar otro tipo sólo sería necesario investigar algún patrón que sea identificativo, ya sea que siempre se pone el mismo título, que en la web siempre aparece alguna cadena concreta, etc.

Bing

Bing, buscador de Microsoft, comparte operadores con Google como pueden ser: ext, filetype, intitle, site, los operadores || y &&.

Por otro lado se introducen algunos distintos, como:

- **ip:**
  - Permitirá buscar por una IP concreta.
- **url**:
  - Sustituye al inur

Aunque Bing es otro buscador ampliamente usado al realizar labores de OSINT o para identificar a un objetivo en un test de intrusión hay que decir que se obtienen menos resultados que en Google, aunque a veces pueden ser distintos, de ahí la necesidad de ejecutar búsquedas con ambos.

Yahoo

Comparte con Google los siguientes: site, link, inurl, intitle, y además tiene:

- **hostname:**
    - Que permite buscar por un dominio concreto

Shodan

¿Qué es Shodan?

Se podría decir que es un buscador como Google, pero tiene una muy importante diferencia y es que Google va recorriendo Internet y cacheando o almacenando la información web, es decir la que se puede visualizar a través de un navegador, ya sea http, https, http-alternativo, ftp, etc.

Shodan, sin embargo, recorre internet escaneado cada IP, sacando los servicios que tiene abiertos, capturando y cacheando el banner que le devuelve el servicio. Esto nos permite hacer búsquedas en la información retornadas por los servicios, de tal forma que permite encontrar, por ejemplo, todos los servidores apache de la versión 2.20, todas las IPs de Madrid que tengan telnet abierto, etc.

A continuación se profundiza en qué es, qué operadores implementa tanto para un profesional que ejecuta un test de intrusión o realiza labores de OSINT, así como la visión que podría tener un atacante.

Los operadores o comandos en Shodan para buscar son los siguientes:

- **city:**
    - búsqueda dirigida en una ciudad. Por ejemplo:

      apache city:"Madrid"

- **country**:
  - Búsquedas por país. Por ejemplo:

    ssh country:es

    Esto mostraría todo lo que encuentre de SSH en España

- **geo**:
  - Especificando una posición mediante latitud y longitud. Por ejemplo:

    scada geo:40.02,4.03

    Permitirá buscar coincidencias de SCADA en Madrid.
- **hostname:**
  - Búsqueda de un dominio concreto. Por ejemplo:
    hostname:uam.es
    o
    hostname:.es
- **net**:
  - Búsqueda en un rango de red. Por ejemplo:
    net:8.8.8.8/32
    o
    net:8.8.8.8

- **os:**
  - Búsqueda por sistema operativo. Por ejemplo.

    os:linux city:"Madrid".

- **port**:
  - Este comando es bastante importante ya que permite filtrar los resultados por puerto. Por ejemplo:
    port:23 city:"Madrid"

- **before/after:**
    - Sirve para filtrar los resultados según cuando los haya escaneado Shodan por última vez.

- **'-':**
    - Para definir que el parámetro que le sigue tiene que excluirlo de los resultados. Por ejemplo.

        port:23 -city:"Madrid"

        Esto permitiría buscar todas las IP que tengan el puerto 23 abierto que NO sean de Madrid

- '|':
    - Para realizar el OR de Google. Por ejemplo:

      port:21 | port:22

Una vez analizados todos los comandos, se procede a realizar combinaciones de los mismos. A continuación se muestra una gran lista de *dorks* para Shodan que se han realizado a modo de ejemplo para que se pueda apreciar ver la cantidad de elementos conectados a internet que es posible encontrar gracias a este buscador.

Para conseguir estos dorks es conveniente seguir una metodología de trabajo específica, y lo primero es tener claro exactamente qué se desea encont*rar* para, una vez definido, visitar las webs de los fabricantes, localizando el producto y localizando especificaciones, fotos que puede tener la web, qué banner tiene, qué puertos le identifican, si es necesario o no autenticación en la página principal, etc Con todos estosdatos es relativamente sencillo ir construyendo dorks que cada delimiten más el objetivo.

En caso de que el dispositivo a buscar tenga contraseña, muchas veces su valor es el que tiene por defecto de fábrica, por lo que únicamente tendremos que localizar el manual del fabricante y comprobar cuál es, de esta forma cualquier atacante podría entrar en un sistema mal configurado de una forma trivial.

A continuación se muestran algunos ejemplos:

<u>Routers por defecto</u>:

Routers de ONO [Password por defecto 'vacio'/admin]:
WWW-Authenticate: Basic realm="Thomson" country:ES org:"ONO"

Routers Huawei [Open]:
title:"huawei home"

Routers D-Link [Open o admin/'vacio']:
title:"home router" server:"siyou server" org:"Mega Cable, S.A. de C.V."

Routers Israel [admin/admin]
title:"residential gateway" "Content-Length: 2771"

Routers GPON [root/admin]
title:"GPON Home Gateway"

Routers y PBX Orange [admin/admin]
title:livebox

Routers Telefónica ["Sistema contra 'hackers'", por defecto 1234/1234]
title:"Home Station"

Routers Telefónica [Pass por defecto 1234/1234, telnet abierto]
title:"Web-base configurator" country:es

<u>Casas Inteligentes:</u>

Casas inteligentes [admin/admin]

title:"My Home" Content-Length: 198
title:"myhome"

Cámaras (Solo algunos ejemplos):

Marca iQeye:
title:iqeye

Android:
title:"Android Webcam Server"
Android Webcam

Referente al tema de las cámaras dentro de poco lanzare una herramienta pensada para realizar pentest y autopwn de estos dispositivos.

Sistemas de Ventilación, Alarmas, Luces, etc..:

title:"real time monitoring avtech"
Server:IQ3 Content-Length: 184 (Regular temperatura, ventilación, alarmas..)

Sistemas SCADA:

Planta petrolífera:
title:"Web Scada" o "INDIAS WEB SCADA"

UPS Liebert:
title:liebert port:80

i.Lon Enterprise energy management and streetlight management [(ilon/ilon)]:
title:"i.LON SmartServer 2.0"
Server: WindRiver-WebServer/4.4

Sistemas Satélites ComTech [comtech/comtech]:
title:comtech

Servidores de almacenamiento:
title:"synology nas"
title:"diskstation nas"

Sistemas de audio:
title:"qsa 500"
Server: eCos/1.0 200

Sistemas solares:
title:"atlas solar"
title:"control solar"

Termostatos / aire:
title:"Thermostat control"
title:"Control | Air Post"

Sistemas Catherpila:
title:"Communicator" Server: Z-World Rabbit

Controlador Switch:
title:"Calypso"

Cajas fuertes:
title:"CacheTalk III"

Controladores SCADA Schneider y otros:
title:" PowerLogic" ('vacio'/admin)
title:"FactoryCast" (USER/USER y ftp:ntpupdate/ntpupdate)

Control de los pacientes en hospitales:
title:"tracevue"

Semáforos:
executive 3.3
title:"Pips Technology P357 Error"
ruggedcom port:23 (admin/admin o guest/guest !!!!)
ruggedcom port:80 (admin/admin o guest/guest !!!!)

Paneles de carreteras:
Daktronics

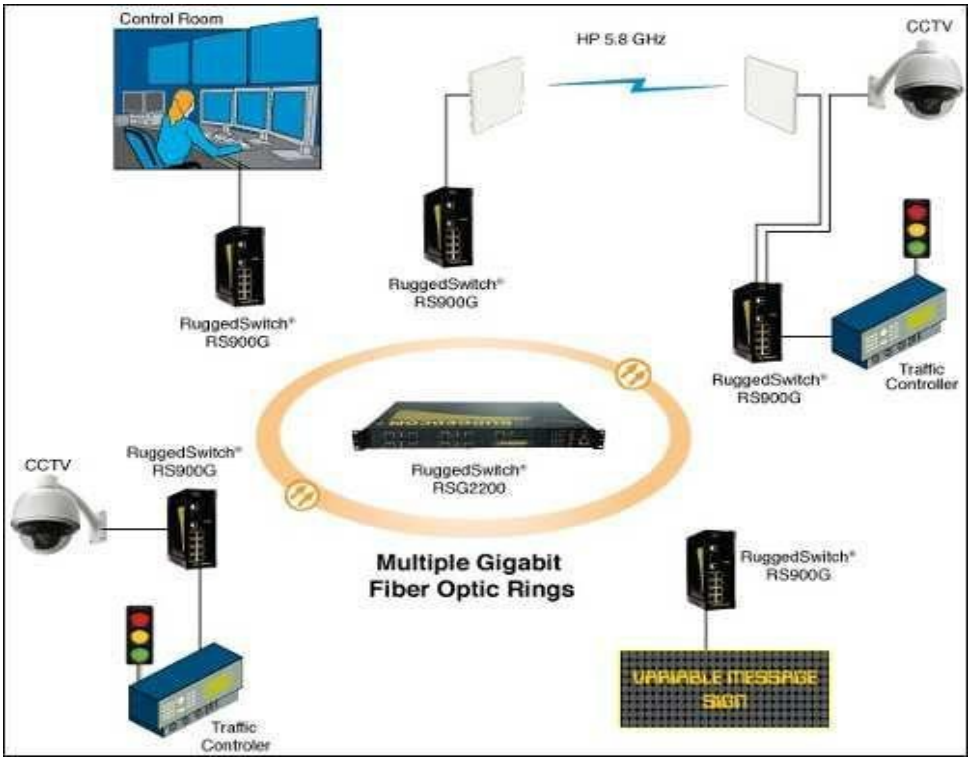
Sensores:
akcp

Varios:
title:"smartcontrol" (Backdoor superman/superman)
title:Bacnet

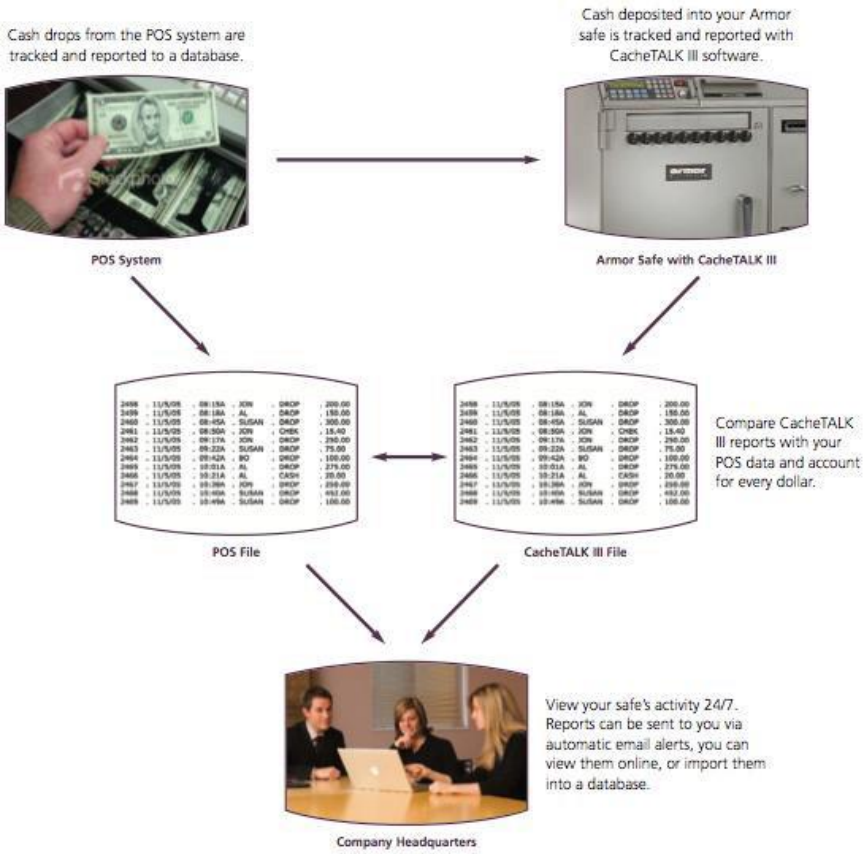
A continuación se muestran algunas imágenes de utilidad cuando se está buscando ciertos dispositivos como pueden ser cajas fuertes conectadas a internet o los controladores de semáforos.

Imágenes de algunos ejemplos:

Control de sistemas de tráfico:



Control de cajas fuertes desde internet:



AUTOR: Eduardo Arriols Nuñez
Correo: Eduardo@highsec.es