Práctica #2: Análisis ético de la exposición IoT mediante Shodan

Curso: *Seguridad en Comunicaciones Inalámbricas*
Fecha de entrega:

1. ¿Qué es Shodan?

Shodan `https://www.shodan.io/` es un motor de búsqueda que permite consultar una base de datos que indexa información previamente recolectada por sus propios servidores, funcionando como un buscador de servicios expuestos en Internet. Los servicios se detectan a través del análisis de los **banners**. Los banners incluyen (en formato texto) información que permite identificar interfaces de acceso o características de los servicios habilitados. Recuerda que Shodan es una herramienta legítima y valiosa que debe utilizarse de manera ética y legal.

En algunas fuentes Shodan está catalogado como una de las herramientas más adecuadas para aprender Ciberseguridad en IoT debido a la facilidad de uso tanto de su interfaz web 1 cómo la de su API.

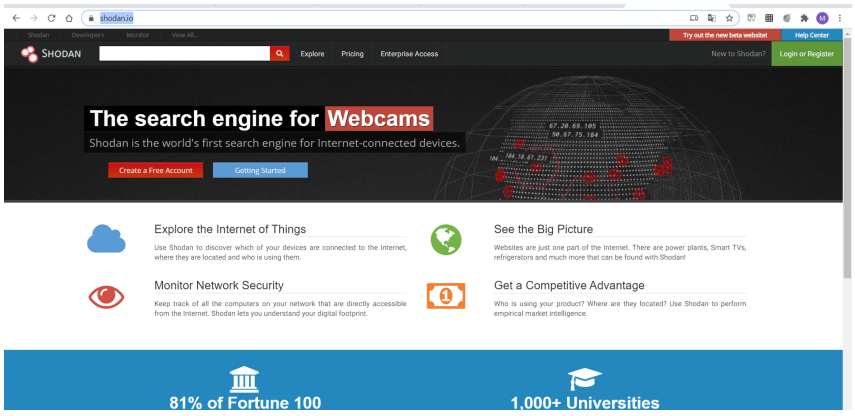

Figura 1: Interface Web de Shodan

Los resultados obtenidos en las búsquedas realizadas dependen del tipo de cuenta de que disponga el usuario. Por eso para poder realizar esta práctica se recomienda registrarse en Shodan con la cuenta de correo académica (ull.edu.es) y luego **solicitar un upgrade a cuenta académica**. Para realizar este upgrade se debe enviar un correo electrónico a la dirección `academic@shodan.io`, desde la cuenta que quieren actualizar, solicitando el upgrade[1].

Con este upgrade podrán realizar más búsquedas, usar un conjunto más amplio de filtros, uso, tener acceso a la herramienta Shodan maps (`https://maps.shodan.io/`), etc.

[1] `https://help.shodan.io/the-basics/academic-upgrade`

Shodan indexa la "superficie pública" de Internet: recoge banners de servicios (el texto que responden servidores en puertos como encabezados HTTP, banners SSH, respuestas MQTT, etc.), los puertos abiertos y los servicios asociados (p. ej. 80/443, 1883, 502, 5683, 47808), y metadatos relevantes como la fecha de la última captura, ASN/organización, geolocalización aproximada, hostnames, encabezados HTTP y certificados SSL. Además incorpora etiquetas automáticas que identifican producto y versión detectados y, cuando procede, una lista de CVE asociadas en el campo vulns; en algunos casos también indexa capturas o recursos (pantallas, endpoints, imágenes) que el dispositivo expone.

Esa información facilita la detección de vulnerabilidades porque los banners suelen revelar el nombre y la versión del software, lo que permite comparar rápidamente contra bases de datos CVE para identificar software no parcheado. Los puertos abiertos (p. ej. paneles de administración HTTP o brokers MQTT sin TLS) muestran la superficie de ataque y posibles vectores de acceso remoto; los metadatos (fecha, ASN) ayudan a priorizar activos críticos y correlacionar infraestructuras como gateways IoT. Además, dado que muchos sensores inalámbricos (BLE, Zigbee, etc.) se conectan a Internet a través de pasarelas, una pasarela expuesta en Internet puede convertirse en la vía para afectar toda la red inalámbrica interna.

2. Primeras búsquedas

Shodan ofrece un listado de resultados al ejecutar una búsqueda tal y cómo se refleja en la figura 2

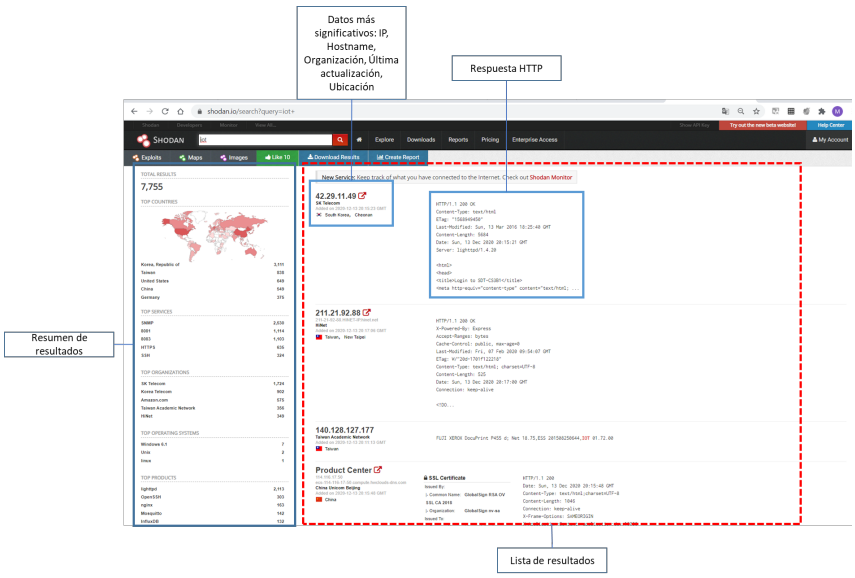

Figura 2: Listado de resultados de una búsqueda

En la figura 3 se pueden apreciar los resultados obtenidos para un dispositivo concreto. Además de los datos de identificación del dispositivo, se puede tener acceso a información sobre vulnerabilidades detectadas y tecnologías web encontradas en el dispositivo. Dichas vulnerabilidades están identificadas con la codificación CVE correspondiente.

Para realizar búsquedas más eficientes Shodan incluye **filtros**. A continuación se describen algunos de ellos.

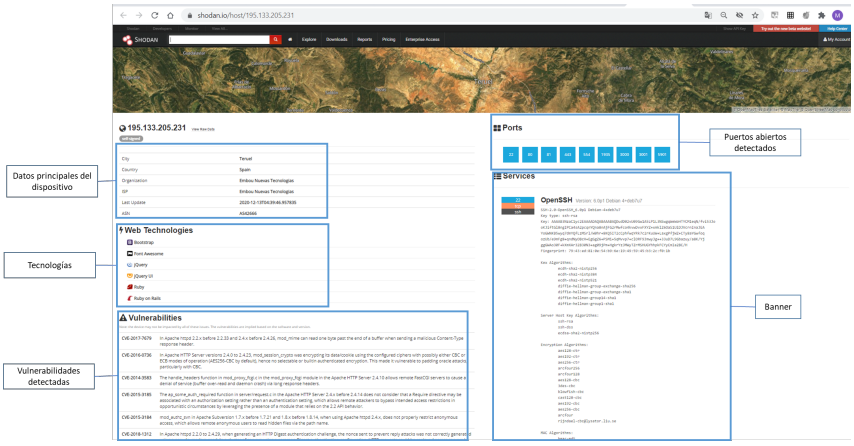

Figura 3: Resultados para un dispositivo concreto

- country: Para buscar en un país en específico. country:py (`https://en.wikipedia.org/wiki/ISO_3166-1_alpha-2`
- city: Filtro por ciudad. city:"Los Angeles"
- port: Para buscar dispositivos que tengan un puerto abierto. port:3306
- net: Búsqueda de una ip específica o rangos de ip. ip:182.93.44.0/24
- hostname: Busca el texto que le indiquemos en el nombre del host. hostname:iplocal
- geo: Buscar dispositivos mediante coordenadas. geo:32.9775,-70.1293
- os: Para listar un sistema operativo determinado. os:Linux
- after: Dispositivos agregados después de la fecha.
- before: Lo mismo, pero antes de la fecha. after/before:27/03/2015
- *has_screenshot*: Nos muestra dispositivos de los cuales hay una captura. *has_screenshot* : *true*
- vuln: busca directamente por vulnerabilidad, pero está actualmente sujeto a pago o a usuarios académicos: vuln:"CVE..."

Se pueden consultar más filtros en `https://github.com/JavierOlmedo/shodan-filters`, `https://www.shodan.io/search/filters`.

3. ¿Qué aporta Shodan?

Las diferentes vulnerabilidades de los sistemas que pueden ser explotadas a través de SHODAN tienen su origen principalmente en tres errores de configuración de seguridad:

- Cuentas de usuarios por defecto o de fácil identificación.
- Contraseñas por defecto o contraseñas débiles, que pueden ser fácilmente adivinadas o se encuentran accesibles.

- Falta de mecanismos de bloqueo de cuenta al detectar cierto número de intentos fallidos.

Por tanto, se puede introducir claves en el campo de búsqueda de acuerdo a los contenidos de los banners obtenidos. Para ello es interesante conocer los códigos de estado HTTP que nos indican los requerimientos de autenticación. De manera muy resumida, estos son los mostrados en la figura 4.

Status Code	Description
200 OK	Request succeeded
401 Unauthorized	Request requires authentication
403 Forbidden	Request is denied regardless of authentication

Figura 4: Códigos de estado de autenticación

Shodan permite encontrar fallos de seguridad provocados por configuraciones erróneas siguiendo los pasos siguientes:

1. Exploración e Identificación de equipos con vulnerabilidades de seguridad.
2. Búsqueda en la web de contraseñas débiles o por defecto vinculadas a los equipos identificados
3. Intento de acceso a sus diferentes servicios o puertas de acceso vulnerables asociados mediante logeo empleando contraseñas débiles (contraseñas por defecto) o por mecanismos de fuerza bruta.

4. Shodan en el ámbito industrial

Shodan permite evaluar la seguridad de dispositivos incluidos aquellos de entornos específicos como el caso de los Sistemas de Control Industrial (ICS). En este ámbito los protocolos usados para la comunicación suelen ser a menudo exclusivos del fabricante. Por ejemplo, en los controladores lógicos programables (PLC) uno de los más comunes es Modbus.

Cuadro 1: Protocolos industriales y SCADA más comunes

Protocolo	**Puerto**	**Descripción breve / uso principal**
Modbus/TCP	502	Protocolo maestro-esclavo para comunicación entre PLCs y dispositivos de campo. Transmite registros de sensores y comandos de control. No incluye autenticación ni cifrado.

Continúa en la siguiente página

Continuación de la tabla 1

Protocolo	Puerto	Descripción breve / uso principal
BACnet/IP	47808 (0xBAC0)	Usado en sistemas de gestión de edificios (BMS): HVAC, iluminación, ascensores. Permite comunicación entre controladores y estaciones de supervisión. Carece de autenticación robusta.
DNP3	20000	Utilizado en redes eléctricas y automatización de subestaciones. Permite comunicación entre RTUs, PLCs y centros SCADA. Las versiones antiguas no cifran los datos.
IEC 60870-5-104	2404	Estándar europeo para telecontrol en sistemas eléctricos e industriales. Similar a DNP3 pero con implementación TCP/IP. Carece de seguridad nativa.
PROFINET / S7comm	102, 34962–34964	Protocolos de Siemens y PROFIBUS para comunicación Ethernet entre PLCs y dispositivos de campo. Las versiones antiguas no incluyen autenticación.
EtherNet/IP (CIP)	44818	Basado en TCP/IP; usado por Allen-Bradley, Rockwell y otros fabricantes. Facilita intercambio de datos entre controladores y sensores. Puede permitir acceso no autorizado si está expuesto.
OPC UA / OPC DA	4840 / 135	Plataforma estándar de interoperabilidad industrial. OPC UA incluye autenticación y cifrado, pero a menudo se despliega sin configurarlos.
MELSEC / MC Protocol	5007, 5006	Protocolo propietario de Mitsubishi para controladores industriales. Permite lectura/escritura de registros PLC. No tiene seguridad por defecto.
TriStation / Triconex	1502	Usado en sistemas de control de seguridad industrial (SIS). Vulnerable si se expone: permite comunicación con controladores críticos.
CODESYS	2455	Entorno común en PLCs de varios fabricantes; utiliza servicios TCP sin cifrado por defecto. Vulnerabilidades documentadas (p. ej. CVE-2021-30186).
Niagara AX / N4	1911 / 4911	Plataforma de control para edificios inteligentes (BMS). Paneles web de administración accesibles con frecuencia; riesgo por contraseñas por defecto.

Continúa en la siguiente página

Continuación de la tabla 1

Protocolo	Puerto	Descripción breve / uso principal
MQTT (uso industrial)	1883 / 8883	Protocolo ligero publish/subscribe usado en IIoT. En entornos industriales transmite telemetría; sin TLS es fácilmente interceptable.
CoAP (IIoT)	5683	Protocolo REST para dispositivos con recursos limitados. Puede usarse en gateways industriales IoT. Sin DTLS, los datos viajan en claro.
SNMP	161 / 162	Protocolo de gestión de red. Versiones 1 y 2c usan contraseñas en texto claro; versión 3 añade cifrado y autenticación.
TFTP	69	Empleado para cargar firmware o configuraciones en equipos industriales. No incluye autenticación ni cifrado.
FTP / FTPS	21 / 990	Transferencia de archivos en sistemas de automatización. FTP transmite credenciales en claro; FTPS añade cifrado pero a menudo mal configurado.

Modbus es un protocolo de comunicaciones en serie publicado originalmente por Modicon (ahora Schneider Electric) en 1979 para su uso con sus controladores lógicos programables (PLC). Modbus se ha convertido en un protocolo de comunicación estándar de facto en los sistemas SCADA / ICS. Modbus permite la comunicación entre muchos dispositivos conectados a la misma red, por ejemplo, un sistema que mide la temperatura y la humedad y comunica los resultados a una computadora. Modbus se usa a menudo para conectar una computadora de supervisión con una unidad terminal remota (RTU) en Sistemas de control de supervisión y adquisición de datos (Supervisory Control and Data Acquisition, SCADA). Este protocolo usa el puerto 502.

Figura 5: Información adicional

Realiza la búsqueda referente a este protocolo. Indica toda la información que puedes extraer a través de las opciones de la figura 5.

¿Has encontrado la interfaz de login de usuario y contraseña en alguno de los resultados?
¿De qué dispositivo se trata?

Realiza la búsqueda correspondiente a dispositivos que usen el puerto 47808 restringiéndola a dispositivos en España.
¿Has encontrado la interfaz de login de usuario y contraseña en alguno de los resultados?
¿De qué dispositivo se trata?
Intenta encontrar con un buscador de propósito general las credenciales por defecto.
¿Dónde está ubicado?

El resto de la práctica consiste en seleccionar 3 búsquedas del listado que se presenta a continuación. Debes ejecutarlas e incluir en un informe la descripción de los resultados obtenidos incluyendo el tipo de dispositivo buscado y para qué se utiliza. Es probable que para contestar tengas que usar también un buscador de propósito general con diferentes palabras claves como puede ser el fabricante o vendedor del dispositivo e incluso consultar los manuales o datasheets disponibles en Internet. Usa el modelo de tabla siguiente para incluirlas en el informe. Usa una tabla para cada dispositivo.
En caso de encontrar una pantalla de acceso intenta conseguir información para conseguir acceso al dispositivo.

1. dispositivos webcamxp.
2. vulnerabilidad cve-2014-0160
3. dispositivos ExacqVision
4. dispositivos JUNG KNX
5. dispositivos iKettle
6. html:Softneta
7. http.title:"Tesla PowerPack System"
8. dispositivos Linksys WVC80N
9. DICOM
10. P372 ANPR enabled"

5. Búsquedas automáticas

Shodan permite automatizar las búsquedas a través APIs. (Para más información consultar: ¿Qué es una API?). Se pueden usar los lenguajes Python, Ruby, PHP, C#, Go, Haskell, Java, Node.js, Perl, PowerShell, and Rust. Además se puede también usar una REST API adicional para recopilar información sobre los exploits disponibles para las vulnerabilidades detectadas.

Seguidamente se incluye un ejemplo en Python basado en el que aparece en la referencia que se incluye a continuación.

Antes de ejecutarlo hay que sustituir “INSERT HERE YOUR API KEY” por la clave asociada la cuenta de usuario que han generado. La pueden encontrar entre las opciones de la cuenta 7.

Ficha para reportar la información encontrada con Shodan

Descripción	
Parámetros de búsqueda	
# de resultados	
Parámetros de acceso por defecto	

Hallazgos

IP:	
Organización:	
Localización:	
Puerto:	
URL:	
Observación:	
Evidencias:	
Observación	

Figura 6: Tablas para registrar las evidencias.

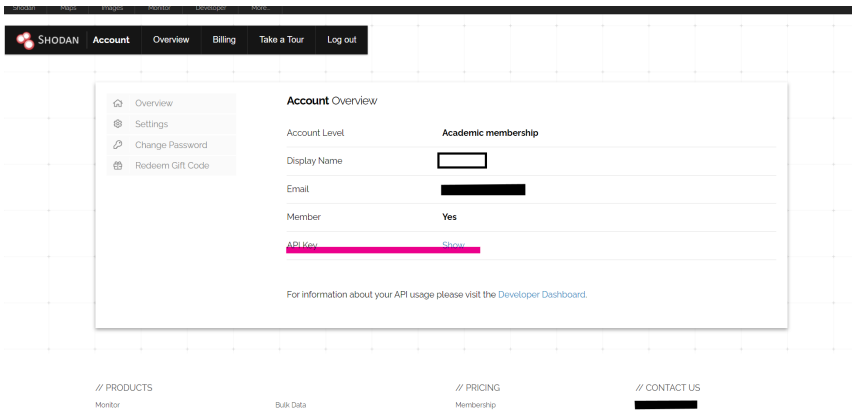

Figura 7: Acceso a la clave de la API

```
1  import shodan
2  from time import sleep
3
4  SHODAN_API_KEY = "INSERT HERE YOUR API KEY
5  api = shodan.Shodan(SHODAN_API_KEY)
6
7  query = 'medical'
8
9  try :
10
11   # Step 2 - Search using Shodan API
12   results = api.search(query)
13   print('Total number of results: {}'.format(results['total']))
14
15   for result in results ['matches']:
```

```
16
17      # Step 3 - Print IP and country for every obtained result
18      print('IP: {}'.format(result['ip_str'])) # The IP for each result is
        printed
19      print(result['data']) # To print raw data for each result
20      host = api.host(result['ip_str'])
21      print('- Country: {0}'.format(host.get('country_name', 'n/a')))
22      print('')
23      sleep(1) # A 1- second delay is necessary to respect Shodan API
      restrictions
24
25      # Step 4 - For each device IP , vulnerabilities and exploits are
      listed
26      try :
27        if str(host.get('vulns')) != 'None':
28        print('-------------------- Exploit list --------------------')
29        for vulnerability in host.get('vulns'):
30        exploits = api.exploits.search(vulnerability)
31        sleep(1)
32        print('Found {0} exploits for vulnerability "{1}" \n'.format(
33          exploits.get('total'), vulnerability))
34    except shodan.APIError as erro:
35      print('Error during exploit query: "{0}"'.format(query))
36      print('Shodan error: {0} '.format(erro))
37
38  except shodan.APIError as e:
39  print ('Error: {} '.format(e))
```

6. Línea de comandos

La interfaz de línea de comandos (CLI) de Shodan está incluida con la biblioteca oficial de Python para Shodan. Consulta `https://cli.shodan.io/` para más detalles. Igual que sucede cuando se usa la API de Shodan, también en este caso debes inicializar el entorno con la Clave de la API de usuario. `shodan init YOUR_API_KEY`

En `https://cli.shodan.io/#commands` puedes ver alguno de los comandos más utilizados.

7. Referencias

Fernández-Caramés, T.M.; Fraga-Lamas, P. Teaching and Learning IoT Cybersecurity and Vulnerability Assessment with Shodan through Practical Use Cases. Sensors 2020, 20, 3048.