CEA Comité Español de Automática
XLV JORNADAS AUTOMÁTICA MÁLAGA 2024
UNIVERSIDADE DA CORUÑA

Jornadas de Automática

Herramienta de disección de tramas para protocolos IoT

Narciandi-Rodríguez, Diego[a], Aveleira-Mata, Jose[b,*], Merayo Corcoba, Alicia[c], Rubiños, Manuel[d], Arcano-Bea, Paula[d], Alaiz-Moretón, Héctor[a]

[a] *RIASC. Instituto de Ciencias Aplicadas a la Ciberseguridad. Universidad de León, MIC. Campus de Vegazana, s/n, León, 24071, Spain*
[b] *SECOMUCI Group de Investigación. Dept. de Ingeniería Eléctrica y de Sistemas y Automática. Universidad de León., Escuela de Ingenierías. Campus de Vegazana, s/n, León, 24071, Spain*
[c] *Dept. de Ingeniería Eléctrica y de Sistemas y Automática, Universidad de León, Campus of Vegazana s/n, León, Spain*
[d] *Dept. de Ingeniería Industrial, Universidad de A Coruña, CTC, CITIC Research, Rúa Mendizábal, Ferrol, A Coruña, Spain*

To cite this article: Narciandi-Rodríguez,Diego., Aveleira-Mata, Jose., Merayo Corcoba, Alicia., Rubiños, Manuel., Arcano-Bea, Paula., Alaiz-Moretón, Héctor. 2024. Frame Dissection Tool for IoT Protocols. Jornadas de Automática, 45. https://doi.org/10.17979/ja-cea.2024.45.10804

Resumen

Desde hace unos años la aparición y uso de dispositivos IoT (Internet de las Cosas), los cuales destacan por el uso de protocolos ligeros debido a su baja carga computacional, hace que surgan nuevos vectores de ataque en los sistemas con dispositivos IoT. Es por ello que es necesario entrenar y desarrollar modelos de aprendizaje automático a partir de datos reales, que se implementen en sistemas de deteccion de intrusiones (IDS). Aquí es donde intervienen los datasets los cuales posibilitan esta actividad gracias al desarrollo efectivo de estos modelos. En este trabajo se presenta el desarrollo de un disector de tramas que facilita la generación datasets específicos para los diferentes protocolos IoT existentes que sean útiles para crear modelos de aprendizaje automático a partir de los mismos.

Palabras clave: Control de las redes, Sistemas de control de tráfico, Sistemas en red, Internet de las cosas, Codiseño de software

Frame Dissection Tool for IoT Protocols

Abstract

In recent years, the emergence and use of IoT (Internet of Things) devices, which stand out for their use of lightweight protocols due to their low computational load, has led to the emergence of new attack vectors in systems with IoT devices. This is why it is necessary to train and develop machine learning models from real data, which are implemented in intrusion detection systems (IDS). This is where datasets come in, which make this activity possible thanks to the effective development of these models. This paper presents the development of a frame dissector that facilitates the generation of specific datasets for the different existing IoT protocols that are useful to create machine learning models from them.

Keywords: Control of networks, Traffic control systems, Networked systems, Internet of Things, Arquitectura de software de control

1. Introducción

El término Internet de las cosas (en inglés Ïnternet of Things", IoT) hace referencia a la conexión a Internet de objetos cotidianos. Especialmente utilizados en el área doméstica, sector industrial (Industria 4.0) y sector sanitario (IoMT, Internet of Medical Things) de tal manera que ofrece nuevas funcionalidades al interactuar con sus sensores o actuadores. Todo dispositivo IoT destaca por una electrónica de pequeño tamaño, eficiencia energética, una baja capacidad de cómputo y el uso de protocolos de red ligeros, lo cual repercute negativamente en la seguridad del dispositivo y sus conexiones,

*Autor para correspondencia: jose.aveleira@unileon.es

convirtiendo a estos entornos en el objetivo de los ataques más grandes que se producen en la actualidad. Un incidente de ciberseguridad relacionado con las características de estos dispositivos es el software Mirai el cual provocó la caida de varios servicios de Internet usando dispositivos IoT zombie para realizar DDoS (Distributed Denial of Services) (Ibrahim et al., 2022).

Para abordar esta problemática, se están empezando a integrar los sistemas de detección de intrusiones o IDS, por sus siglas en inglés. Estos sistemas son una excelente alternativa para el análisis de tráfico de red sin alterar ni ralentizar el curso normal de la red.

Los sistemas IDS son una solucion viable ya que al centrarse en el analisis del trafico de red no es necesario cambiar los dispositivos ni las configuraciones existentes. Estas herramientas son cada vez más utilizadas en empresas e instituciones ante cualquier tipo de infraestructura o topología de red (Liao et al., 2013). Estas herramientas realizan una detección de patrones en el tráfico de la red que son susceptibles de ser maliciosos, con el fin de lanzar alarmas o realizar acciones de mitigación.

Los IDS utilizan principalmente dos métodos de detección: detección basada en la firma y detección basada en anomalías Khraisat et al. (2019). Los métodos de detección de firmas utilizan huellas digitales de amenazas conocidas para identificarlas. El método de detección de anomalías crea un modelo de comportamiento "típico" del sistema. Cualquier comportamiento anómalo se etiqueta como amenaza potencial y genera alertas. También existen técnicas que utilizan la inteligencia artificial para detección de amenazas. Buscando una mejora de los métodos de detección, ya sea para crear nuevas reglas o adaptar patrones de detección en el IDS. Para la implementación de técnicas basadas en técnicas de aprendizaje automático, como pueden ser modelos de aprendizaje automático, es imperativo desarrollo y utilizar conjuntos de datos o datasets con tráfico malicioso y normal (Singh and Khare, 2022).

La implementación de modelos de aprendizaje automático es muy interesante ya que mitigan en gran medida diferentes deficiencias y limitaciones que presentan los IDS como pueden ser:

- Los sistemas IDS basados en firmas, utilizan un conjunto de reglas predefinidas para detectar los ataques, cuantos más ataques detectan más complejas y extensas son estas reglas. Por lo que su mantenimiento y actualización se vuelven en muchos casos casi imposible. Siendo necesario el uso de otras herramientas o métodos complementarios, como puede ser la generación de reglas dinámicas (Otoum and Nayak, 123).

- En IoT hay muchas amenazas desconocidas. Los IDS basados en patrones no son capaces de detectarlos y los IDS basados en anomalías requieren de la creación de un modelo, por lo que será necesario un dataset. Para la prueba de concepto del IDS desarrollado, se utiliza el protocolo MQTT para crear el dataset y realizar modelos de aprendizaje automático, debido a ser un protocolo muy ligero y ampliamente utilizado en entornos IoT (Naidu and Kumar, 2019; Rizos et al., 2020).

  Un fichero .pcap es un tipo de extensión usada por analizadores de red, como Tcpdump, lipcap o Wireshark que contienen el tráfico existente entre dos periodos de tiempo. Por ello, para desarrollar un análisis de tráfico, es necesario separar la información del archivo en formato "pcap", por los campos relevantes del tráfico por cada protocolo IoT analizado. Wireshark ofrece la posibilidad de desarrollar un disector mediante el lenguaje de programación LUA (Mika, 2024). En cambio, un disector desarrollado con LUA no ofrece la versatilidad de alterar la combinación de campos por cada dataset. Por dicho motivo, se ha desarrollado una herramienta que se adapte a las necesidades específicas. Otra opción consiste en realizar diferentes scripts, pero tienen complicaciones al escalar en cuanto mantenimiento y actualización más complejos, cuando se quieren analizar gran cantidad de protocolos.

  En entornos IoT, la necesidad de datasets de calidad es cada vez más evidente, especialmente debido a la existencia de diversos modelos de aprendizaje automático dedicados a la detección y clasificación de ataques. El uso de disectores de tramas resulta interesante en este contexto, ya que facilitan el desarrollo de estos datasets para la seguridad en dichos entornos.

  De esta necesidad, en la actualidad ya se están utilizando diferentes datasets para el desarrollo y entrenamiento de modelos de detección de anomalías, destacando conjuntos de datos como KDD99 y UNSW-NB15 (of California, 1999; Moustafa, 2017), que incluyen ataques DoS genéricos aunque no están específicamente enfocados en IoT. También se utilizan datasets como MQTT-IoT-IDS2020 y MQTT-set (Hanan et al., 2020; security group CNR-IEIIT, 2021) específicamente diseñados con tráfico MQTT simulado, cubriendo ataques de escaneo de redes, fuerza bruta y DoS. Los datasets Bot IoT y TON_IoT también contribuyen con registros de ataques de botnets, DoS y DDoS, entre otros.

  El uso de un disector de tramas falicita la creación de datasets debido a la existencia de un gran número de protocolos IoT diferentes, resultado de la gran heterogeneidad de los sistemas IoT.

2. Desarrollo de la herramienta

El software desarrollado se ha llamado "Web dissector", una herramienta flexible para diseccionar las tramas capturadas de protocolos IoT y otros campos de interés que permite el etiquetado de las tramas, facilitando la categorización de tramas que forman parte de un ataque o no.

La herramienta ha sido desarrollada en PHP. Lenguaje seleccionado debido al fácil tratamiento de cadenas, lo que simplifica la estructuración de los datos de cada trama. Además de permitir que funcione como una aplicación web, que se ejecuta desde un servidor XAMPP, lo que permite su acceso a través de navegadores web. El núcleo de la funcionalidad de "Web dissector" es diseccionar cada una de las tramas, según los campos requeridos en cada caso, para ello se integra la herramienta de línea de comandos Tshark, debido a que permite

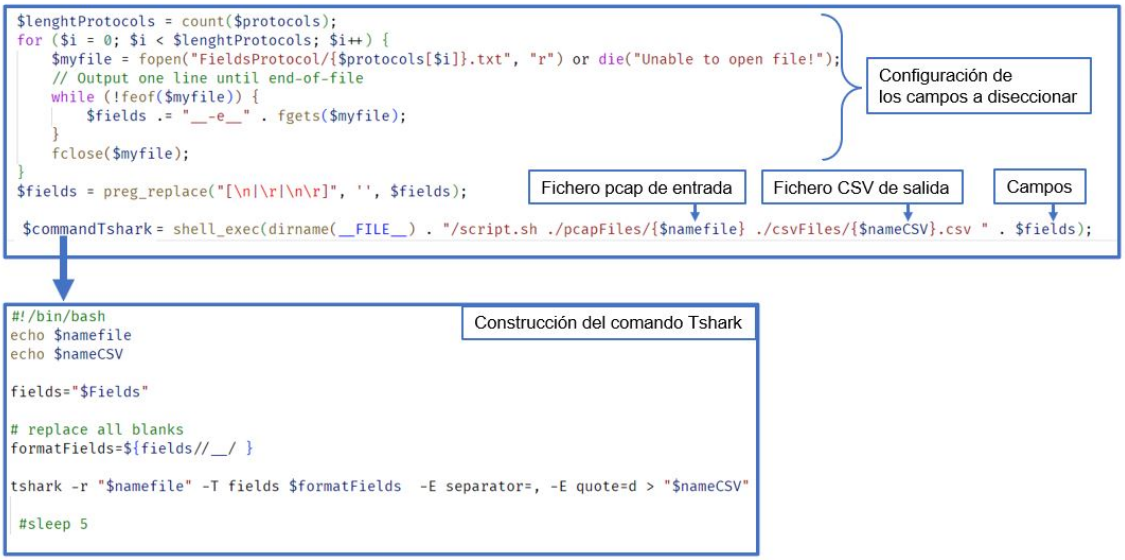

Figura 1: Construcción del comando Tshark de forma dinámica

filtrar paquetes indicando los campos, debido a que hacerlo con comandos manuales serían muy pesado y costoso.

En la Figura 1 se muestra el código creado para que el comando Tshark componga las tramas, seleccionando los campos según las referencias recibidas. Las tramas se extraen de los archivos ”pcap”.

Para la selección de los campos a diseccionar, se toma como referencia el trabajo realizado por (Chatzoglou et al., 2021) al realizar el AWID dataset, que recoge ataques en las redes IEEE 802.11, donde se centra en comunicaciones de diferentes protocolos de comunicación WiFi.

Se toma la estructura de las tramas y se seleccionan varios campos comunes a todas ellas, el resto de los campos se seleccionan teniendo en cuenta el tipo de ataque realizado. Se utiliza la misma metodología del trabajo mencionado, para la selección de campos comunes a todas las tramas, donde se adquiere información relevante, como pueden ser los puertos, las direcciones MAC y los timestamps, los cuales se observan en la Tabla 1.

Como particularidad de nuestro disector, se deben seleccionar todos los campos específicos del protocolo IoT analizado debido a su relevancia en la detección de anomalías.

Se diseccionan teniendo en cuenta los respectivos ficheros para la configuración de campos. Es posible agrupar los campos en varios ficheros de configuración (dentro del directorio “FIeldsProtocol”). Cuenta un fichero de tramas comunes (teniendo en cuenta el trabajo de AWID) y el resto de los ficheros están formados uno por cada protocolo IoT estudiado, dando la opción de elegir entre uno o varios ficheros. Lo que permite combinar estos campos según las necesidades del análisis de la red. Tras la composición de las tramas diseccionadas empleando los campos especificados, se procede a la generación de un archivo en formato CSV (valores separados por comas).

Tabla 1: Campos comunes a las tramas

Nombre del campo	Descripción	Tipo
frame.time_delta	Tiempo delta desde la trama capturada anteriormente	Desfase temporal
frame.time_delta_displayed	Tiempo delta desde la trama visualizada anteriormente	Desfase temporal
frame.time_invalid	Etiqueta un tiempo como invalido	Etiqueta
frame.time_relative	Tiempo transcurrido desde la primera trama	Desfase temporal
ip.src	Dirección IP origen	Dirección IPv4
ip.dst	Dirección IP destino	dirección IPv4
tcp.srcport	Puerto de origen	Entero sin signo,2 bytes
tcp.dstport	Puerto de destino	Entero sin signo,2 bytes
eth.src	Dirección MAC de origen	Ethernet u otra dirección MAC
eth.dst	Dirección MAC de destino	Ethernet u otra dirección MAC
frame.cap_len	Longitud de la trama capturada	Entero sin signo, 4 bytes
frame.coloring_rule.name	Regla de color para el nombre	Cadena de caracteres
frame.coloring_rule.string	Regla de color para la cadena	Cadena de caracteres
frame.comment	Comentarios de la trama	Cadena de caracteres
frame.comment.expert	Comentarios expertos	Etiqueta
frame.encap_type	Tipo de encapsulamiento	Entero con signo, 2 bytes
frame.file_off	Desplazamiento de los ficheros	Entero con signo, 8 bytes
frame.ignored	Si la trama es ignorada	Booleano
frame.incomplete	Si la trama está incompleta	Etiqueta
frame.interface_id	Identificador de la interfaz	Entero sin signo, 4 bytes
frame.interface_name	Nombre de la interfaz	Cadena de caracteres
frame.len	Longitud de la trama por cable	Entero sin signo, 4 bytes
frame.link_nr	Numero de enlace	Entero sin signo, 2 bytes
frame.marked	Si la trama está marcada	Booleano
frame.md5_hash	MD5 Hash de la trama	Cadena de caracteres
frame.number	Numero de trama	Entero sin signo, 4 bytes
frame.offset_shift	Tiempo de desplazamiento de la trama	Desfase temporal

En ocasiones, por las características de ataques como el de DoS se produce gran cantidad de tráfico en un corto periodo de tiempo y las marcas de tiempo o timestamps se solapan, por lo que se hace necesario realizar un desanidado, mediante la función “analisisTrama”. En la Figura 2 se puede observar la función que toma el fichero generado por el comando Tshark y formatea separando en líneas diferentes cada una de las tramas que se solapan por tener el mismo timestamp.

```php
function analisisTrama($trama, $times) {
    $tramas = str_getcsv($trama);
    $lenght = count($times);
    $timeTrama = floatval($tramas[2]);
    $linea = "";
    $comas = 0;
    $ncomas = -1;
    for ($m = 0; $m < count($tramas); $m++) {
        $ncomas = count(explode(",", $tramas[$m]));
        if ($ncomas > $comas) {
            $comas = $ncomas;
        }
    }
    for ($i = 0; $i < $lenght; $i = $i + 2) {
        if ($timeTrama >= $times[$i] && $timeTrama <= $times[$i + 1]) {

            $lineaEspecial = "";
            if ($comas > 1) {

                for ($j = 0; $j < $comas; $j++) {
                    $concat = '';
                    for ($m = 0; $m < $comas; $m++) {
                        $desanidador = explode(",", $tramas[$m]);
                        if (isset($desanidador[$j])) {
                            $tramas[$m] = $desanidador[$j];
                        }

                    }

                    $lineaEspecial .= $concat;
                }
                return $lineaEspecial;
            }
        }
    return $linea;
}
```

Detección de varias tramas dentro de la misma marca de tiempo

Separación de las Tramas anidadas

Figura 2: Función de análisis de tramas

Tras el proceso de formateo del fichero CSV, es posible crear una nueva etiqueta (type) para añadir información adicional como el tipo de ataque. En caso de etiquetar las tramas implicadas en un ataque, es necesario guardar el ”timestamp” en un fichero de texto, con el inicio y fin de cada ataque. La herramienta recupera las marcas de tiempo del fichero, cotejando el campo ”time_epoch” para etiquetar las tramas que se encuentran dentro del intervalo. Las tramas se clasifican en dos categorías: tramas normales y tramas asociadas a un ataque.

En la Figura 3 se representa el diagrama que describe el flujo de trabajo de “Web dissector”, donde se tiene como entrada un fichero en formato “pcap”, compuesto por un comando que utiliza el fichero y los campos a diseccionar que desee el usuario, el comando genera un fichero CSV, sobre el cual se realiza un formateo y un etiquetado. Finalmente, el usuario podrá descargar el fichero en formato CSV con cada trama debidamente etiqueta.

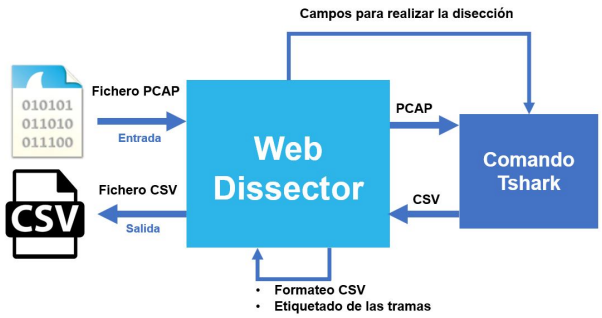

Figura 3: Flujo de trabajo de Web dissector

Se utiliza Bootstrap para el diseño web responsive, el cual se adapta automáticamente al tamaño de la pantalla del dispositivo. Esto garantiza una experiencia de usuario consistente y atractiva en todos los dispositivos. La interfaz está compuesta por un formulario para la que se pueda subir el fichero “pcap”, donde se especifica el nombre del documento CSV de salida y una lista de opciones para seleccionar los criterios que se utilizarán para diseccionar cada trama del documento. Una vez generado el documento CSV se mostrará en un listado dentro de la sección “CSV Files”, donde aparece la opción de ser descargado de forma local, en la Figura 4 se puede ver una captura de pantalla de esta interfaz.

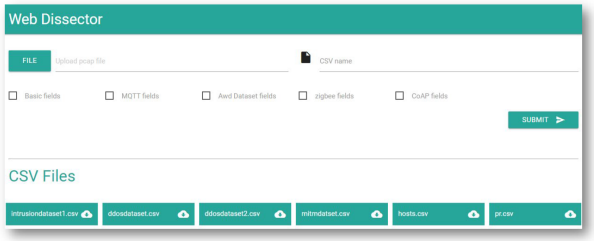

Figura 4: Función de análisis de tramas

3. Funcionamiento de la herramienta

Adentrandonos ahora en el funcionamiento del propio disector web. Cabe destacar que el funcionamiento de la herramienta se reduce exclusivamente a la interacción con la interfaz de la Figura 4 donde se puede observar los siguientes campos.

En primer lugar, se seleccionan, como se cometó anteriormente, un fichero .pcap el cual tenga los diferentes ataques y mensajes convencionales (no reflejan ataques). Tras cargar dicho fichero se selecciona los campos que se va a almanecar en el dataset, marcando los checkbox que aparecen en la parte directamente inferior al campo de carga del fichero .pcap.

Se indica ahora el nombre del fichero csv donde se alamcena el dataset y presionando el botón ”SUBMIT”. A continuación nos aparecerá en ”CSV Files” el fichero del dataset, el cual se puede descargar presionando sobre el mismo.

4. Resultados

El disector se ha testeado sobre tres protocolos distintos MQTT, CoAP y Zigbee aunque es extrapolable a cualquier otro protocolo, teniendo en cuenta el filtro de referencia de wireshark (wireshark, 2024).

A través del uso del disector, se han desarrollado varios datasets para el protocolo MQTT. Estos datasets se han convertido en una herramienta valiosa para la investigación en el ámbito de las comunicaciones IoT. Gracias a la recolección y análisis de los datos proporcionados por este disector, los investigadores han podido explorar diversas áreas como la seguridad y la clasificación de anomalías en redes IoT que utilizan MQTT (MQTT_UAD, 2019).

Los datasets derivados para MQTT han sido ampliamente utilizados en la literatura científica, contribuyendo significativamente a avanzar en el conocimiento de este protocolo. Estos datasets han servido como base para experimentos y han sido citados en diversos artículos científicos, destacando su importancia y utilidad en el campo. (Alaiz-Moreton et al., 2019)

Similar al caso de MQTT, el disector web también ha sido utilizado para el desarrollo de datasets del protocolo CoAP, empleado comúnmente en entornos IoT limitados por recursos. Estos datasets han proporcionado información crucial para estudios relacionados con la eficiencia del protocolo, la se-

guridad, y la adaptabilidad en redes con restricciones de ancho de banda y energía.

Al igual que con MQTT, los conjuntos de datos de CoAP, que incluyen ataques a vulnerabilidades de este protocolo definidas en su RFC Shelby et al. (2014), han sido utilizados en diferentes investigaciones para entrenar modelos aplicables a sistemas de detección de intrusos (IDS) Álvaro Michelena et al. (2023); Timiraos et al. (2023).

En cuanto al protocolo ZigBee, utilizado predominantemente en aplicaciones de domótica y sistemas de sensores, el disector web ha facilitado la creación de un dataset prometedor que actualmente está siendo explorado para su uso en investigaciones académicas. Aunque este dataset está en las etapas iniciales de su aplicación en estudios científicos, ya ha sido catalogado y es accesible mediante un identificador de objeto digital (DOI), asegurando su disponibilidad y referencia para futuras investigaciones.

5. Conclusiones y lineas futuras

Los sistemas IoT ofrecen nuevos retos en cuanto a ciberseguridad debido a sus características de poca capacidad de computo para ser mas eficientes, por lo que utilizan unos protocolos de comunicación mas ligeros, el gran crecimiento de estos sistemas y la gran variedad de protocolos diferentes hacen que surjan nuevos vectores de ataque.

Con la herramienta presentada en este sistema se ha conseguido automatizar de una manera dinámica y escalable la disección y etiquetado (para indicar los diferentes ataques que aprovechen las vulnerabilidades de los protocolos) y así facilitar la creacion de nuevos datasets de calidad gracias a estos datasets es posible mejorar los modelos de aprendizaje automático para IDS.

Es por ello que esta facilita el desarrollo de conjuntos de datos y sistemas de ciberseguridad. Como trabajos futuros se plantea seguir manteniendo esta herramienta y agregar nuevas automatizaciones, además de añadir funcionalidades visuales para mostrar datos de interes de los datasets generados tales como: número de tramas que lo componen, tramas bajo ataque, tramas totales y campos por los que se ha diseccionado el dataset.

Agradecimientos

Quisiera agradecer a todas las personas que contribuyeron a la realización de este artículo. Además de expresar mi gratitud al Plan de Recuperación, Transformación y Resiliencia de la Unión Europea (Next Generation) que por medio del proyecto “Seguridad del Internet de las Cosas en Entornos Domésticos y Empresariales en el Contexto de la Tecnología 5G-IoT” han financiado este proyecto.

Referencias

Alaiz-Moreton, H., Aveleira-Mata, J., Ondicol-Garcia, J., Muñoz-Castañeda, A. L., García, I., Benavides, C., 2019. Multiclass classification procedure for detecting attacks on mqtt-iot protocol. Complexity 2019.
DOI: 10.1155/2019/6516253

Chatzoglou, E., Kambourakis, G., Kolias, C., 2021. Empirical evaluation of attacks against ieee 802.11 enterprise networks: The awid3 dataset. IEEE Access 9, 34188–34205.
DOI: 10.1109/ACCESS.2021.3061609

Hanan, H., Ethan, B., Miroslav, B., Robert, A., Christos, T., Xavier, B., 2020. Mqtt-iot-ids2020 dataset — papers with code.
URL: https://paperswithcode.com/dataset/mqtt-iot-ids2020

Ibrahim, Z. A., Razali, R. A., Ismail, S. A., Azhar, I. H. K., Rahim, F. A., Azilan, A. M. A., 2022. A review of machine learning botnet detection techniques based on network traffic log. 2022 IEEE International Conference on Computing, ICOCO 2022, 204–209.
DOI: 10.1109/ICOCO56118.2022.10031803

Khraisat, A., Gondal, I., Vamplew, P., Kamruzzaman, J., 2019. Survey of intrusion detection systems: techniques, datasets and challenges. The 14th International Conference on Ambient Systems, Networks and Technologies (ANT), March 15-17, 2023, Leuven, Belgium.
URL: https://doi.org/10.1186/s42400-019-0038-7
DOI: 10.1186/s42400-019-0038-7

Liao, H. J., Lin, C. H. R., Lin, Y. C., Tung, K. Y., 1 2013. Intrusion detection system: A comprehensive review. Journal of Network and Computer Applications 36, 16–24.
DOI: 10.1016/J.JNCA.2012.09.004

Mika, 2024. Creating a wireshark dissector in lua - part 1 (the basics) — mika's tech blog.
URL: https://mika-s.github.io/wireshark/lua/dissector/2017/11/04/creating-a-wireshark-dissector-in-lua-1.html

Moustafa, N., 2017. Designing an online and reliable statistical anomaly detection framework for dealing with large high-speed network traffic.
URL: http://hdl.handle.net/1959.4/58748
DOI: 10.26190/UNSWORKS/3298

MQTT_UAD, 2019. Mqtt_uad: Mqtt under attack dataset. a public dataset for the detection of attacks in iot networks using mqtt.
URL: https://figshare.com/s/2036c5c56ce6a3fc1191

Naidu, G. A., Kumar, J., 2019. Wireless protocols: Wi-fi son, bluetooth, zigbee, z-wave, and wi-fi. Lecture Notes in Networks and Systems 65, 229–239.
URL: https://www.researchgate.net/publication/330927333_Wireless_Protocols_Wi-Fi_SON_Bluetooth_ZigBee_Z-Wave_and_Wi-Fi
DOI: 10.1007/978-981-13-3765-9_24

of California, U., 1999. Kdd cup 1999 data.
URL: http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html

Otoum, Y., Nayak, A., 123. As-ids: Anomaly and signature based ids for the internet of things keywords internet of things (iot) security · anomaly-based ids · signature-based ids · deep q-learning · lightweight neural network (lightnet). Journal of Network and Systems Management 29, 23.
URL: https://doi.org/10.1007/s10922-021-09589-6
DOI: 10.1007/s10922-021-09589-6

Rizos, A., Bastos, D., Saracino, A., Martinelli, F., 2020. Distributed ucon in coap and mqtt protocols. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) 11980 LNCS, 35–52.
DOI: 10.1007/978-3-030-42048-2_3

security group CNR-IEIIT, N., 2021. Mqttset.
URL: https://www.kaggle.com/datasets/cnrieiit/mqttset

Shelby, Z., Hartke, K., Bormann, C., 6 2014. The constrained application protocol (coap).
URL: https://www.rfc-editor.org/info/rfc7252
DOI: 10.17487/RFC7252

Singh, G., Khare, N., 2022. A survey of intrusion detection from the perspective of intrusion datasets and machine learning techniques. International Journal of Computers and Applications 44, 659–669.
DOI: 10.1080/1206212X.2021.1885150

Timiraos, M., Michelena, Á., Díaz-Longueira, A., Jove, E., Aveleira-Mata, J., García-Rodiguez, I., Bayón-Gutiérrez, M., Alaiz-MoretÓn, H., Calvo-Rolle, J. L., 2023. Categorization of coap dos attack based on one-class boundary methods. In: García Bringas, P., Pérez García, H., Martínez de Pisón, F. J., Martínez Álvarez, F., Troncoso Lora, A., Herrero, Á., Calvo Rolle, J. L., Quintián, H., Corchado, E. (Eds.), 18th International Conference on Soft Computing Models in Industrial and Environmental Applications (SOCO 2023). Springer Nature Switzerland, Cham, pp. 112–121.

wireshark, 2024. Wireshark · display filter reference: Index.

URL: https://www.wireshark.org/docs/dfref/

Álvaro Michelena, Díaz-Longueira, A., Timiraos, M., Jove, E., Aveleira-Mata, J., García-Rodiguez, I., García-Ordás, M. T., Calvo-Rolle, J. L., Alaiz-Moretón, H., 2023. One-class reconstruction methods for categorizing dos attacks on coap. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) 14001 LNAI, 3–14.
URL: https://link.springer.com/chapter/10.1007/978-3-031-40725-3_1
DOI: 10.1007/978-3-031-40725-3_1