# *IoTAG*: An Open Standard for *IoT* Device Identific*A*tion and Reco*G*nition

Sebastian Fischer
Secure Systems Engineering
Fraunhofer AISEC
Berlin, Germany
email:
sebastian.fischer@aisec.fraunhofer.de

Katrin Neubauer*
and Rudolf Hackenberg†
Dept. Computer Science and Mathematics
Ostbayerische Technische Hochschule
Regensburg, Germany
email:
katrin1.neubauer@oth-regensburg.de*
rudolf.hackenberg@oth-regensburg.de†

Lukas Hinterberger‡
and Bernhard Weber§
Dept. Electrical Engineering and
Information Technology
Ostbayerische Technische Hochschule
Regensburg, Germany
email:
lukas.hinterberger@st.oth-regensburg.de‡
bernhard1.weber@st.oth-regensburg.de§

***Abstract*—With the increasing amount of Internet of Things (IoT) devices in smart homes, insecure and old devices are leading to big security issues. A private network can be attacked over an insecure IoT device, to use it in a botnet or infect it with ransomware and compromise the whole network. Non-technical users do not know which devices in their homes are secure and how to keep track of all the old and new ones. We have built a typical smart home as a test environment to evaluate a scoring system for the security of the whole network. First, all devices are discovered with nmap and then all the possible information, like the open ports or the Wi-Fi technology, are retrieved. In the next step, all the information leads to an overall score for each device. Combined together, the final score for the whole network is created. A non-technical user can now determine, if the network is secure or not. We show the proof of concept of the scoring system with our test environment. However, some challenges exist. Not all information can be retrieved by just scanning the devices over the network. Some devices just return hostnames like "ESP_6A786B". It is nearly impossible to tell the kind of device and the manufacturer. Additionally, no information about the running firmware is provided. To calculate a meaningful score, much more information has to be collected. To collect the missing data, we introduce the first version of a new, open standard for *IoT* Device Identific*A*tion and Reco*G*nition (IoTAG). This JSON based model provides all the important information about the device. Besides the device name, type and the manufacturer, it shows a list of the services, the firmware version and the supported encryption. IoTAG allows to create an overview of the whole IoT network and the development of an automated scoring system. In the future, additional information about security vulnerabilities can be collected from the Internet, to warn the user about insecure devices.**

***Keywords*—*Internet of Things; device identification; open standard; IoTAG; security rating.***

## I. Introduction

Internet of Things (IoT) is an ongoing innovation and trend in nearly all industries and smart homes. The development is extremely fast and most of the time, the security risks of IoT networks are underestimated or not even taken into account at all. This leads to insecure devices, e.g., with missing encryption or authentication. Overall, a large number of IoT devices in general, are critical to operate. Some risks are comparable harmless attacks, which just destroy the device, but others can lead to hijacking of complete company networks [1] [2].

To avoid these problems, the user should be able to tell which devices are in the network and if they are running with the latest software. Currently, there are no existing systems for automated device scanning. It is possible to obtain parts of the required information in single steps. For example, the network scanners Nmap [3] or Fing [4] can be used for finding addressable network ports. But the results of this scans will not be analyzed or evaluated. To help a non-technical user, an easy to use scoring system for IoT devices is necessary.

The first scan of a network detects all the containing IoT devices. Each detected device gets a security rank based on the provided meta data, information collected by the scanner itself and a database of known vulnerabilities, which are collected from multiple publicly available sources. All the ranks together will provide an overall network rank. The scanner should be able to show the rank, a list of all known vulnerabilities and general risks of the IoT setup to the user.

The goal of this project is to identify requirements for the development of a standard, which provides the needed metadata and also checks the authenticity of the received information. In this paper, we present the first version of IoTAG. The paper is structured as follows. Section II describes the related work. Section III introduces our hardware setup and device scanning, while Section IV defines the security criteria. Section V shows the device rating and Section VI the results. The standard IoTAG is presented in Section VII, followed by a conclusion in Section VIII.

## II. Related Work

One possible solution for IoT device identification uses device fingerprints. Miettinen et al. [5] are categorizing and classifying (secure and insecure) IoT devices by device fingerprint. Another research project [6] is developing a sys-

 ISBN: 978-1-61208-746-7 

tem for anomaly recognition (smart home networks). There are several publications [7]–[10] covering the subject device identification with device fingerprints and similar approaches. These publications are demonstrating working approaches for the detection of IoT devices in a network. However, it is not possible to identify detailed information such as the current firmware version or a device ID for further recognition.

Some researchers provide mechanisms to evaluate the security and privacy for IoT devices with different security ratings. One very similar approach [11] uses protocols, open ports and the encryption to create the rating. But it is not very flexible and user-friendly because of the missing weighting of each criteria and the missing overall score of the network. Park et al. [12] and Ali et al. [13] are offering a very good approach for the focus of the risk, which can be used to evaluate the weighting. Both papers do not provide a rating, but a list of security requirements in IoT services. Another approach uses vulnerabilities and known exploits to generate a metric value for the security of an IoT device [14].

With the Device Description Language for the T in IoT from Khaled et al. [15] and the Thing Description as Enabler of Semantic Interoperability on the Web of Things from Kaebisch et al. [16], there are some publications, proposing a machine readable description for IoT devices. These descriptions are only for the functionality of a device and cover information like the turn off command. With IoTAG, we do not want to get the functions of a device, instead we want to get the security characteristics. It is possible to extend these descriptions with our IoTAG information.

This paper extends the initial work of Hinterberger, introducing the evaluation criteria and scanning methods for the device rating [17], with further research and the new IoTAG standard.

## III. Hardware Setup and Device Scanning

We have built a small smart home environment with ten devices, as seen in Table I and started a network scan to detect all the connected devices. Some devices reply with their hostname, but, in most cases, the response contains something like “ESP” or it is totally missing. In the next step, a deeper scan with “nmap -p 1-65535 192.168.0.0/24” is performed. Additional information about the devices on port 80 HTTP and a list of all open TCP (Transmission Control Protocol) and UDP (User Datagram Protocol) ports are shown in Table II. With this information, we can give more details about the running services and the device communication. For example, with an open port 80, an unencrypted connection is likely.

However, with all the given information, it is still impossible to detect the exact devices. The iPhone and Google Home mini are detectable with their hostname, but only if the hostname is not changed.

TABLE I. HARDWARE OVERVIEW

| device | hostname |
|---|---|
| Amazon Echo 2 | amazon-183e3c119 |
| Apple iPhone 5 | Kluges-iPhone |
| Floureon M32B | |
| Google Home mini | Google-Home-Mini |
| Grandstream GXP1610 | |
| Raspberry Pi 3 Model B | raspberrypi |
| Sonoff Wi-Fi Smart Switch | ESP_6A768B |
| Wi-Fi Smart Bulb | ESP_4C3210 |
| Wi-Fi Smart Plug | ESP_3D1EB6 |
| Wi-Fi Touch Switch | ESP_469ACF |

TABLE II. OVERVIEW OF OPEN AND RESTRICTED PORTS

| **Raspberry Pi 3 Model B** | | | | |
|---|---|---|---|---|
| **port** | | **state** | **service** | **reason** |
| 22 | TCP | open | ssh | syn-ack |
| 53 | TCP | open | domain | syn-ack |
| **Sonoff Wi-Fi Smart Switch** | | | | |
| **port** | | **state** | **service** | **reason** |
| | | restricted | | |
| **Wi-Fi Touch Switch** | | | | |
| **port** | | **state** | **service** | **reason** |
| 8081 | TCP | open | blackice-icecap | syn-ack |
| **Wi-Fi Smart Plug** | | | | |
| **port** | | **state** | **service** | **reason** |
| 10000 | TCP | open | snet-sensor-mgmt | syn-ack |
| **Grandstream GXP1610** | | | | |
| **port** | | **state** | **service** | **reason** |
| 22 | TCP | open | ssh | syn-ack |
| 80 | TCP | open | http | syn-ack |

## IV. Security Criteria

In order to define a test scheme that can be applied individually to any device, it is necessary to develop a procedure that allows the security risks to be assessed separately for each device. Afterwards, the individual assessments can be offset against each other in order to obtain the overall assessment of a device.

For the evaluation scheme, a three-level point system is defined as the basis for evaluation. If a security criterion is completely violated, the equipment in question is assessed zero points in that category. For non-critical violations one point and for no violations two points are awarded. Several individual evaluations are offset against each other by calculating an average value. It should be noted that individual categories can be weighted differently. The used security criteria are listed in Table III and described as follows in detail.

### A. Wi-Fi technology

As the encryption technology for wireless networks, the WPA2 (Wi-Fi Protected Access) and WPA3 standards are rated with the highest score. Networks based on the WPA or WEP (Wired Equivalent Privacy) standard cannot be classified as secure because the “RC4” encryption method used, is no longer state of the art and considered as broken [18].

### B. Services

This evaluation criterion deals with the services provided at network level and can be used to communicate with the respective device. In particular, it checks whether the communication procedures offered are based on encryption. The assessment is based on a presorting of known services and

TABLE III. SECURITY CRITERIA

| audit criteria | | | score |
|---|---|---|---|
| **radio technology** | | | |
| WPA/WEP or no encryption | | | 0 |
| WPA2/WPA3 | | | 2 |
| Bluetooth version | | | 0-2 |
| ZigBee version | | | 0-2 |
| **manufacturer** | | | |
| unknown manufacturer | | | 0 |
| usual patch time | | | 0-2 |
| experience | | | 0-2 |
| known unpatched devices | | | 0-2 |
| bug bounty program | | | 0/2 |
| **services** | | | |
| **service** | **default port** | **comment** | |
| HTTP | 80 | unencrypted login details | 0 |
| MQTT | 1883 | unencrypted control data | 0 |
| UPnP | 49152/1900 | firewall manipulation | 0 |
| rtsp | 554 | unencrypted video data | 0 |
| SIP | 5060 | unencrypted | 0 |
| **service** | **default port** | **comment** | |
| HTTPS | 443 | encrypted | 2 |
| MQTTS | 8883 | encrypted | 2 |
| SCP | 10001 | encrypted | 2 |
| SIPS | 5061 | encrypted | 2 |
| SSH | 22 | encrypted | 2 |
| **LAN and WAN communication** | | | |
| **service** | **default port** | **comment** | |
| HTTP | 80 | unencrypted login details | 0 |
| MQTT | 1883 | unencrypted control data | 0 |
| UPnP | 49152/1900 | firewall manipulation | 0 |
| rtsp | 554 | unencrypted video data | 0 |
| SIP | 5060 | unencrypted | 0 |
| **service** | **default port** | **comment** | |
| HTTPS | 443 | encrypted | 2 |
| MQTTS | 8883 | encrypted | 2 |
| SCP | 10001 | encrypted | 2 |
| SIPS | 5061 | encrypted | 2 |
| SSH | 22 | encrypted | 2 |
| **other** | | | |
| vulnerable to replay attacks | | | 0 |
| create own Wi-Fi | | | 0 |
| data retrieval without authentication | | | 0 |
| vulnerable to jamming | | | 0-2 |
| vulnerable to Denial of Service (DoS) | | | 0-2 |
| insecure configuration | | | 0 |
| continuous device number | | | 0-2 |
| known vulnerabilities | | | 0 |
| support lifetime | | | 0-2 |
| insecure / default password | | | 0/2 |
| software version | | | 0-2 |
| technical guidelines | | | 0-2 |
| certification | | | 0-2 |

protocols in black and white lists. Services on the black-list are rated with zero points, services on the white-list with two points and unknown services with one point.

### C. Communication

As with device services, device communication is tested for the use of encryption methods. Since the used protocols cannot be queried by scanning the devices, the current communication must be analyzed. In addition to the encryption technology, it is also possible to check the number of external resources a device communicates with and where they are located. Predefined protocol lists are also used for this evaluation criterion. The communication is separated in LAN (local area network) and WAN (Wide area network), to cover the different security requirements. In Table III, both are displayed in the same section.

### D. Default passwords

The use of standard passwords assigned by device manufacturers, that can be applied to multiple devices, is a major problem with the safety of IoT devices. It is important to check whether authentication on a device is possible using known passwords. In this case, the device is considered to be at risk and should therefore be evaluated with zero points.

### E. Firmware version

Known security vulnerabilities are often stored in public accessible databases and can be accessed by potential attackers. A known outdated software version of a device can be used for systematic attacks. It must be possible to check which software version is running on a device and whether updates are available for it. If no updates are available and security gaps are known for the existing software, the device must be classified as severely endangered. If updates are available but not installed, they are considered to be at risk, otherwise they are considered to be safe.

## V. Device Rating

In this section, we describe the proceeding to receive the information for all the security criteria and how they are rated in detail.

### A. Wi-Fi technology

The encryption technology of the wireless network can be queried in the router configuration. In the case of our experimental environment, the task of the router is taken over by a Raspberry Pi as Wi-Fi access point. The setup query is made via the configuration file of the access point software “hostapd”. Thus, the configuration is done in the file “/etc/hostapd”. The entry “wpa=2” indicates the exclusive use of the WPA2 standard. This leads to a score of two points for each device. If an unsafe technology is used, this will also affect the evaluation of each individual device, as the entire network will be endangered. In this case, all devices have to be rated with zero points in this category.

### B. Services

The running services are checked by scanning the network components. For this purpose, Nmap is used for both TCP and UDP connections [19]. The scan might produce the output shown in Table IV.

TABLE IV. PORT SCAN

| port | protocol |
|---|---|
| 22 | ssh |
| 80 | http |
| 5060 | sip |

Based on these results, the device can be rated. The already mentioned categorization lists are used. The example in Table IV leads to a rating with 0.66 points, because http and sip are rated with zero and ssh with two points.

### C. Communication

The communication of the devices to external resources is analyzed by recording and analyzing the network traffic. Existing technologies, like the tshark [20] software, are used. From the communication packets, the MAC address of the local resource, source and destination port, as well as the used protocols, are extracted. Incoming and outgoing traffic are handled separately. Analogous to the evaluation of the services, the evaluation of the communication is also based on predefined protocol lists. With the scan results in the output shown in Table V, the device will be rated with zero points in this category.

TABLE V. COMMUNICATION SCAN

| source device | destination port | protocol |
|---|---|---|
| 00:11:22:33:44:55 | 5060 | sip |

### D. Default passwords

In order to check whether an insecure password has been configured for a device, a dictionary attack against the corresponding device is carried out with the aid of the THC-Hydra [21] software. Both the user name and the password are attacked with known and frequently used terms. The required specification for which type of service a login should be performed, is taken from the previous service scan. The software tests all possible combinations with a brute force attack. If a device turns out to be vulnerable, it is highly vulnerable. Otherwise, it will be classified as harmless. If we consider an ssh login with “root” as the user and a well-known default password like “admin” as possible, this would lead to an rating with zero points.

An undefined handling of nonstandard, manufacturer-specific login procedures can lead to a problem with this kind of password check. For each specific procedure, a separate test algorithm must be developed, which may require adaptation after a software update by the device manufacturer. As an example of a manufacturer-specific login procedure, the challenge-response-mechanism that AVM uses for the Web interface of their Fritz!Box Routers can be mentioned [22].

### E. Firmware

It was not possible to develop an automated procedure for checking the firmware version, because of the lack of a standardized interface for querying information about the device software. The use of Nmap makes rough assumptions about the operating system of a device possible. But these are not sufficient for a valid risk assessment due to the gross inaccuracies. Furthermore, Nmap is only able to identify systems where an identification has already taken place [23]. It would be possible to create a Nmap fingerprint for each network device and include it in the database for system identification, but this procedure is not relevant in practice, as it requires specific knowledge of the software. Also, it is not guaranteed that the detection characteristics will not change after a software update, making it impossible to clearly determine the version. The same applies to independent test procedures, developed outside Nmap.

### F. Overall rating

After all ratings have been performed, an overall rating for a device can be calculated, by determining the average score. This score describes the vulnerability of a device based on Table VI. A ports score of 0.66 points, a communication score of 0.00 points and a password score of 0.00 points will lead to an overall device rating of 0.22 points and indicates a highly vulnerable device. The average score is used to compare the different devices. If we used the minimum score, each device would get zero points. Normally, the weakest point is attacked, but every missing or insecure security criteria does not necessary lead to a vulnerability.

TABLE VI. VULNERABILITY CATEGORIES

| score | category |
|---|---|
| 0.00 to 0.80 | high vulnerability |
| 0.81 to 1.80 | moderate vulnerability |
| 1.81 to 2.00 | small vulnerability |

## VI. RESULTS

To validate the concept of the rating system, the following devices have been evaluated: Amazon Echo 2 (1), Apple iPhone 5 (2), Floureon M32B (3), Google Home mini (4), Renkforce RenkCast (5), Sonoff Wi-Fi Smart Switch (6), Wi-Fi Smart Bulb (7), Wi-Fi Smart Plug (8) and Wi-Fi Touch Switch (9). Exemplary (not final) results can be found in Table VII (a dash indicates the parameter was not determined on that device). Afterwards, the devices were manually tested regarding their security. The evaluation has been compared with the previous determined scores. The conclusion is an overall success: the scoring fits the manual evaluation most of the time. This proves that the scoring system fulfills its purpose and can be used as a time saving way to rate the security of IoT devices. The process of scoring can be automated once the information is collected, which helps speeding up the security rating of an IoT network.

TABLE VII. EXAMPLE RESULTS

| parameter | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|---|---|---|
| Wi-Fi encryption | 2.00 | 2.00 | 2.00 | 2.00 | 2.00 | 2.00 | 2.00 | 2.00 | 2.00 |
| services | 1.00 | 2.00 | 0.33 | 1.00 | 2.00 | 2.00 | 1.00 | 2.00 | 2.00 |
| LAN communication | 2.00 | 2.00 | 2.00 | 2.00 | 2.00 | 2.00 | 2.00 | 2.00 | 2.00 |
| WAN communication | 2.00 | 2.00 | 2.00 | 2.00 | 2.00 | 2.00 | 2.00 | 2.00 | 2.00 |
| wired connection | 1.00 | 1.00 | 2.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 |
| cloud only | 1.00 | - | 2.00 | 1.00 | 2.00 | - | - | - | - |
| default password | 2.00 | 2.00 | 1.00 | 2.00 | 2.00 | 2.00 | 2.00 | 2.00 | 2.00 |
| **overall score** | **1.57** | **1.83** | **1.62** | **1.57** | **1.86** | **1.83** | **1.67** | **1.83** | **1.83** |

With all the device scores, an overall network score can be achieved by taking the lowest single device score. The weakness of a network is always defined by its weakest device.

After we evaluated the scoring system, we tried to find a solution for an automated process to gather all the necessary

 ISBN: 978-1-61208-746-7 

information. As stated in Section III, a completely automated scan without any additional information is not reliable. Therefore, we introduce an Open Standard for IoT Device Identification and Recognition (IoTAG), which will allow an automated and secure way to identify and index all IoT devices in a certain network.

## VII. IoTAG

Every IoT device should provide detailed information about itself and the current running software and firmware version. This enables an easy overview of the network and the security level with the previously shown device rating. We suggest to use a Transport Layer Security (TLS) 1.3 request to get the device information from the device. The response should use the JavaScript Object Notation (JSON) as standardized in ECMA-404 [24] and RFC 8259 [25]. JSON is faster to progress and uses less storage than for instance XML [26]. This benefits low powered IoT devices with restricted hardware.

The following information should be provided by the device:

- device ID
- device name
- device type
- manufacturer
- connectivity (e.g., Ethernet, Wi-Fi, Bluetooth, ...)
- firmware version
- firmware update URL
- software version (client)
- software update URL (client)
- auto updates enabled
- services and associated ports
- supported encryption

The device ID should be unique for each device, to allow a recognition. The device name can be extended with a revision number to ensure an exact assignment through multiple device versions.

Some possible device types are:

- sensor
- control
- camera
- smart TV
- smart speaker
- entertainment
- gaming
- household
- lightning

The device types are not exhaustive and can be extended. The manufacturer should allow a clear assignment to the responsible company. With a list of all the connectivity, the security rating can be extended and new threads in transmission technologies can be reported in a timely manner.

The firmware and possible existing client software version is very important for the scoring and to keep the whole network up to date. Additional to the version, a Uniform Resource Locator (URL), should be given. This URL must provide the current version and a secondary link to the new software version. This enables a third device to check the software version. In addition, the current auto update setting should be provided. In case this function is disabled, a security warning can be displayed.

As described in Section V, services and associated ports are a big part of the scoring system. The information about all running services improves the score and enables the possibility to check the proper configuration of the device. The protocol version can be used to identify outdated versions.

To check if the device can be used in a secure network, information about the supported encryption is necessary. This can be used to detect old devices with insecure encryption algorithms or exclude devices with no encryption at all.

The following data provides an example for the Google Home mini:

```
{
  "ID": "af0eb0335f952132b4e65999a373ce20",
  "name": "Home Mini revX",
  "type": "smart speaker",
  "manufacturer": "Google LLC",
  "connectivity": {
    "Wi-Fi": {
        "802.11": {
          "b": true,
          "g": true,
          "n": true,
          "ac": true
        },
        "frequencies": {
              "2.4": true,
              "5": true
        }
    },
    "bluetooth": "4.1"
  },
  "firmwareVersion": "1.27.090",
  "firmwareURL": "https://support.google.com/
     googlehome/answer/7365257?hl=en",
  "softwareVersion": "",
  "softwareURL": "",
  "autoUpdatesEnabled": true,
  "services": [
    {
      "name": "http",
      "port": "8008",
      "protocol": "tcp",
      "protocolVersion": "",
      "softwareVersion": ""
    },
    {
      "name": "ajp13",
      "port": "8009",
      "protocol": "tcp",
      "protocolVersion": "",
      "softwareVersion": ""
    },
    {
      "name": "https-alt",
      "port": "8443",
      "protocol": "tcp",
      "protocolVersion": "",
```

```
      "softwareVersion": ""
    },
    {
      "name": "cslistener",
      "port": "9000",
      "protocol": "tcp",
      "protocolVersion": "",
      "softwareVersion": ""
    },
    {
      "name": "scp-config",
      "port": "10001",
      "protocol": "tcp",
      "protocolVersion": "",
      "softwareVersion": ""
    }
  ],
  "encryption" : {
      ...
  }
}
```

These information provides no authenticity. Every device can send false IoTAG data and an attacker can impersonate a harmless device. Because of this, it is strongly recommended to sign this information with a private key, which can be trusted and verified over a public key infrastructure.

If an attacker has access to the network and uses the provided information from IoTAG to scan for insecure or unpatched devices, it brings out the importance for software and firmware updates. If all the devices use IoTAG, a central gateway (e.g., the router) can periodical check all devices. In case of a new vulnerability or missing software updates, the gateway can send a security warning or temporary disable the communication with the insecure device.

## VIII. Conclusion and future work

The operation of a secure IoT network in the context of a smart home is currently not possible for non-technical users. One solution can be the reoccurring scoring of the network. First, the complete network is scanned and all devices are rated with different criteria. With this device scoring, an overall score for the network is calculated, which is easy to read by a non-technical user. These ratings can be used to improve the security by updating old firmware or software versions, as well as replacing old, insecure devices with new ones. By performing this scan and rating on a daily basis, a quick response to new threads is possible. In the future, we plan to improve this approach by scanning vulnerability databases. If a new vulnerability emerges for a device in the network, the user can be warned immediately.

For an accurate and detailed device identification and recognition, the new standard IoTAG must be implemented by every manufacturer. State of the art network scans can not provide enough information to rate the security of a device. For example, with nmap it is possible to guess the running services but not their software version.

We are currently working on a test environment and application to demonstrate the benefits of IoTAG. However, for this tool to be widely used, we need the feedback and cooperation of IoT manufacturers. Also, we are planning to improve the network scoring system by testing it on further networks.

## References


[1] D. Goodin, Rash of in-the-wild attacks permanently destroys poorly secured IoT devices, Ars Technica, 2017. [Online]. Available from: https://arstechnica.com/information-technology/2017/04/rash-of-in-the-wild-attacks-permanently-detroys-poorly-secured-iot-devices/ [retrieved: 05, 2019].

[2] J. Wallen, Five nightmarish attacks that show the risks of IoT security, ZDNet, 2017. [Online]. Available from: https://www.zdnet.com/article/5-nightmarish-attacks-that-show-the-risks-of-iot-security/ [retrieved: 09, 2019].

[3] G. Lyon, Nmap: the Network Mapper - Free Security Scanner. [Online]. Available from: https://nmap.org [retrieved: 10, 2019].

[4] Fing Limited, Fing - IoT device intelligence for the connected world. [Online]. Available from: https://www.fing.com [retrieved: 10, 2019].

[5] M. Miettinen et al., "IOT SENTINEL Demo: Automated Device-Type Identification for Security Enforcement in IoT", IEEE 37th International Conference on Distributed Computing Systems (ICDCS), pp. 2511-2514, 2017.

[6] T. D. Nguyen et al., "DOT: A Federated Self-learning Anomaly Detection System for IoT", CoRR, pp. 756-767, 2019.

[7] T. Kohno, A. Broido, and K. C. Claffy, "Remote physical device fingerprinting", IEEE Trans. Dependable Secure Comput., vol. 2, no. 2, pp. 93108, April 2005.

[8] J. Cache, Fingerprinting 802.11 implementations via statistical analysis of the duration field, Uninformed, org 5, 2006.

[9] J. Franklin, D. McCoy, P. Tabriz, V. Neagoe, J. Van Randwyk, and D. Sicker, "Passive data link layer 802.11 wireless device driver fingerprinting", in USENIX Security Symposium, USENIX, pp. 167-178, 2006.

[10] V. Brik, S. Banerjee, M. Gruteser, and S. Oh, "Wireless device identification with radiometric signatures", International Conference on Mobile Computing and Networking, ACM, pp. 116127, 2008.

[11] F. Loiy, A. Sivanathany, H. H. Gharakheiliy, A. Radford, and V. Sivaraman, "Systematically Evaluating Security and Privacy for Consumer IoT Devices", IoT S&P 2017, pp. 1-6, 2017.

[12] K. C. Park and D. Shin, "Security assessment framework for IoT service", Telecommun Syst, pp. 193209, 2017.

[13] B. Ali and A. I. Awad, "Cyber and Physical Security Vulnerability Assessment for IoT-Based Smart Homes", sensors journal, vol 18(3), pp. 817, 2018.

[14] R. I. Bonilla, J. Crow, L. Basantes, and L. Cruz, "A Metric for Measuring IoT Devices Security Levels", IEEE 15th Intl Conf on Dependable, Autonomic and Secure Computing, pp. 704-709, 2017.

[15] A. E. Khaled, H. Abdelsalam, L. Wyatt, and L. Choonhwa, "IoT-DDL device description language for the T in IoT", IEEE Access 6, pp. 24048-24063, 2018.

[16] S. Kaebisch and A. Darko, "Thing description as enabler of semantic interoperability on the Web of Things", IoT Semantic Interoperability Workshop, pp. 1-3, 2016.

[17] L. Hinterberger, "Automated Risk Analysis of IoT-Infrastructures", Applied Research Conference, pp. 586-588, 2019.

[18] J. Schmidt, Cryptography in IT - recommendations on encryption and procedures, Kryptographie in der IT - Empfehlungen zu Verschluesselung und Verfahren, 2017. [Online]. Available from: https://www.heise.de/security/artikel/Kryptographie-in-der-IT-Empfehlungen-zu-Verschluesselung-und-Verfahren-3221002.html [retrieved: 09, 2019].

[19] G. Lyon, Service and version detection, Dienst- und Versionserkennung. [Online]. Available from: https://nmap.org/man/de/man-version-detection.html [retrieved: 09, 2019].

[20] Wireshark Foundation, tshark - Dump and analyze network traffic. [Online]. Available from: https://www.wireshark.org/docs/man-pages/tshark.html [retrieved: 10, 2019].

[21] The Hacker's Choice, thc-hydra. [Online]. Available from: https://github.com/vanhauser-thc/thc-hydra [retrieved: 10, 2019].

[22] AVM GmbH, Login to the FRITZ!Box Web Interface, 2018. [Online]. Available from: https://avm.de/fileadmin/user_upload/Global/Service/Schnittstellen/ Session-ID_english_13Nov18.pdf [retrieved: 07, 2019].

[23] G. Lyon, OS Detection - Nmap Network Scanning. [Online]. Available from: https://nmap.org/book/man-os-detection.html [retrieved: 09, 2019].


 ISBN: 978-1-61208-746-7 

[24] Ecma International, ECMA-404: The JSON Data Interchange Syntax, 2017.
[25] Internet Engineering Task Force (IETF), The JavaScript Object Notation (JSON) Data Interchange Format, https://tools.ietf.org/html/rfc8259, 2017.
[26] N. Nurseitov, M. Paulson, R. Reynolds, and C. Izurieta, Comparison of JSON and XML Data Interchange Formats: A Case Study, CAINE, 2009.

 ISBN: 978-1-61208-746-7 