IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received September 16, 2019, accepted October 3, 2019, date of publication October 7, 2019, date of current version October 17, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2946081

Threat Analysis for Wearable Health Devices and Environment Monitoring Internet of Things Integration System

TZU WEI TSENG, CHIA TUNG WU, AND FEIPEI LAI, (Senior Member, IEEE)
Department of Computer Science and Information Engineering, National Taiwan University, Taipei 10617, Taiwan

Corresponding author: Tzu Wei Tseng (chaaa463@gmail.com)

This work was supported by the Ministry of Science and Technology Taiwan under Grant MOST 107-2634-F-002 -015.

ABSTRACT With the rapid development of Internet of Things (IoT) applications, heterogeneous device management issues tend to arise in architecture security due to hardware computing power, types of software, data transmission interfaces, and networking protocols. Even during data exchange between devices and systems, traditional IoT devices are prone to the disclosure of personal information, which compromises privacy. Thus, planning an effective information security management strategy has become an essential part of application development. This paper presents a strategy to achieve information security verification and risk assessment for an IoT-based personal health information system. Using several interfaces of IoT devices, including Wi-Fi and Bluetooth, we simulate possible attack hypotheses and define test methods and evaluation methods suitable for each device. In our application systems for information security analysis, we also consider and integrate weaknesses of the system architecture to achieve a more complete information security threat analysis.

INDEX TERMS Internet of Things (IoT), security, risk analysis, privacy.

I. INTRODUCTION

The Internet of Things (IoT) is the concept of connecting devices to the Internet and each other to provide services. In recent years, the Internet of things has been regarded as a new technology with a significant influence on big data analysis and artificial intelligence [1], [2]. In medical services [3], [4], the IoT can assist the development of personal health care. For example, personal health care uses the application model of wearable devices to help process biological characteristics and ensure early detection and monitoring of some diseases [5]. Personal privacy information is a considerable challenge for the IoT [6]–[8]. Due to environment and hardware, most IoT devices have resource limitations. With insufficient computing power, information security protection mechanisms for data and transmission processes are abandoned to achieve efficiency. On the issue of device heterogeneity, it is difficult to achieve hardware efficiency in managing devices and integrating systems. There are many security vulnerabilities [9]–[13] and cyber threats, such as the infamous Mirai malware [14]. After infection by Mirai malware, the device continues to scan the IoT for Internet devices' IP addresses and uses the default username and password to login to these devices. While the infected device will continue to work, it becomes a member of the Mirai botnet.

II. RELATED WORK

In the literature on wearable devices of the IoT, many studies have focused on the issue of personal information disclosure [15], [16]. Testing methods used include port scanning, security vulnerability scanning, and vulnerability exploitation with well-known tools such as Nmap [17], Nessus [18], OpenVAS [19], Wireshark [20] and Burp Suite [21]. Based on differences in networking modes and capabilities, IoT devices, may be classified into two transmission modes—Wi-Fi and Bluetooth—which are implemented using different protocols. In the relevant researches on Bluetooth security, the attack dimension of each study can be classified as either a direct attack of hardware RAM [22], packet collection of network transmission packets [23], firmware cracking [24], or an App cracking attack of a Bluetooth device [25].

After obtaining the results of an information security vulnerabilities test, we enter the risk assessment stage.

The associate editor coordinating the review of this manuscript and approving it for publication was Young Jin Chun.

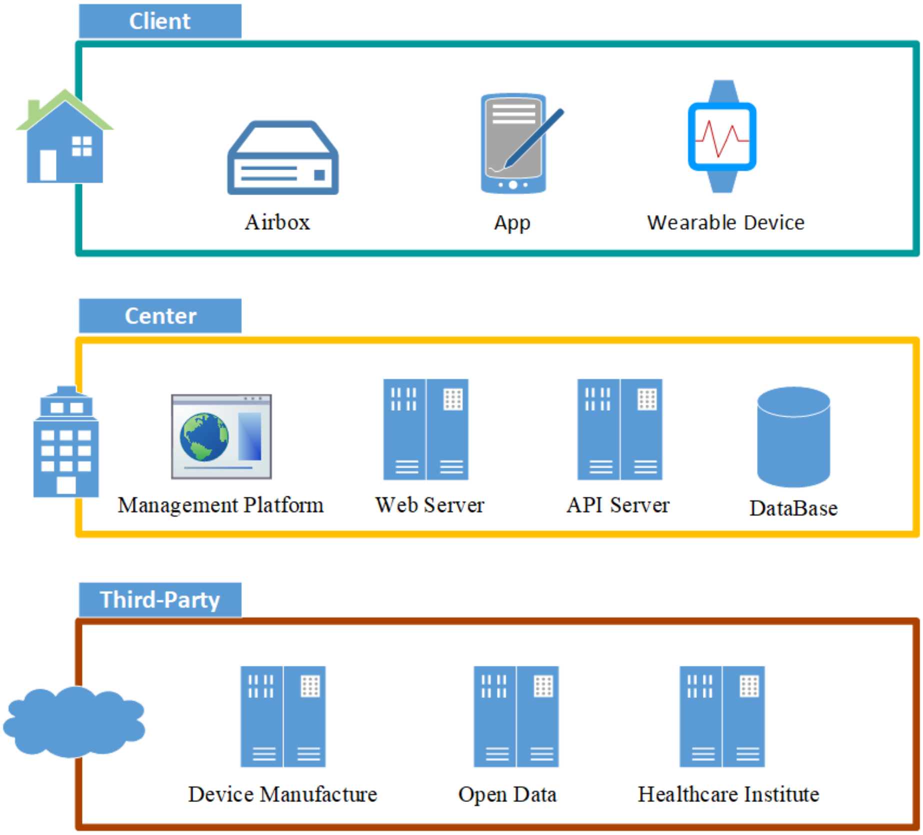


FIGURE 1. Integration framework for internet of things devices and third-party systems.

According to IoT risk assessment research [6], [26], the method of risk assessment can be classified as either quantitative evaluation or qualitative evaluation. Quantitative evaluation uses historical statistical data for analysis. Qualitative evaluation uses analysis to identify the probability of risk occurrence and analyze the impact of risk occurrence on the target, as well as other factors. Here, we can adopt methods such as the STRIDE model [27] and the DREAD model [28], [29]. In the STRIDE model-based risk assessment studies, either the DREAD model is used to calculate the risk ratio, a Data Flow Diagram (DFD) [30] is used to analyze more subtle risks, a custom risk assessment model is used [31]–[35], or risk analysis and protective measures are developed in the form of clauses.

III. USE CASE: PERSONAL HEALTH DATA COLLECTION SYSTEM

The platform aims to assess the user's potential environmental risk factors and offer healthy lifestyle recommendations. Medical service providers can also use the case management provided by the platform to accurately grasp the case situation and treatment. To achieve the goal of personalized medicine, the framework uses IoT devices, such as mobile phones, smartwatches, and environmental monitoring devices, to monitor the lifestyles and home environments of patients outside of hospitals. It can collect data on lifestyle, household environment, and public environmental information from publicly-available government data.

A. THE IoT SYSTEM INTEGRATION FRAMEWORK

The healthy lifestyle application, as shown in Figure 1, can capture the user's current mobile phone location (using background procedures and sensor information) and immediately send this location to the server. When the user returns home, he or she can also examine the average air quality the user has been exposed to during the day.

Regarding data collection, the main functions of the healthy life app are: (1) to examine the user's home and local environmental data such as PM2.5, PM10, CO_2, CO, temperature, and humidity; (2) to collect lifestyle information, such as sleep quality, activity, heart rate, blood pressure, and vital signs; and (3) to collect publicly-available government environmental information, such as ultraviolet, air quality, and weather conditions. The platform is used by case managers and doctors. In the case of care, specific wearable devices are needed, and airboxes are installed in the home. The platform provides immediate access to external environmental data, internal environmental data (such as temperature, humidity, PM2.5, and PM10) using IoT sensors, and personal health information (activities, calories burned, heart rate, blood pressure, and blood oxygen) through a public third-party API. The data transmission process is transmitted by the HTTPS protocol to ensure the confidentiality and integrity of the data. All data are synchronized to the database every five minutes. The platform implements access control for patient privacy.

B. WORKFLOW

We use five workflows to illustrate examples of use cases: user registration, data query, management platform, sensor data collection, and open data collection.

1) USER REGISTRATION

The user registration workflow consists of four steps: (a) the user uses a web browser to access third-party websites in a medical application; (b) after registering with the third-party server, the user logs in with a username and password, which calls back the authentication token to the central API server using OAuth2.0; (c) when the central API server receives a response authentication token, the token is sent to the database to be saved; and (d) the medical application is updated during the user's launch of the application service on the phone.

2) DATA QUERY

The data query workflow consists of four steps: (a) the medical app requests an authentication token and metadata from the central API server; (b) if the central API server does not have the user's data, it will reply to the third-party website that the user needs to log in; (c) if the central API server has the user data, it will respond to the authentication token; and (d) medical applications use authentication tokens to obtain personal health data from the third-party API.

3) MANAGEMENT PLATFORM

The management platform workflow consists of four steps: (a) the user registers an account on the management platform; (b) when logging in to the portal, the user is authenticated according to the permissions of his or her role; (c) the user is required to enter the patient's identity information, and the request is sent to the web server; and (d) the web server retrieves relevant metadata and sends the data to the central API server, requesting that personal health data be sent to the database.

4) SENSOR DATA COLLECTION

The sensor data collection workflow consists of three steps: (a) the sensor acquires data and sends it to the app via Bluetooth; (b) after the third-party obtains the new data, a notification is sent to the central API; and (c) the central API server requests new data using the user's authentication token, and then sends the data to the database.

5) OPEN DATA COLLECTION

The open data collection workflow consists of one step: (a) the scheduling service requests the open data API and retrieves the data to be stored in the database.

IV. THREAT MODELING

To create a threat model for this application, we define a security motivation that includes security basics such as confidentiality, integrity, and availability. We then create an application overview to understand the functionality of the application, which involves (1) identifying data for application sources and targets, (2) understanding the purpose of the application, and (3) analyzing the function of the application.

The application can be divided into levels that define external entities, trust boundaries, data flows, and entry points, with areas of vulnerability between them. It can also help with attack surface analysis. The following activities help decompose the application: (1) enumerate external dependencies, (2) list the entry points, (3) determine the components, (4) specify the trust level, and (5) draw a data flow chart. External dependencies define the application's dependencies on external entities such as servers, firewalls, security policies, operating systems, networks, and so on. Although these entities are beyond the control of the application, they are still within the control of the organization. Identifying these external dependencies may help minimize the overall risk to the application. An entry point is an interface where a user enters data or interacts with an application. Potential attackers can use these entry points to attack applications. Applications can have multiple entry points; for example, external entries are exposed to internal entry points that are exposed to subcomponents across the application layer. Attackers can bypass the first level of the entry point to directly attack the internal entry point. Typical entry points include login and authentication, management interfaces, query and search capabilities, transaction interfaces (such as APIs), and business workflows. An application's assets are of value to the attacker. They are important targets of a threat and one reason why applications are at risk of attack. It is critical to identify all assets in the application that need to be protected against unauthorized access.

The trust level defines the access that an application should grant to external entities. We define possible roles, including a set of privileges and trust levels assigned to roles.

Figure 2 illustrates how we analyze system architecture security from an information security perspective to achieve superior risk identification, risk analysis, risk assessment, and improvement measures. After security requirements are identified, the application's security design becomes simpler. Security design principles are practices or guidelines that developers follow during the development phase. Threat modeling is the process of identifying, analyzing, and mitigating application threats. It is a structured approach that allows developers to assess threats based on the application's architecture and implementation, executed during the design phase of the security development lifecycle. Threat modeling is an iterative process, starting with the design phase of the application and iterating through the application lifecycle until all possible threats to the application are identified. The output of threat modeling is a threat that exposes all possible

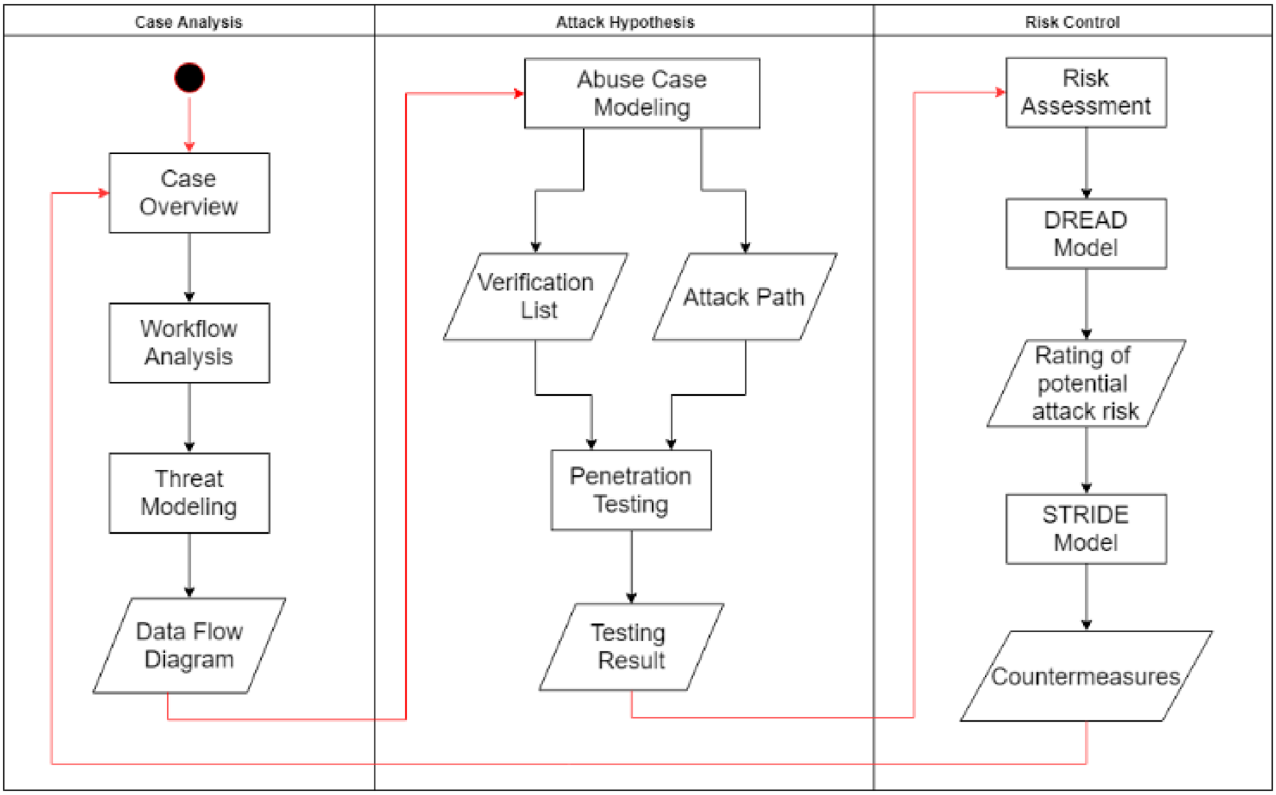


FIGURE 2. Threat modeling and risk assessment process.

vulnerabilities in the application. Security solutions include input validation, database layer abstraction, server configuration, proxies, web application firewalls, data encryption, and operating system hardening. The steps are as follows: (1) attack surface evaluation, where the application is decomposed and its entry points are reviewed from an attacker's perspective; (2) threat identification, where each entry point is reviewed against potential threats; (3) impact analysis, where the impact of the potential threat is calculated in terms of risk; and (4) control recommendations, where security controls are recommended to meet security objectives.

A. DATA FLOW DIAGRAM

The information collected, such as external entities, entry points, assets, and trust levels, improves the accuracy of the application model when using a data flow diagram (DFD). The DFD helps to explain how data flows through the application and what happens to the data as it flows. Starting with a higher level DFD, then continuing to create a lower-level DFD, we decompose the application into different processes and their lower-level subprocesses or functions. The higher-level DFD helps with analysis of the overall scope of the application, while the lower-level DFD illustrates the operation of specific lower-level processes.

Once a DFD is created at the lowest level, we identify all external dependencies, entry points, and trust levels in the application. This aids in quickly determining what data need to be provided to a particular process in the DFD and what the corresponding attacker is targeting to exploit the capabilities of that specific process. This target is used in the DFD to determine the threat path in the application. We identify all possible attackers' targets and vulnerabilities that attackers use to achieve their goals. In this study, the architecture overview of the use case is established using the Microsoft Threat Modeling tool [36] for DFD analysis of the architecture, as shown in Figure 3.

B. ABUSE CASE MODELING

For our abuse cases, we divided the test into two parts: the security weakness of the target and the possibility of data disclosure. In this study, we use a penetration test. However, a third-party API test can seriously affect legal security issues; therefore, we will only carry out security verification on the IoT devices and the related applications of the self-developed platform used by users, as shown in Table 1. If the device transmits with Wi-Fi and the TCP/IP Protocol is adopted, we use Nmap for active scanning detection. During device operation, Wireshark is used to collect and analyze the transmitted data packets of the device on the network. If the device transmits with Bluetooth, we use either Host Controller Interface (HCI) packets through an Android or a Bluetooth sniffer to collect broadcast packets. The test methods are described in this section.

1) AIRBOX

We ensure that there is no risk of information disclosure or vulnerability to the device in the usage scenarios

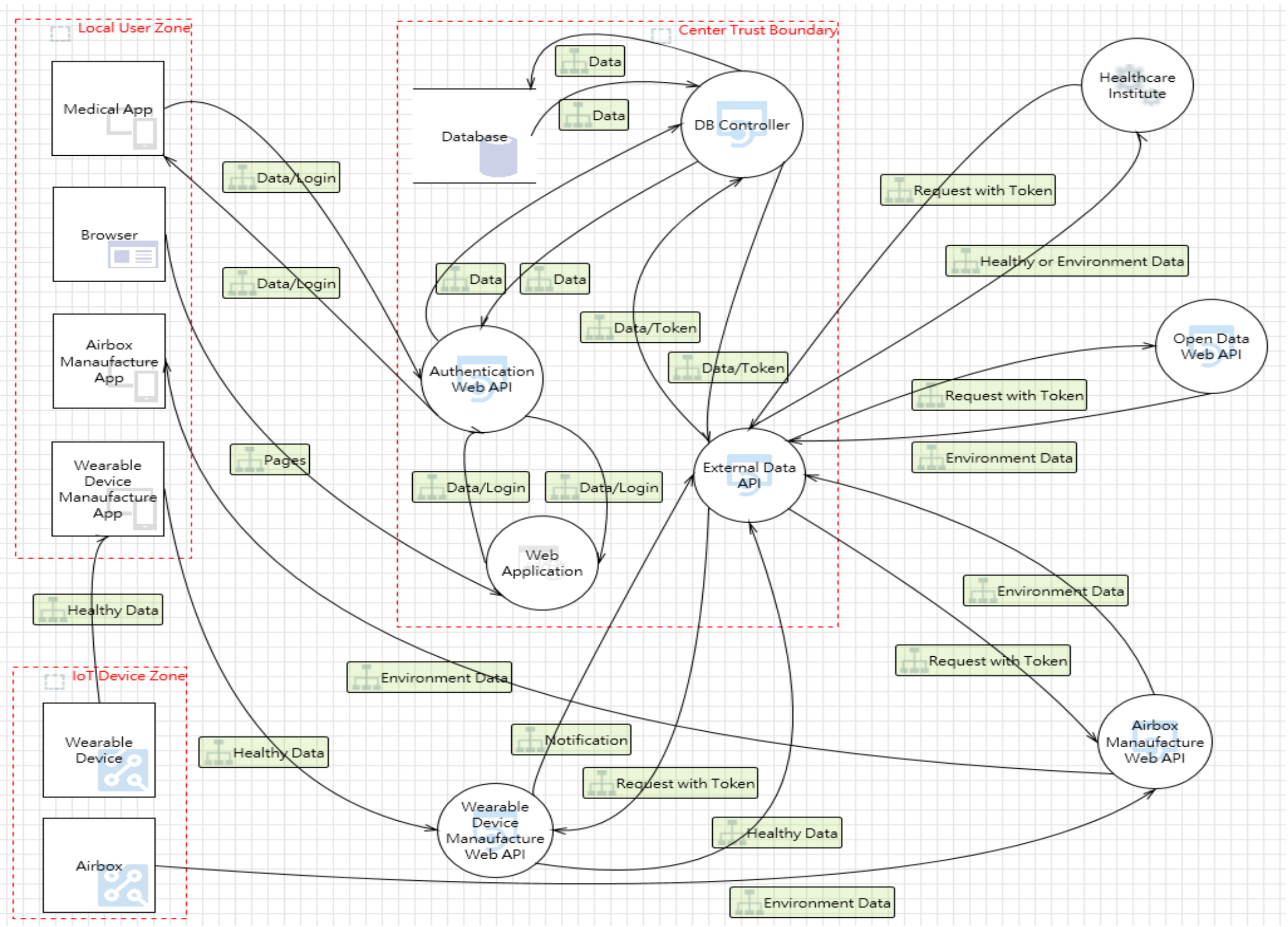


FIGURE 3. Data flow diagram of personal health data collection system.

TABLE 1. Security verification list.

Boundary	Target	Type	Connection	TCP/IP Port	Data Type	Gateway	Data Destination	Testing Tool	Expected Risk
Third party	Edimax Airbox	IoT	Wi-Fi	5678	Environment	X	Manufacture API	Wireshark Nmap RouterSploit	Weak Service Information leakage
Third party	Garmin Vivosmart3	IoT	Bluetooth	X	Healthy	App	Manufacture API	Ubertooth One HCI Log	Weak Service Information leakage
Data Center	Medical App	App	Wi-Fi	443	Healthy Environment	X	Data Center	Burp Suite	MITM
Data Center	Management Platform	Web App	Ethernet	443	Healthy Environment	X	Data Center	Sqlmap OWASP ZAP	Injection Attack

employed in this study, as shown in Figure 4. Based on the abuse model, we use Wireshark, Nmap, and RouterSploit to examine the device.

Wireshark is a tool that can be used to capture data in the network. It can display and interpret captured data in a single-packet format (frame, datagram, packet, and segment) under many network protocol scenarios. Figure 5 illustrates that the tool is typically used to (a) find the root cause of a known problem, (b) search between devices for a particular protocol or data stream, and (c) analyze a particular sequence or protocol flag for each packet.

The main function of Nmap is to scan TCP/IP ports, to discover both to the open port of the target host and the corresponding network service type, application software

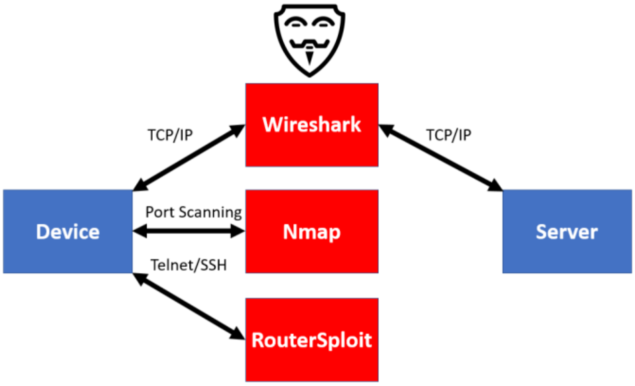
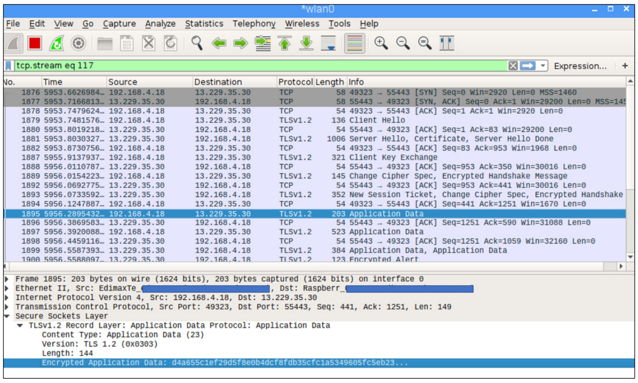


FIGURE 4. The attack path of the airbox.

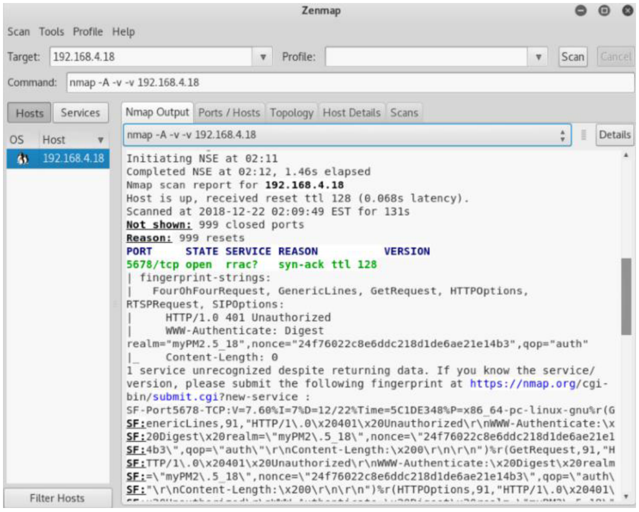

*wlan0
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Expression...
+

```
tcp.stream eq 117
```

```
No.  Time              Source         Destination    Protocol Length Info
1876 5953.6626984…  192.168.4.18   13.229.35.30   TCP        58 49323 → 55443 [SYN] Seq=0 Win=2920 Len=0 MSS=1460
1877 5953.7166813…  13.229.35.30   192.168.4.18   TCP        58 55443 → 49323 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0 MSS=145
1878 5953.7479624…  192.168.4.18   13.229.35.30   TCP        54 49323 → 55443 [ACK] Seq=1 Ack=1 Win=2920 Len=0
1879 5953.7481576…  192.168.4.18   13.229.35.30   TLSv1.2   136 Client Hello
1880 5953.8019218…  13.229.35.30   192.168.4.18   TCP        54 55443 → 49323 [ACK] Seq=1 Ack=83 Win=29200 Len=0
1881 5953.8030327…  13.229.35.30   192.168.4.18   TLSv1.2  1006 Server Hello, Certificate, Server Hello Done
1882 5953.8730756…  192.168.4.18   13.229.35.30   TCP        54 49323 → 55443 [ACK] Seq=83 Ack=953 Win=1968 Len=0
1887 5955.9137937…  192.168.4.18   13.229.35.30   TLSv1.2   321 Client Key Exchange
1888 5956.0110787…  13.229.35.30   192.168.4.18   TCP        54 55443 → 49323 [ACK] Seq=953 Ack=350 Win=30016 Len=0
1889 5956.0154223…  192.168.4.18   13.229.35.30   TLSv1.2   145 Change Cipher Spec, Encrypted Handshake Message
1892 5956.0692775…  13.229.35.30   192.168.4.18   TCP        54 55443 → 49323 [ACK] Seq=953 Ack=441 Win=30016 Len=0
1893 5956.0733592…  13.229.35.30   192.168.4.18   TLSv1.2   352 New Session Ticket, Change Cipher Spec, Encrypted Handshake
1894 5956.1247887…  192.168.4.18   13.229.35.30   TCP        54 49323 → 55443 [ACK] Seq=441 Ack=1251 Win=1670 Len=0
1895 5956.2895432…  192.168.4.18   13.229.35.30   TLSv1.2   203 Application Data
1896 5956.3869583…  13.229.35.30   192.168.4.18   TCP        54 55443 → 49323 [ACK] Seq=1251 Ack=590 Win=31088 Len=0
1897 5956.3920088…  192.168.4.18   13.229.35.30   TLSv1.2   523 Application Data
1898 5956.4459116…  13.229.35.30   192.168.4.18   TCP        54 55443 → 49323 [ACK] Seq=1251 Ack=1059 Win=32160 Len=0
1899 5956.5587393…  13.229.35.30   192.168.4.18   TLSv1.2   384 Application Data, Application Data
1900 5956.5588097…  13.229.35.30   192.168.4.18   TLSv1.2   123 Encrypted Alert
```

```
▸ Frame 1895: 203 bytes on wire (1624 bits), 203 bytes captured (1624 bits) on interface 0
▸ Ethernet II, Src: EdimaxTe_[redacted], Dst: Raspberr_[redacted]
▸ Internet Protocol Version 4, Src: 192.168.4.18, Dst: 13.229.35.30
▸ Transmission Control Protocol, Src Port: 49323, Dst Port: 55443, Seq: 441, Ack: 1251, Len: 149
▾ Secure Sockets Layer
  ▾ TLSv1.2 Record Layer: Application Data Protocol: Application Data
      Content Type: Application Data (23)
      Version: TLS 1.2 (0x0303)
      Length: 144
      Encrypted Application Data: d4a655c1ef29d5f8e0b4dcf8fdb35cfc1a5349605fc5eb23...
```

FIGURE 5. Use wireshark to capture packet data during network transfer.



Target: 192.168.4.18

Profile:

Command: nmap -A -v -v 192.168.4.18

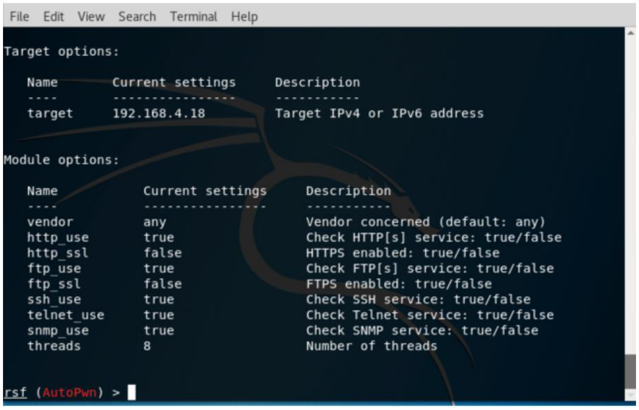
Host: 192.168.4.18

```
nmap -A -v -v 192.168.4.18
```

```
Initiating NSE at 02:11
Completed NSE at 02:12, 1.46s elapsed
Nmap scan report for 192.168.4.18
Host is up, received reset ttl 128 (0.068s latency).
Scanned at 2018-12-22 02:09:49 EST for 131s
Not shown: 999 closed ports
Reason: 999 resets
PORT     STATE SERVICE REASON         VERSION
5678/tcp open  rrac?   syn-ack ttl 128
| fingerprint-strings:
|   FourOhFourRequest, GenericLines, GetRequest, HTTPOptions,
RTSPRequest, SIPOptions:
|     HTTP/1.0 401 Unauthorized
|     WWW-Authenticate: Digest
realm="myPM2.5_18",nonce="24f76022c8e6ddc218d1de6ae21e14b3",qop="auth"
|_    Content-Length: 0
1 service unrecognized despite returning data. If you know the service/
version, please submit the following fingerprint at https://nmap.org/cgi-
bin/submit.cgi?new-service :
SF-Port5678-TCP:V=7.60%I=7%D=12/22%Time=5C1DE348%P=x86_64-pc-linux-gnu%r(G
SF:enericLines,91,"HTTP/1\.0\x20401\x20Unauthorized\r\nWWW-Authenticate:\x
SF:20Digest\x20realm=\"myPM2\.5_18\",nonce=\"24f76022c8e6ddc218d1de6ae21e1
SF:4b3\",qop=\"auth\"\r\nContent-Length:\x200\r\n\r\n")%r(GetRequest,91,"H
SF:TTP/1\.0\x20401\x20Unauthorized\r\nWWW-Authenticate:\x20Digest\x20realm
SF:=\"myPM2\.5_18\",nonce=\"24f76022c8e6ddc218d1de6ae21e14b3\",qop=\"auth\
SF:"\r\nContent-Length:\x200\r\n\r\n")%r(HTTPOptions,91,"HTTP/1\.0\x20401\
```

FIGURE 6. Scan the target for service weaknesses and compare device fingerprints.

name, and version. Figure 6 illustrates that Nmap can detect system packet information for the target host. Nmap provides a variety of scanning methods that meet the needs of TCP/IP detection for the deep-penetration test. Zenmap is a GUI version of Nmap that uses Nmap's core when performing scans but has a more user-friendly graphical interface.



```
Target options:

   Name       Current settings     Description
   ----       ----------------     -----------
   target     192.168.4.18         Target IPv4 or IPv6 address

Module options:

   Name          Current settings     Description
   ----          ----------------     -----------
   vendor        any                  Vendor concerned (default: any)
   http_use      true                 Check HTTP[s] service: true/false
   http_ssl      false                HTTPS enabled: true/false
   ftp_use       true                 Check FTP[s] service: true/false
   ftp_ssl       false                FTPS enabled: true/false
   ssh_use       true                 Check SSH service: true/false
   telnet_use    true                 Check Telnet service: true/false
   snmp_use      true                 Check SNMP service: true/false
   threads       8                    Number of threads

rsf (AutoPwn) > 
```

FIGURE 7. Quick automated detection of SSH, Telnet.

RouterSploit is an open-source development framework that facilitates the testing of vulnerabilities in embedded devices. As shown in Figure 7, it provides several modules that help with the penetration testing process: (a) Vulnerability, to exploit known vulnerabilities in the past, (b) Creds, which assists with network service credentials, (c) Scanner, to scan the target for possible attacks, (d) Payload, to generate payloads for injection points, and (e) Generics, to perform generic attacks.

After testing these tools, we found that the IoT device did not disclose any confidential information. In the network transport layer, it has an encrypted transport protocol using HTTPS, so there is no need to worry about eavesdropping during the transmission. During a man-in-the-middle (MITM) attack, eavesdropping will not happen because it is impossible to install forged credentials on the device. Although the device has an open listening service section, it does not open unnecessary ports for Telnet or SSH. Therefore, the device does not suffer from the possibility of brute force cracking by remote login.

2) WEARABLE DEVICE

Many of the IoT devices make extensive use of Bluetooth for communication to achieve not only low power but also low latency and cost. These scenarios include IoT applications for medical data, smart homes, personal health monitoring devices, and others. In the Bluetooth environment, not only can a device support multiple transmission modes, but the current development of the Bluetooth standard is open and license-free, making it easier to adopt existing Bluetooth technologies. This phase of the test method, shown in Figure 8, is primarily aimed at wearable devices transmitted by Bluetooth. Ubertooth One [37] is used to scan the target device and collect Bluetooth protocol packets to enable deep analysis and discussion of protocol weaknesses experienced in the past.

Ubertooth One is a 2.4GHz transceiver built for Bluetooth monitoring and traffic injection. Ubertooth One was developed as an open-source project to provide Bluetooth

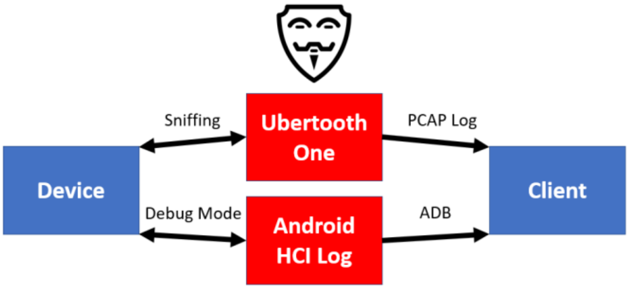
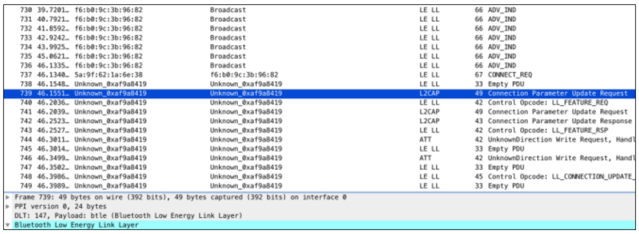


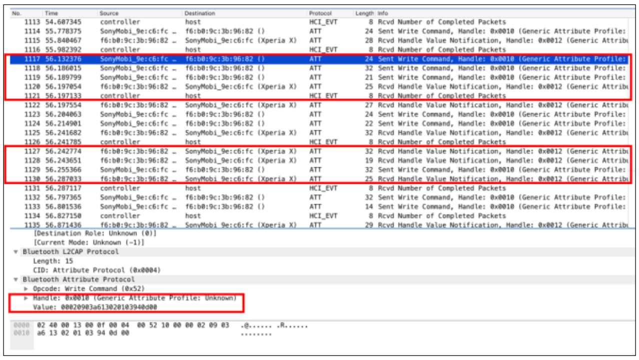
FIGURE 8. The attack path of the wearable device.



```
730 39.7201… f6:b0:9c:3b:96:82        Broadcast                  LE LL   66 ADV_IND
731 40.7921… f6:b0:9c:3b:96:82        Broadcast                  LE LL   66 ADV_IND
732 41.8592… f6:b0:9c:3b:96:82        Broadcast                  LE LL   66 ADV_IND
733 42.9242… f6:b0:9c:3b:96:82        Broadcast                  LE LL   66 ADV_IND
734 43.9925… f6:b0:9c:3b:96:82        Broadcast                  LE LL   66 ADV_IND
735 45.0621… f6:b0:9c:3b:96:82        Broadcast                  LE LL   66 ADV_IND
736 46.1335… f6:b0:9c:3b:96:82        Broadcast                  LE LL   66 ADV_IND
737 46.1340… 5a:9f:62:1a:6e:38        f6:b0:9c:3b:96:82          LE LL   67 CONNECT_REQ
738 46.1548… Unknown_0xaf9a8419       Unknown_0xaf9a8419         LE LL   33 Empty PDU
739 46.1551… Unknown_0xaf9a8419       Unknown_0xaf9a8419         L2CAP   49 Connection Parameter Update Request
740 46.2036… Unknown_0xaf9a8419       Unknown_0xaf9a8419         LE LL   42 Control Opcode: LL_FEATURE_REQ
741 46.2039… Unknown_0xaf9a8419       Unknown_0xaf9a8419         L2CAP   49 Connection Parameter Update Request
742 46.2523… Unknown_0xaf9a8419       Unknown_0xaf9a8419         L2CAP   43 Connection Parameter Update Response
743 46.2527… Unknown_0xaf9a8419       Unknown_0xaf9a8419         LE LL   42 Control Opcode: LL_FEATURE_RSP
744 46.3011… Unknown_0xaf9a8419       Unknown_0xaf9a8419         ATT     42 UnknownDirection Write Request, Handl
745 46.3014… Unknown_0xaf9a8419       Unknown_0xaf9a8419         LE LL   33 Empty PDU
746 46.3499… Unknown_0xaf9a8419       Unknown_0xaf9a8419         ATT     42 UnknownDirection Write Request, Handl
747 46.3502… Unknown_0xaf9a8419       Unknown_0xaf9a8419         LE LL   33 Empty PDU
748 46.3986… Unknown_0xaf9a8419       Unknown_0xaf9a8419         LE LL   45 Control Opcode: LL_CONNECTION_UPDATE_
749 46.3989… Unknown_0xaf9a8419       Unknown_0xaf9a8419         LE LL   33 Empty PDU
▶ Frame 739: 49 bytes on wire (392 bits), 49 bytes captured (392 bits) on interface 0
▶ PPI version 0, 24 bytes
  DLT: 147, Payload: btle (Bluetooth Low Energy Link Layer)
▼ Bluetooth Low Energy Link Layer
```

FIGURE 9. Sniff packet of logical link control and adaptation layer protocol.

security analysis for anyone. As shown in Figure 9, the project includes hardware specifications, software, and firmware that make Ubertooth One an effective Bluetooth sniffer, allowing it to sniff data over a base-rate Bluetooth communication connection automatically.



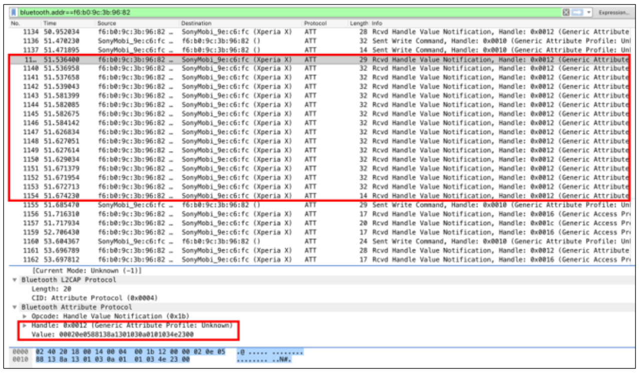
```
No.  Time       Source               Destination                  Protocol  Length Info
1113 54.607345  controller           host                         HCI_EVT    8 Rcvd Number of Completed Packets
1114 55.778375  SonyMobi_9e:c6:fc …  f6:b0:9c:3b:96:82 ()         ATT       24 Sent Write Command, Handle: 0x0010 (Generic Attribute Profile:
1115 55.840467  f6:b0:9c:3b:96:82 …  SonyMobi_9e:c6:fc (Xperia X) ATT       28 Rcvd Handle Value Notification, Handle: 0x0012 (Generic Attribu
1116 55.982392  controller           host                         HCI_EVT    8 Rcvd Number of Completed Packets
1117 56.132376  SonyMobi_9e:c6:fc …  f6:b0:9c:3b:96:82 ()         ATT       24 Sent Write Command, Handle: 0x0010 (Generic Attribute Profile:
1118 56.186015  SonyMobi_9e:c6:fc …  f6:b0:9c:3b:96:82 ()         ATT       32 Sent Write Command, Handle: 0x0010 (Generic Attribute Profile:
1119 56.189799  SonyMobi_9e:c6:fc …  f6:b0:9c:3b:96:82 ()         ATT       21 Sent Write Command, Handle: 0x0010 (Generic Attribute Profile:
1120 56.197054  f6:b0:9c:3b:96:82 …  SonyMobi_9e:c6:fc (Xperia X) ATT       25 Rcvd Handle Value Notification, Handle: 0x0012 (Generic Attribu
1121 56.197133  controller           host                         HCI_EVT    8 Rcvd Number of Completed Packets
1122 56.197554  f6:b0:9c:3b:96:82 …  SonyMobi_9e:c6:fc (Xperia X) ATT       27 Rcvd Handle Value Notification, Handle: 0x0012 (Generic Attribu
1123 56.204063  SonyMobi_9e:c6:fc …  f6:b0:9c:3b:96:82 ()         ATT       24 Sent Write Command, Handle: 0x0010 (Generic Attribute Profile:
1124 56.214901  SonyMobi_9e:c6:fc …  f6:b0:9c:3b:96:82 ()         ATT       22 Sent Write Command, Handle: 0x0010 (Generic Attribute Profile:
1125 56.241682  f6:b0:9c:3b:96:82 …  SonyMobi_9e:c6:fc (Xperia X) ATT       32 Rcvd Handle Value Notification, Handle: 0x0012 (Generic Attribu
1126 56.241785  controller           host                         HCI_EVT    8 Rcvd Number of Completed Packets
1127 56.242774  f6:b0:9c:3b:96:82 …  SonyMobi_9e:c6:fc (Xperia X) ATT       32 Rcvd Handle Value Notification, Handle: 0x0012 (Generic Attribu
1128 56.243651  f6:b0:9c:3b:96:82 …  SonyMobi_9e:c6:fc (Xperia X) ATT       19 Rcvd Handle Value Notification, Handle: 0x0012 (Generic Attribu
1129 56.255366  SonyMobi_9e:c6:fc …  f6:b0:9c:3b:96:82 ()         ATT       32 Sent Write Command, Handle: 0x0010 (Generic Attribute Profile:
1130 56.287033  f6:b0:9c:3b:96:82 …  SonyMobi_9e:c6:fc (Xperia X) ATT       25 Rcvd Handle Value Notification, Handle: 0x0012 (Generic Attribu
1131 56.287117  controller           host                         HCI_EVT    8 Rcvd Number of Completed Packets
1132 56.797365  SonyMobi_9e:c6:fc …  f6:b0:9c:3b:96:82 ()         ATT       32 Sent Write Command, Handle: 0x0010 (Generic Attribute Profile:
1133 56.801536  SonyMobi_9e:c6:fc …  f6:b0:9c:3b:96:82 ()         ATT       14 Sent Write Command, Handle: 0x0010 (Generic Attribute Profile:
1134 56.827150  controller           host                         HCI_EVT    8 Rcvd Number of Completed Packets
1135 56.871436  f6:b0:9c:3b:96:82 …  SonyMobi_9e:c6:fc (Xperia X) ATT       29 Rcvd Handle Value Notification, Handle: 0x0012 (Generic Attribu
    [Destination Role: Unknown (0)]
    [Current Mode: Unknown (-1)]
▼ Bluetooth L2CAP Protocol
    Length: 15
    CID: Attribute Protocol (0x0004)
▼ Bluetooth Attribute Protocol
  ▶ Opcode: Write Command (0x52)
  ▶ Handle: 0x0010 (Generic Attribute Profile: Unknown)
    Value: 00020903a613020103940d00

0000  02 40 00 13 00 0f 00 04  00 52 10 00 00 02 09 03   .@...... .R......
0010  a6 13 02 01 03 94 0d 00                            ........
```

FIGURE 10. The app sends vibration commands to the wearable device via bluetooth.

With Android 4.4 or later, we can actively collect Bluetooth HCI logs. These logs capture the part of the HCI package that contains Bluetooth communication. For most Android devices, the log is stored in a file named ''btsnoop_hci.log.'' To record all data, the user needs to enable Bluetooth HCI logging through the Developer option, which allows Bluetooth HCI snoop to capture all HCI packages in the file during communication between the IoT device and the Android device. One example is a wearable device that synchronizes data and triggers vibrations, as shown in Figures 10 and 11.

Although the process of collecting packets by Ubertooth One and HCI Log did not reveal any privacy breaches, this test demonstrated a capability for sensitive data and communication security detection when applied to commercial products. The use of Bluetooth's low-power wireless transmission technology in the medical environment will be a significant trend in the future of personal health information and medical services, but confidential patient data must be kept secure and private, accessible only by authorized personnel. To ensure the privacy of messages and provide the confidentiality and integrity of information security for medical data, the device requires additional computing power for encryption of the transmission process and associated data.



```
bluetooth.addr==f6:b0:9c:3b:96:82                                                             Expression…
No.  Time       Source               Destination                  Protocol  Length Info
1134 50.952034  f6:b0:9c:3b:96:82 …  SonyMobi_9e:c6:fc (Xperia X) ATT       28 Rcvd Handle Value Notification, Handle: 0x0012 (Generic Attribute
1136 51.470230  SonyMobi_9e:c6:fc …  f6:b0:9c:3b:96:82 ()         ATT       32 Sent Write Command, Handle: 0x0010 (Generic Attribute Profile: Unl
1137 51.471895  SonyMobi_9e:c6:fc …  f6:b0:9c:3b:96:82 ()         ATT       14 Sent Write Command, Handle: 0x0010 (Generic Attribute Profile: Unl
  11… 51.536400  f6:b0:9c:3b:96:82 …  SonyMobi_9e:c6:fc (Xperia X) ATT       29 Rcvd Handle Value Notification, Handle: 0x0012 (Generic Attribute
1140 51.536958  f6:b0:9c:3b:96:82 …  SonyMobi_9e:c6:fc (Xperia X) ATT       32 Rcvd Handle Value Notification, Handle: 0x0012 (Generic Attribute
1141 51.537658  f6:b0:9c:3b:96:82 …  SonyMobi_9e:c6:fc (Xperia X) ATT       32 Rcvd Handle Value Notification, Handle: 0x0012 (Generic Attribute
1142 51.539043  f6:b0:9c:3b:96:82 …  SonyMobi_9e:c6:fc (Xperia X) ATT       32 Rcvd Handle Value Notification, Handle: 0x0012 (Generic Attribute
1143 51.581399  f6:b0:9c:3b:96:82 …  SonyMobi_9e:c6:fc (Xperia X) ATT       32 Rcvd Handle Value Notification, Handle: 0x0012 (Generic Attribute
1144 51.582085  f6:b0:9c:3b:96:82 …  SonyMobi_9e:c6:fc (Xperia X) ATT       32 Rcvd Handle Value Notification, Handle: 0x0012 (Generic Attribute
1145 51.582675  f6:b0:9c:3b:96:82 …  SonyMobi_9e:c6:fc (Xperia X) ATT       32 Rcvd Handle Value Notification, Handle: 0x0012 (Generic Attribute
1146 51.584142  f6:b0:9c:3b:96:82 …  SonyMobi_9e:c6:fc (Xperia X) ATT       32 Rcvd Handle Value Notification, Handle: 0x0012 (Generic Attribute
1147 51.626834  f6:b0:9c:3b:96:82 …  SonyMobi_9e:c6:fc (Xperia X) ATT       32 Rcvd Handle Value Notification, Handle: 0x0012 (Generic Attribute
1148 51.627051  f6:b0:9c:3b:96:82 …  SonyMobi_9e:c6:fc (Xperia X) ATT       32 Rcvd Handle Value Notification, Handle: 0x0012 (Generic Attribute
1149 51.627614  f6:b0:9c:3b:96:82 …  SonyMobi_9e:c6:fc (Xperia X) ATT       32 Rcvd Handle Value Notification, Handle: 0x0012 (Generic Attribute
1150 51.629034  f6:b0:9c:3b:96:82 …  SonyMobi_9e:c6:fc (Xperia X) ATT       32 Rcvd Handle Value Notification, Handle: 0x0012 (Generic Attribute
1151 51.671379  f6:b0:9c:3b:96:82 …  SonyMobi_9e:c6:fc (Xperia X) ATT       32 Rcvd Handle Value Notification, Handle: 0x0012 (Generic Attribute
1152 51.671954  f6:b0:9c:3b:96:82 …  SonyMobi_9e:c6:fc (Xperia X) ATT       32 Rcvd Handle Value Notification, Handle: 0x0012 (Generic Attribute
1153 51.672713  f6:b0:9c:3b:96:82 …  SonyMobi_9e:c6:fc (Xperia X) ATT       32 Rcvd Handle Value Notification, Handle: 0x0012 (Generic Attribute
1154 51.674230  f6:b0:9c:3b:96:82 …  SonyMobi_9e:c6:fc (Xperia X) ATT       14 Rcvd Handle Value Notification, Handle: 0x0012 (Generic Attribute
1155 51.685470  SonyMobi_9e:c6:fc …  f6:b0:9c:3b:96:82 ()         ATT       29 Sent Write Command, Handle: 0x0010 (Generic Attribute Profile: Unl
1156 51.716310  f6:b0:9c:3b:96:82 …  SonyMobi_9e:c6:fc (Xperia X) ATT       17 Rcvd Handle Value Notification, Handle: 0x0016 (Generic Access Pr
1157 51.717934  f6:b0:9c:3b:96:82 …  SonyMobi_9e:c6:fc (Xperia X) ATT       20 Rcvd Handle Value Notification, Handle: 0x001c (Generic Access Pr
1159 52.706430  f6:b0:9c:3b:96:82 …  SonyMobi_9e:c6:fc (Xperia X) ATT       17 Rcvd Handle Value Notification, Handle: 0x0016 (Generic Access Pr
1160 53.604367  SonyMobi_9e:c6:fc …  f6:b0:9c:3b:96:82 ()         ATT       24 Sent Write Command, Handle: 0x0010 (Generic Attribute Profile: Unl
1161 53.696789  f6:b0:9c:3b:96:82 …  SonyMobi_9e:c6:fc (Xperia X) ATT       28 Rcvd Handle Value Notification, Handle: 0x0012 (Generic Attribute
1162 53.697812  f6:b0:9c:3b:96:82 …  SonyMobi_9e:c6:fc (Xperia X) ATT       17 Rcvd Handle Value Notification, Handle: 0x0016 (Generic Access Pr
    [Current Mode: Unknown (-1)]
▼ Bluetooth L2CAP Protocol
    Length: 20
    CID: Attribute Protocol (0x0004)
▼ Bluetooth Attribute Protocol
  ▶ Opcode: Handle Value Notification (0x1b)
  ▶ Handle: 0x0012 (Generic Attribute Profile: Unknown)
    Value: 00020e0588138a1301030a0101034e2300

0000  02 40 20 18 00 14 00 04  00 1b 12 00 00 02 0e 05   .@ ..... ........
0010  88 13 8a 13 01 03 0a 01  01 03 4e 23 00            ........ ..N#.
```

FIGURE 11. The app sends synchronous personal health information commands to the wearable device via Bluetooth.

3) MOBILE APPLICATION

Burp Suite is based on an HTTP proxy used by a web browser. With HTTPS—the secure HTTP protocol that uses SSL—Burp Suite can implement an MITM attack to intercept the HTTP message in plaintext, as shown in Figure 12.

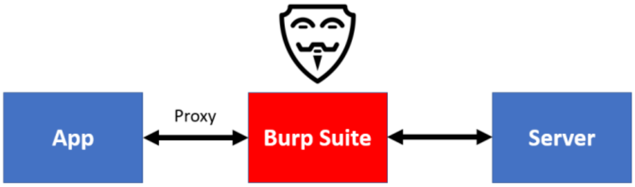
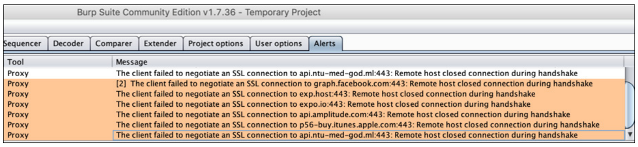


FIGURE 12. The attack path of the mobile app.

The Client obtains a public key certificate from the Server, using an asymmetric cryptography algorithm to negotiate the symmetric key and ensure confidentiality and protection of HTTP messages. SSL is designed to withstand an MITM attack, so HTTPS proxies cannot be implemented through a traditional MITM attack. The purpose of Burp Suite is to disguise itself as the target HTTPS server and convince the client that Burp Suite is the target site. To achieve this, Burp Suite generates a pair of public and private keys, and then binds and encapsulates the public key and target domain name in a certificate. The certificate is verified by convincing the browser of its authenticity. Burp Suite adds a root certificate to the operating system that allows the client to trust all Burp Suite certificates. Burp Suite then has two sets of symmetric keys, one for interacting with the client and one for communicating with the server, where it can obtain the HTTPS plaintext.

In earlier versions of Android, by default, the application trusts not only the system with the Certificate Authority (CA) certificate installed but also the user where the CA certificate has been added. Thus, using an MITM proxy such as Burp Suite, the user can install a fake certificate on the device and then intercept and observe network traffic in plaintext. To prevent an MITM attack, the system can adopt the Certificate-Pinning security mechanism, which prevents attackers from using an MITM attack. Starting with Android 7.0 (API level 24), a new feature allows adding security settings to App resources to prevent an MITM attack; earlier versions of Android require the use of credential binding to prevent an MITM attack. As a result, intercepting application traffic with proxies is harder to implement. The App cannot use an untrusted credential to connect back to the central server, as shown in Figure 13.



Tool	Message
Proxy	The client failed to negotiate an SSL connection to api.ntu-med-god.ml:443: Remote host closed connection during handshake
Proxy	[2] The client failed to negotiate an SSL connection to graph.facebook.com:443: Remote host closed connection during handshake
Proxy	The client failed to negotiate an SSL connection to exp.host:443: Remote host closed connection during handshake
Proxy	The client failed to negotiate an SSL connection to expo.io:443: Remote host closed connection during handshake
Proxy	The client failed to negotiate an SSL connection to api.amplitude.com:443: Remote host closed connection during handshake
Proxy	The client failed to negotiate an SSL connection to p56-buy.itunes.apple.com:443: Remote host closed connection during handshake
Proxy	The client failed to negotiate an SSL connection to api.ntu-med-god.ml:443: Remote host closed connection during handshake

FIGURE 13. The proxy failed to connect with the server.

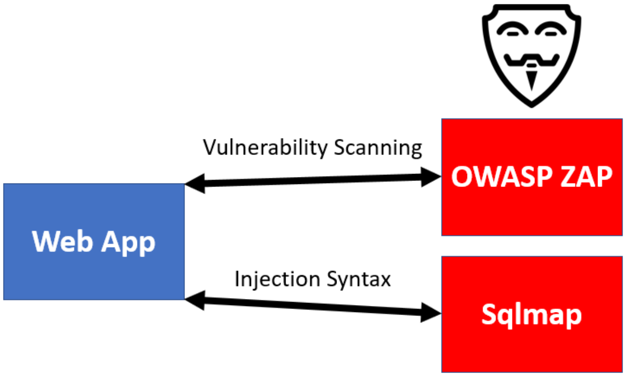


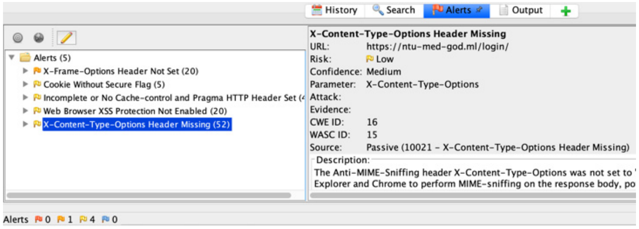
FIGURE 14. Attack paths for web applications.

4) WEB APPLICATION

In the usage scenario architecture analyzed in this study, one component of the web application is responsible for querying personal health data and environmental quality information for the client. We will use penetration testing methods such as automatic scanning tools and manual packet analysis to find possible attack injection points in the web application, as shown in Figure 14.

OWASP ZAP [38] is an open-source project of the Open Web Application Security Project (OWASP) under the category of web vulnerability security scanning and local agent tools. It is a set of multi-functional evaluation tools developed for web systems that performs active scanning, violent searching, and automatic crawling of web pages. OWASP ZAP's local proxy can be used to intercept, snoop, and tamper with client requests and responses, enabling the user to test or attack web application security vulnerabilities under multiple test scenarios.

Another tool is Sqlmap, an open-source penetration testing tool that automatically detects and exploits SQL injection vulnerabilities and servers accessing the database. It has a powerful detection engine and attack injection syntax with multiple databases and can both extract fingerprint identification information from the database and execute SQL commands to access compromised data or control the database.

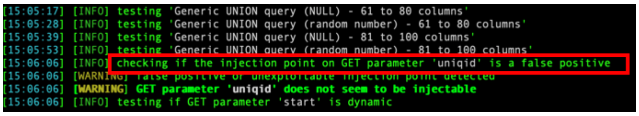


Alerts (5)
- X-Frame-Options Header Not Set (20)
- Cookie Without Secure Flag (5)
- Incomplete or No Cache-control and Pragma HTTP Header Set (4
- Web Browser XSS Protection Not Enabled (20)
- X-Content-Type-Options Header Missing (52)

X-Content-Type-Options Header Missing
URL: https://ntu-med-god.ml/login/
Risk: Low
Confidence: Medium
Parameter: X-Content-Type-Options
Attack:
Evidence:
CWE ID: 16
WASC ID: 15
Source: Passive (10021 – X-Content-Type-Options Header Missing)
Description:
The Anti-MIME-Sniffing header X-Content-Type-Options was not set to '
Explorer and Chrome to perform MIME-sniffing on the response body, po

Alerts 0 1 4 0

FIGURE 15. Scan the web application for vulnerabilities.



```
[15:05:17] [INFO] testing 'Generic UNION query (NULL) - 61 to 80 columns'
[15:05:28] [INFO] testing 'Generic UNION query (random number) - 61 to 80 columns'
[15:05:39] [INFO] testing 'Generic UNION query (NULL) - 81 to 100 columns'
[15:05:53] [INFO] testing 'Generic UNION query (random number) - 81 to 100 columns'
[15:06:06] [INFO] checking if the injection point on GET parameter 'uniqid' is a false positive
[15:06:06] [WARNING] false positive or unexploitable injection point detected
[15:06:06] [WARNING] GET parameter 'uniqid' does not seem to be injectable
[15:06:06] [INFO] testing if GET parameter 'start' is dynamic
```

FIGURE 16. SQL injection attacks on the web application.

As shown in Figure 15, the results from scanning using OWASP ZAP uncovered web pages that have both moderate- and low-level risks for the following web vulnerabilities: (1) X-Frame-Option Header Not Set, which is used to indicate whether a website can load an iframe element, ensuring its content is not maliciously-embedded in the site and preventing click-hijacking attacks, (2) Cookie Without Secure Flag, which makes cookies available only over HTTPS, and (3) X-Content-Type-Options Header Missing, which prevents content-type from being tampered with maliciously. To address these vulnerabilities, we add security settings to the back-end web server. Figure 16 illustrates that the penetration test portion of Sqlmap uses many parameter injection points and can be used by the client in the attacker's mind but fails to obtain the access right to the database. Accordingly, it is recommended that a relevant protection mechanism is used: either a web application firewall (WAF) or input verification.

V. RISK ASSESSMENT FRAMEWORK

The source of risk is the focus of system safety protection. Several studies have proposed effective information security evaluation methods. Factors affecting system information security policy include the following:

(1) Assets: identify assets and value;

(2) Threats: exploit the vulnerability of an asset to harm it;

(3) Vulnerabilities: system weaknesses that can be exploited and lead to unexpected results; and

(4) Impact probability: the probability of each vulnerability to result in damage caused by threat exploitation.

TABLE 2. DREAD model of attack potential.

Threat	High (3)	Medium (2)	Low (1)
Damage Potential	The attacker can ignore the protection mechanism of the security system and obtain authorization from the administrator to execute arbitrary programs.	The attacker can obtain the user's sensitive information.	The attacker can only obtain general identifiable information.
Reproducibility	The attack can be made every time and can be launched at any time without restriction.	The attack can only be repeated in a particular situation or point in time.	Even if there are known security vulnerabilities or the technical complexity is too high, it is difficult to repeat the attack.
Exploitability	The novice programmer can attack directly using simple scripts or tools.	The skilled programmer can perform attacks but requires relevant technical knowledge.	Requires the expertise of an information security professional who is skilled in each attack.
Affected Users	All users.	The users in the range specified by a particular condition.	Affects only a small percentage of users, and can recover quickly.
Discoverability	The vulnerability information is defined as severity levels that exist in common service functions and are easy to implement repeatedly.	This vulnerability exists in a service that is rarely used, and its impact is so small that it requires specific information to identify it.	This vulnerability attack is vague. The implementation requires specific conditions and is extremely difficult and not easy to detect.

TABLE 3. Ratings of potential attack risks.

Target	Threat	D	R	E	A	D	Total	Rating
Airbox	Obtain sensitive information by monitoring network traffic	1	1	2	2	1	7	Low
Wearable Device	Use Ubertooth One to fetch Bluetooth packets and analyze the sensitivity of the information	1	2	1	1	1	6	Low
Wearable Device	Use HCI Log to capture Bluetooth packets and analyze service information vulnerability	1	2	1	1	1	6	Low
Mobile App	Use a proxy such as Burp Suite to conduct a man-in-the-middle attack on the application	2	3	1	1	1	8	Medium
Web App	Obtain identity credentials by monitoring traffic	3	2	2	2	2	11	Medium
Web App	Inject SQL commands into the application	3	2	2	3	3	13	High

Establishing a complete information security policy is difficult. In addition to the security and reliability requirements of the information infrastructure, issues of data and human risks, such as data errors and equipment abuse, must be considered. We can adhere to relevant international information security standards to improve policy, such as

TABLE 4. Risk principles of the STRIDE model.

Threat	Security Property	Description	Action	Countermeasure
Spoofing	Authentication	Unauthorized access to the system using an illegal identity or illegal erroneous data to mislead the data set.	This threat uses the credentials of other legitimate users to trick the system into logging in or accessing data.	1. Do not allow sensitive data to be passed in plaintext. 2. Do not allow credentials to be stored in plaintext. 3. Use more stringent authentication.
Tampering	Integrity	Malicious unauthorized modification of data or code to deceive the recipient.	The threat of malicious modification of data, such as data in databases, file systems, and network traffic.	1. Use legal digital signatures. 2. Perform fingerprint comparisons using hash values.
Repudiation	Non-repudiation	Users declare that they have not and cannot perform this action.	The event lacks the ability to audit and track evidence.	1. Use system access log auditing and tracking. 2. Use legal digital signatures.
Information disclosure	Confidentiality	Data information is disclosed to unauthorized users.	Access to unauthorized files or data in transit.	1. Use publicly-authenticated and more secure encryption algorithms. 2. Do not allow sensitive to be passed in plaintext.
Denial of service	Availability	Deny or interrupt the services of legitimate users from using system applications or services.	Deny access to valid users, such as making web servers, databases, and applications temporarily unavailable.	1. Use bandwidth control or load-balancing to disperse the flow.
Elevation of privilege	Authorization	Users with limited permissions can use the application to enhance their permissions without authorization.	Gain privileged access to resources to gain unauthorized access to information or damage the system.	1. Follow the principle of least privilege.

ISO/IEC 27001:2013 [39], ISO/IEC 27002:2013 [40], and NIST SP 800-30 [41].

Quantitative risk analysis, qualitative risk analysis, or a combination of the two, can be used to assess and prioritize risk. Quantitative risk analysis is used to calculate the probability of event occurrence, possible loss, and historical data for statistics and reference. Qualitative risk analysis is used to observe the occurrence of risks according to the judgment, intuition, and experience of managers, and to analyze these risks based on probabilities, the severity of threats, and the sensitivity of assets. Qualitative risk analysis uses rigorous procedures to assess the value of assets and the likelihood of threats, and then classifies risks by the knowledge and experience of experts, such as high, medium, and low.

As this research framework is a new system without previous data, we use the DREAD and STRIDE models for risk assessment using the qualitative analysis method.

A. DREAD MODEL

The DREAD model represents Damage, Reproducibility, Exploitability, Affected users, and Discoverability, as shown in Table 2. It is used to calculate the risk of each threat to the application, as shown in Table 3. Severity levels are assigned to threats to enable the identification of threat mitigations by severity. In the definition of a comprehensive score of severity level, Low ranges from 5 to 7, Medium ranges from 8 to 11, and High ranges from 12 to 15.

B. STRIDE MODEL

To identify threats, we need to broadly classify them into the categories shown in Table 4. This threat classification provides a structured way to identify application threats. The threat identification process uses the STRIDE model to classify threats and examine aspects of application security. The STRIDE model classifies application threats according to the target and purpose of the attack, thus helping developers develop security policies. It also includes countermeasures against all threat categories.

VI. CONCLUSION

In this study, we provide a more complete information security risk analysis process. We first present use cases through a DFD to evaluate the system and workflows on the attack surface area. Then, we use an abuse model to identify several possible attack paths. To further identify potential future threats and preventive measures, we use the DREAD and STRIDE models as guidelines. Based on the test results in this study—for third-party IoT devices, a mobile application, and a self-developed data center management platform—we find no serious information disclosures across entry points. With the development of intelligent IoT applications, big data, and artificial intelligence, IoT devices are used in diverse fields. With many applications come many potential threats and risks. We need to be careful not to compromise personal health data on wearable devices, mobile apps, data center platforms, third-party systems, and other platforms.

IoT information security testing and risk assessment have become important components of the development of international standards such as Considerations for Managing IoT Cybersecurity and Privacy Risks (NISTIR 8228) [42] and IoT Security Guidelines Overview Document (GSMA) [43]. Other well-known international standards such as Information Technology Security Techniques Guidelines for Security and Privacy in the Internet of Things (ISO/IEC WD 27030) [44] are still under development. In future work, we will reference international standards to carry out more accurate verification and audit steps to establish a superior security protection mechanism.

REFERENCES

[1] P. Punithavathi, S. Geetha, M. Karuppiah, S. K. H. Islam, M. M. Hassan, and K.-K. R. Choo, "A lightweight machine learning-based authentication framework for smart IoT devices," *Inf. Sci.*, vol. 484, pp. 255–268, May 2019.

[2] Y. Sun, H. Song, A. J. Jara, and R. Bie, "Internet of Things and big data analytics for smart and connected communities," *IEEE Access*, vol. 4, pp. 766–773, 2016.

[3] S. M. R. Islam, D. Kwak, M. H. Kabir, M. Hossain, and K.-S. Kwak, "The Internet of Things for health care: A comprehensive survey," *IEEE Access*, vol. 3, pp. 678–708, 2015.

[4] S. Pirbhulal, W. Wu, and G. Li, "A biometric security model for wearable healthcare," in *Proc. IEEE Int. Conf. Data Mining Workshops (ICDMW)*, Nov. 2018, pp. 136–143.

[5] A. Raji, P. G. Jeyasheeli, and T. Jenitha, "IoT based classification of vital signs data for chronic disease monitoring," in *Proc. 10th Int. Conf. Intell. Syst. Control (ISCO)*, Jan. 2016, pp. 1–5.

[6] J.-H. Han, Y. Jeon, and J. Kim, "Security considerations for secure and trustworthy smart home system in the IoT environment," in *Proc. Int. Conf. Inf. Commun. Technol. Converg. (ICTC)*, Oct. 2015, pp. 1116–1118.

[7] H. Tao, M. Z. A. Bhuiyan, A. N. Abdalla, M. M. Hassan, J. M. Zain, and T. Hayajneh, "Secured data collection with hardware-based ciphers for IoT-based healthcare," *IEEE Internet Things J.*, vol. 6, no. 1, pp. 410–420, Feb. 2019.

[8] M. Frustaci, P. Pace, G. Aloi, and G. Fortino, "Evaluating critical security issues of the IoT world: Present and future challenges," *IEEE Internet Things J.*, vol. 5, no. 4, pp. 2483–2495, Aug. 2018.

[9] J. Choi, Y. Shin, and S. Cho, "Study on information security sharing system among the industrial IoT service and product provider," in *Proc. Int. Conf. Inf. Netw. (ICOIN)*, Jan. 2018, pp. 551–555.

[10] S. Sharmeen, S. Huda, J. H. Abawajy, W. N. Ismail, and M. M. Hassan, "Malware threats and detection for industrial mobile-IoT networks," *IEEE Access*, vol. 6, pp. 15941–15957, 2018.

[11] J. Abawajy, S. Huda, S. Sharmeen, M. M. Hassan, and A. Almogren, "Identifying cyber threats to mobile-IoT applications in edge computing paradigm," *Future Gener. Comput. Syst.*, vol. 89, pp. 525–538, Dec. 2018.

[12] S. Rizvi, A. Kurtz, J. Pfeffer, and M. Rizvi, "Securing the Internet of Things (IoT): A security taxonomy for IoT," in *Proc. 17th IEEE Int. Conf. Trust, Secur. Privacy Comput. Commun./12th IEEE Int. Conf. Big Data Sci. Eng. (TrustCom/BigDataSE*, Aug. 2018, pp. 163–168.

[13] S. Seneviratne, Y. Hu, T. Nguyen, G. Lan, S. Khalifa, K. Thilakarathna, M. Hassan, and A. Seneviratne, "A survey of wearable devices and challenges," *IEEE Commun. Surveys Tuts.*, vol. 19, no. 4, pp. 2573–2620, 4th Quart., 2017.

[14] C. Kolias, G. Kambourakis, A. Stavrou, and J. Voas, "DDoS in the IoT: Mirai and other botnets," *Computer*, vol. 50, no. 7, pp. 80–84, 2017.

[15] R. Williams, E. McMahon, S. Samtani, M. Patton, and H. Chen, "Identifying vulnerabilities of consumer Internet of Things (IoT) devices: A scalable approach," in *Proc. IEEE Int. Conf. Intell. Secur. Inform. (ISI)*, Jul. 2017, pp. 179–181.

[16] S. Siboni, V. Sachidananda, Y. Meidan, M. Bohadana, Y. Mathov, S. Bhairav, A. Shabtai, and Y. Elovici, "Security testbed for Internet-of-Things devices," *IEEE Trans. Rel.*, vol. 68, no. 1, pp. 23–44, Mar. 2019.

[17] Nmap. (2019). *Free Security Scanner for Network Exploration and Security Audits*. [Online]. Available: https://nmap.org/

[18] Nessus. (2019). *A Network Vulnerability Scanner, Tenable Network Security*. [Online]. Available: http://www.tenable.com/products/nessusvulnerability- scanner

[19] OpenVAS. (2019). *A Framework for Vulnerability Scanning and Vulnerability Management*. [Online]. Available: http://openvas.org/

[20] Wireshark. (2019). *A Network Protocol Analyzer*. [Online]. Available: https://www.wireshark.org/

[21] Burp Suite. (2019). *A Graphical Tool for Web Application Security*. [Online]. Available: https://portswigger.net/burp

[22] O. Arias, J. Wurm, K. Hoang, and Y. Jin, "Privacy and security in Internet of Things and wearable devices," *IEEE Trans. Multi-Scale Comput. Syst.*, vol. 1, no. 2, pp. 99–109, Apr./Jun. 2015.

[23] K. Lotfy and M. L. Hale, "Assessing pairing and data exchange mechanism security in the wearable Internet of Things," in *Proc. IEEE Int. Conf. Mobile Services (MS)*, Jun./Jul. 2016, pp. 25–32.

[24] A. R. Chandan and V. D. Khairnar, "Security Testing Methodology of IoT," in *Proc. Int. Conf. Inventive Res. Comput. Appl. (ICIRCA)*, Jul. 2018, pp. 1431–1435.

[25] M. Lee, K. Lee, J. Shim, S.-J. Cho, and J. Choi, "Security threat on wearable services: Empirical study using a commercial smartband," in *Proc. IEEE Int. Conf. Consum. Electron.-Asia (ICCE-Asia)*, Oct. 2016, pp. 1–5.

[26] P. K. Chouhan, S. McClean, and M. Shackleton, "Situation assessment to secure IoT applications," in *Proc. 5th Int. Conf. Internet Things, Syst., Manage. Secur.*, Oct. 2018, pp. 70–77.

[27] Z. Yang and Z. Zhang, "The study on resolutions of STRIDE threat model," in *Proc. 1st IEEE Int. Symp. Inf. Technol. Appl. Educ.*, Nov. 2007, pp. 271–273.

[28] S. Hussain, H. Erwin, and P. Dunne, ''Threat modeling using formal methods: A new approach to develop secure Web applications,'' in *Proc. 7th Int. Conf. Emerg. Technol.*, Sep. 2011, pp. 1–5.

[29] V. L. Shivraj, M. A. Rajan, and P. Balamuralidhar, ''A graph theory based generic risk assessment framework for Internet of Things (IoT),'' in *Proc. IEEE Int. Conf. Adv. Netw. Telecommun. Syst. (ANTS)*, Dec. 2017, pp. 1–6.

[30] R. Khan, K. McLaughlin, D. Laverty, and S. Sezer, ''STRIDE-based threat modeling for cyber-physical systems,'' in *Proc. IEEE PES Innov. Smart Grid Technol. Conf. Eur. (ISGT-Eur.)*, Sep. 2017, pp. 1–6.

[31] S. N. M. García, J. L. Hernández-Ramos, and A. F. Skarmeta, ''Test-based risk assessment and security certification proposal for the Internet of Things,'' in *Proc. IEEE 4th World Forum Internet Things (WF-IoT)*, Feb. 2018, pp. 641–646.

[32] A. Nhlabatsi, J. B. Hong, D. S. Kim, R. Fernandez, N. Fetais, and K. M. Khan, ''SpiralŜRA: A threat-specific security risk assessment framework for the cloud,'' in *Proc. IEEE Int. Conf. Softw. Qual., Rel. Secur. (QRS)*, Jul. 2018, pp. 367–374.

[33] F. Alsubaei, A. Abuhussein, and S. Shiva, ''Security and privacy in the Internet of medical things: Taxonomy and risk assessment,'' in *Proc. IEEE 42nd Conf. Local Comput. Netw. Workshops (LCN Workshops)*, Oct. 2017, pp. 112–120.

[34] J. Hernández-Serrano, L. J. Muñoz, O. León, L. Mikkelsen, H.-P. Schwefel, and A. Broring, ''Privacy risk analysis in the IoT domain,'' in *Proc. Global Internet Things Summit (GIoTS)*, Jun. 2018, pp. 1–6.

[35] J. Sturgess, J. R. C. Nurse, and J. Zhao, ''A capability-oriented approach to assessing privacy risk in smart home ecosystems,'' in *Proc. Living Internet Things, Cybersecur. (IoT)*, Mar. 2018, pp. 1–8.

[36] Microsoft Threat Modeling Tool. (2019). *A Visualizing Tool for Data Flows and Security Boundaries*. [Online]. Available: https://aka.ms/threatmodelingtool

[37] Ubertooth One. (2019). *An Open-Source Wireless Development Platform Suitable for Bluetooth Experimentation*. [Online]. Available: http://ubertooth.sourceforge.net/

[38] OWASP ZAP. (2019). *A Free Security Tool for Finding the Vulnerabilities in the Web Application*. [Online]. Available: https://www.owasp.org/index.php/OWASP_Zed_Attack_Proxy_Project

[39] *Information Technology–Security Techniques–Information Security Management Systems—Requirements*, Standard ISO/IEC 27001:2013, 2019. [Online]. Available: https://www.iso.org/standard/54534.html

[40] *Information Technology–Security Techniques–Code of Practice for Information Security Controls*, Standard ISO/IEC 27002:2013, 2019. [Online]. Available: https://www.iso.org/standard/54533.html

[41] *Guide for Conducting Risk Assessments*, Standard NIST SP 800-30, 2019. [Online]. Available: https://csrc.nist.gov/publications/detail/sp/800-30/rev-1/final

[42] *Considerations for Managing Internet of Things (IoT) Cybersecurity and Privacy Risks*, Standard NISTIR 8228, 2019. [Online]. Available: https://csrc.nist.gov/publications/detail/nistir/8228/final

[43] (2019). *GSMA—IoT Security Guidelines Overview Document*. [Online]. Available: https://www.gsma.com/iot/iot-security-guidelines-overview-document/

[44] *Information Technology–Security Techniques–Guidelines for Security and Privacy in Internet of Things (IoT)Title Missing*, Standard ISO/IEC WD 27030, 2019. [Online]. Available: https://www.iso.org/standard/44373.html

TZU WEI TSENG received the B.S.E.E. degree from Tamkang University, in 2015, and the M.S. degree in biomedical informatics from Taipei Medical University, in 2017. He is currently pursuing the Ph.D. degree with the Department of Computer Science and Information Engineering, National Taiwan University. His research interests include network security, the Internet of Things security, risk assessment, electronic medical records systems, and patient privacy protection.

CHIA TUNG WU received the B.S.T. degree from Tunghai University, in 2015, and the M.S. degree in biomedical electronics and bioinformatics from National Taiwan University, in 2017. He is currently pursuing the Ph.D. degree with the Department of Computer Science and Information Engineering, National Taiwan University. His main research interests include medical information systems, the Internet of Things, and artificial intelligence.

FEIPEI LAI (SM'94) received the B.S.E.E. degree from National Taiwan University, in 1980, and the M.S. and Ph.D. degrees in computer science from the University of Illinois at Urbana–Champaign, in 1984 and 1987, respectively. He was a Vice Superintendent of National Taiwan University Hospital, the Chairman of the Taiwan Network Information Center, and a Visiting Professor with the Department of Computer Science and Engineering, University of Minnesota, Minneapolis, USA. He was also a Guest Professor with the University of Dortmund, Germany, and a Visiting Senior Computer System Engineer with the Center for Supercomputing Research and Development, University of Illinois at Urbana–Champaign. He is currently a Professor with the Department of Computer Science and Information Engineering, the Department of Electrical Engineering, and the Graduate Institute of Biomedical Electronics and Bioinformatics, National Taiwan University. He currently holds ten Taiwan patents and four U.S. patents.