JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue
JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)
An International Scholarly Open Access, Peer-reviewed, Refereed Journal

Iot Platform Traffic Signature Identification

K.Anusha, K.Nitya, A.Asritha, K.Manikanta, P.Hemanth
UG Student, Assistant Professor, UG Student
Usha Rama College of Engg & Tech

Abstract : The proposal of "fingerprinting" for IoT platform traffic represents a significant advancement in bolstering the security of IoT ecosystems. By introducing a tool like IOTPF, capable of discerning between the traffic of various mainstream IoT platforms, a critical need in cybersecurity is being addressed. The structured workflows devised for traffic capturing, fingerprint feature extraction, and model construction offer a systematic framework for identifying and analysing IoT platform traffic. This approach streamlines intrusion detection and vulnerability assessment processes,allowing security experts to concentrate on the specific traffic associated with IoT platforms. Conducting usability and performance tests on IOTPF is pivotal to validate its effectiveness. Through thorough evaluation across diverse conditions and scenarios, confidence in itsreal-world applicability and reliability can be ensured. Going forward, sustained research and development in this field will be imperative to keep pace with the ever- evolving landscape of IoT security threats. As IoT devices become increasingly ubiquitous and interconnected, the demand for robust intrusion detectionmechanisms and tools like IOTPF will only continue to escalate**.**

***IndexTerms* - Fingerprinting, Internet of Things,Transmitting sensitive, Traffic analysis**

I. Introduction

Internet-connected consumerdevices, also known as smart home or Internet of things (IoT) devices, have seen increasingly widespread adoption in recent years. These new technologies create new challenges and research opportunities for ubiquitous computing. Conventional challenges include security (e.g., distributed denial of service attacks by IoT botnets); privacy (e.g., toys transmitting sensitive informationabout children to third parties); and device inventory that rely on the collection or analysis of data from these devices. Today, almost every part of our lives is digitalized: products, services, businessoperations. With the constant increase in connectivity and digitalization, huge amounts of personal data are often collected without any real justification or need. On the otherhand, there is a growing concern over who is in possessionof this data and how it is being used. With increasingly more privacy-aware individuals and with ever stricter data protection requirements (GDPR, e Privacy CCPA), organizations are seeking a compromise that will enable them to collect and analyse their users' data, to innovate, optimize, and grow their businesses, while at the same timecomply with legal frameworks and keep trust and confidence of their users. When individuals themselves usetechnologies like end-to-end encryption to protect their data, this can greatly improve their privacy online because the service providers never see raw data. But when a serviceprovider does not have access to raw data, it cannot analysethe data and it thus cannot offer functionalities like search or data classification. Indeed, almost all rich functionality to which users are accustomed today is out of the question when encryption is used. However, there are encryption techniques which do not impose a drastic reduction of datautility and consequently functionality. The probably most known such technique is Homomorphic Encryption (HE). HE enables additions and multiplications over the encrypted data, which consequently enables higher level functionality such as machine learning on the encrypted data. However, HE is computationally expensive and significantly reduces service performance. Another technique, perhaps lesser known, is Functional key can Encryption (FE). Similarly, as HE, it allows computationon encrypted data. More precisely, the owner of a decryption key can learn a function of the encrypted data.This gives a possibility to use the encrypted data for various analysis or machine learning models by controlling the information one can get from it. In this paper we present first two fully-fledged FE libraries, we outline how they can be used to build machine learning services on encrypted data, and we discuss strengths andlimitations are compared to the he approach. The numberof IoT devices connected to the Internet, such as cameras,voice-activated assistants, network-attached storage devices, smart appliances, etc., has been growing at an increasingly accelerating pace . Unfortunately,partly due to market forces that push down the cost of such devices, even well-known elementary security mechanism are often neglected by IoT device developers and vendors. This has caused a new security crisis of sorts, which has been highlighted by recent major Internet-wide security incidents originating from massivenumbers of compromised IoT devices .It is therefore important to enumerate potentially vulnerable IoT devices across the Internet, as a way to estimate global Internet risks and provide a way for network operators to assess the hygiene of their own networks. Forinstance, systems such as Censys and Shodan perform periodic scans of the entire IPv4 space, using active probing techniques to identify network services reachable from the public Internet. These services leverage banner grabbing and other fingerprinting approaches to identify and enumerate the type of devicesthat expose those network services, which includes identifying exposed IoT devices. This allows organizations to assess what others can learn about their networks, thus gaining a better understanding of the attack surface they expose. Unfortunately, active probingis unable to identify IoT devices that are hosted behind some middleboxes, such as NATs or firewalls. At the same time, mapping these "hidden" IoT devices is important.

A)Related Work

The identification of network users and that of devices are two differentiated research directions butare closely related. Earlier device identification technologies mainly obtain the information of the hardware, operating systems, network protocols, and other parameters by collecting and analyzing the physical signalsor traffic generated by the device. For example, in the physical layer , the TCP packet time stamps are analyzedto obtain the clock skew , and the Ethernet frames are analyzed to obtain the differences among the analog signalsof different devices . While in the operating system layer, the active scanning algorithm used by the wireless device driver might be inferred by analyzing the interval time of 802.11 probe request frames . In the applicationlayer, the User-Agent field, IP address, browser cookie, user login ID, and other identity information are extracted through the traffic analysis in clear text . The interval time, number, direction, and other attributes of the encrypted wireless packets are analyzed for the distinction of different terminal applications . Other researches have applied different threat models to achieve the identification of devices, e.g., the device recognition basedon browsers and that based on mobile applications.

The above-mentioned identification technologies are merely, in essence, the identification of a single browser or a single terminal device. They are far from being capable enough to identify the user's cross-device activities. For example, in the scenario of intrusion detection, if some intruder occupies an authorized device, we cannot detect the intrusion by using the device identification technologies. So it is necessary to carry out the research on the user identification technology based onbehavioral fingerprints. Essentially, user identification technologies based on behavioral fingerprinting are biometric, which use the inherent physiological characteristics or behavioural characteristics of the human body for identification. They can be categorized into two types. The former one has been widely used by employingthe characteristics of human body parts, such process, and as fingerprint identification, face identification, and iris identification. The behaviour-based identification technology extracts the features for identification with the information of the user's operation skills, knowledge, styles, preferences, and strategies revealed inbehaviours. For example, researchers have found that different users differentiate from each other in moving, clicking, dragging, and releasing the mouse . Some may be different in key stroking when keyboarding .All of these differences can help to extract fingerprints for effective identification. In the network area, users have different behavioral patterns for network access dueto different preferences, habits, etc. Different behavior patterns lead to different traffic flows. Therefore,researchers believe that the network traffic generated by the user can be regarded as biometric for user identification .

II. LITERATURE REVIEW

Recent research efforts have focused on ensuring the safety and security of IoT devices, particularly concerning physical interaction control. Ding and Hu (2018) addressed this issue by exploring the safety implications of IoT device physical interaction control, highlighting potential vulnerabilities and proposing strategies for mitigation. Fernandes et al. (2016) conducted a comprehensive security analysis of emerging smart home applications, shedding light on various security risks and vulnerabilities associated with these systems. Furthermore, Zhang et al. (2019) delved into the security risks posed by voice-controlled third-party functions on virtual personal assistant systems, offering insights into understanding and mitigating these risks effectively. Additionally, Yao et al. (2019) focused on identifying privilege separation vulnerabilities in IoT firmware using symbolic execution techniques, providing valuable insights into firmware security. Beyond specific research endeavors, the industry has also witnessed the emergence of various IoT platforms and ecosystems, such as Joylink by JD.com (2021), Mijia by Xiaomi (2021), and iRobot (2021), each with its own set of security considerations and challenges. Moreover, standards organizations like the International Electrotechnical Commission (IEC) have published guidelines, such as the "IoT 2020: Smart and Secure IoT Platform" whitepaper (IEC, 2020), aimed at promoting smart and secure IoT platforms. Major technology companies like Amazon (2021) with AWS IoT Core and Samsung (2021) with SmartThings have also made significant contributions to the IoT ecosystem, offering robust platforms with built-in security features and capabilities.

Recent research has extensively explored security issues within the Internet of Things (IoT) landscape, aiming to bolster a user-centric authorization framework tailored for IoT environments, emphasizing user control and privacy Zhou et al. (2019) delved into the security implications of interactions among IoT devices, mobile apps, and cloud platforms in smart home ecosystems, highlighting the need for better understanding and management of security risks in such interactions. Alrawi et al. (2019) presented a security evaluation of home-based IoT deployments, shedding light on vulnerabilities and recommending strategies for enhanced security. Jia et al. (2020) focused on the security risks posed by general messaging protocols on IoT clouds, identifying potential threats and proposing mitigation techniques. Furthermore, Zhu et al. (2019) proposed FIoT, a framework aimed at detecting memory corruption in lightweight IoT device firmware, addressing a critical security concern. In addition to academic research, industry giants like Microsoft (2021) have contributed to the IoT security landscape with platforms like Azure, offering robust tools and services to address security challenges in IoT deployments. These efforts collectively contribute to enhancing the security posture of IoT ecosystems and safeguarding against potential threats.

III. PROPOSED METHODOLOGY

When a large number of traditional household hardware vendors enter the IoT application and add intelligentmodules to their own hardware devices, they are always faced with the challenge of designing and implementing cloud services for product users, which is beyond the business scope of traditional hardware device manufacturers, thus making the IoT platform emerge as akind of centralized IoT applications. The IoT platform isgenerally provided by large network service providers, who also provide traditional cloud services at the same time. Amazon's AWS IoT Core service is a typical case.Amazon is a large cloud service provider with rich experience in designing and developing cloud service solutions while possessing advanced cloud serviceinfrastructure provides software development kits (SDKs) on the device side and mobile side, as well as management systems in the cloud as a whole IoT platformsolution. With this solution, device manufacturers can realize IoT services suitable for their own hardware products through simple configuration and development deploy them using Amazon's cloud service infrastructure. At the same time, the large-scale

IoT platform integrates different IoT hardware vendors to a certain extent, so as toavoid the compatibility problem. There are more than five hundreds of IoT platform solutions that have emerged in recent years . In this work, we focus on the popular mainstream IoT platforms. We define the concept of the mainstream from two aspects. From an industry perspective, some IoT platforms have a large level of market share, which are built by large network service providers or information technology industry infrastructure providers, and have a vast user base. Generally speaking, they provide entire IoT platform solutions for many hardware manufacturers at the same time, and have individual integration capabilities within the industry. Thispart of the IoT platform has high research value. From the academic point of view, the design of mainstream IoT platforms has many commonness, and the research on these platforms can be extended to other platforms, which have high research value. Scholars have studied architecture andpossible security problems of these IoT platforms from various aspects, and published achievements with certain academic level. Mainstream IoT platforms also provide many user-friendly services and are strong in functionality,which has been recognized by a large scale of users. However, the security problem is difficult to guarantee dueto the complex functions, which has attracted the attention of a large number of researchers.

***A)Traffic decryption**:*

To extract the essential features required for precise fingerprinting, decryption of encrypted network traffic is paramount. Table-I provides insight into the protocols employed by various types of traffic and the feasibility of decrypting captured traffic. For decrypting HTTPS communication, we employ a Man-in-the-Middle (MITM)attack during the capturing process. The MITM attackinvolves intercepting and relaying communication betweenparties, allowing us to decrypt the encrypted HTTPS trafficfor subsequent analysis and feature extraction. Thisapproach enables us to uncover crucial details necessary for

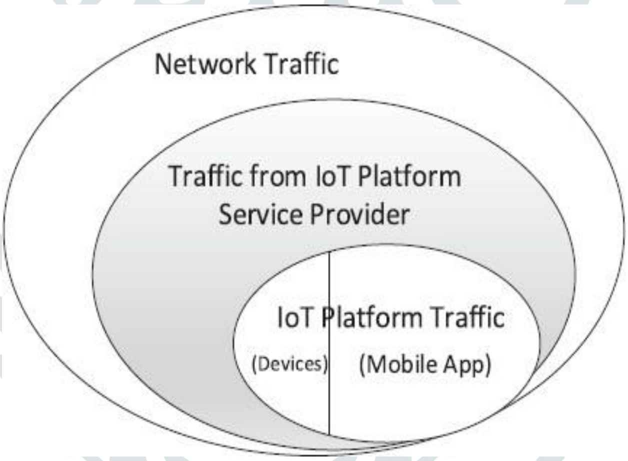
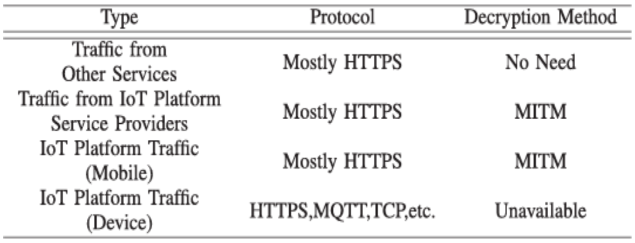


Fig.1. Containment relationship between IoT platformtraffic and network traffic

Table-I PROTOCOL , DECRYPTION METHOD



Type	Protocol	Decryption Method
Traffic from Other Services	Mostly HTTPS	No Need
Traffic from IoT Platform Service Providers	Mostly HTTPS	MITM
IoT Platform Traffic (Mobile)	Mostly HTTPS	MITM
IoT Platform Traffic (Device)	HTTPS,MQTT,TCP,etc.	Unavailable

The decryption and capture of HTTP(S) sessions are facilitated by the Fiddler tool , which integrates capture and decryption functionalities. In our testenvironment, comprising interactions between a mobile app and an IoT platform, alongside a MITM device for packet capturing and multiple IoT devices, we successfully obtained a substantial volume of plaintext traffic. This was achieved by installing and trusting Fiddler's root certificate on the mobile device, and routingall traffic through the MITM device equipped with the Fiddler tool. This process encompassed not only IoT traffic but also non-IoT traffic from service providers of IoT platforms. However, our analysis revealed a diverse array of protocols employed across different layers (application, transport, and data link) between devices and IoT cloud servers, as illustrated in Table I. Moreover proprietary IoT platform solutions often employ encryptionfor IoT device traffic using private protocols, posing challenges for MITM attacks, particularly in cases of SSL/TLS encryption where trust in third-party rootcertificates is required. Nevertheless, some IoT platforms exhibit discernible features even in encrypted traffic, enabling us to achieve fingerprinting successfully.

According to cloud's interacted entities (i.e., mobile app and devices), IoT platform traffic can be further divided into two subtypes. The containment relationship between original network traffic, traffic from IoT platform providers, and IoT platform traffic is shown in Fig. 1.

B) Fingerprint Features Identification:

As described in the preceding section, our construction of IoT platform fingerprints primarily relies on decrypted IoTplatform traffic transmitted over the HTTP(S) protocolbetween mobile applications and IoT platform cloud servers. A comprehensive breakdown of the typicalstructure of HTTP packets in various IoT platforms isprovided in Appendix A. These HTTP packets typically encompass HTTP requests and HTTP responses. The HTTP requests, originating from the client and directed towards the cloud server, typically include the HOST While HTTP requests typically contain identifiable information facilitating platform differentiation, HTTP responses may lack obvious identification markers. In the feature extraction process of HTTP responses, we categorize the difficulty into three levels based on the varying implementations across different IoT platforms, each requiring distinct measurement approaches. For each testedplatform's traffic, a range of features is employed to facilitate fingerprinting.

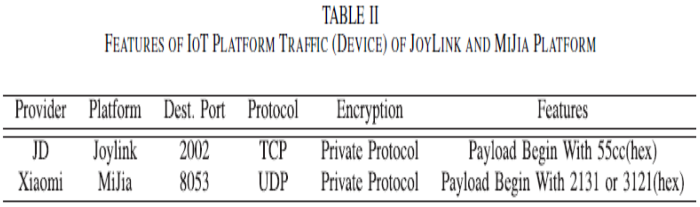


TABLE II
FEATURES OF IOT PLATFORM TRAFFIC (DEVICE) OF JOYLINK AND MIJIA PLATFORM

Provider	Platform	Dest. Port	Protocol	Encryption	Features
JD	Joylink	2002	TCP	Private Protocol	Payload Begin With 55cc(hex)
Xiaomi	MiJia	8053	UDP	Private Protocol	Payload Begin With 2131 or 3121(hex)

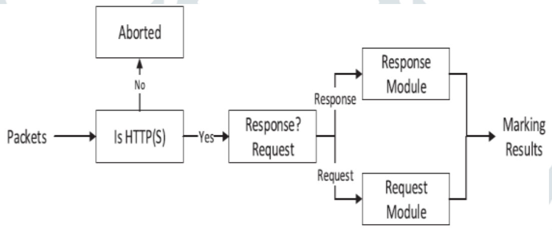


Fig. 2. Workflow of IoTPF with one HTTP(S) session.

C)Fingerprint Features:

1)Easy level: Some platforms like Samsung SmartThings have fixed values in specific fields (e.g., "server" field) that directly identify the platform.
2)Medium level: Other platforms, like AWS IoT, require additional analysis and comparison with non-IoT traffic to distinguish them accurately. This may involve filtering based on specific tags or labels unique to IoT services.
3)Difficulty level: Certain platforms, such as MiJia, may have less helpful information in their response packets. In such cases, manual analysis is necessary to identify patterns or specific JSON payloads unique to IoT services. We have achieved results from Joylink and MiJia, as shown in Table II.

D) Feature labeling:

Positive and negative features are identified and labeled based on their relevance to specific IoT platforms. Positive features indicate a positive relation to a certain IoT platform, while negative features indicate a negative relation (e.g., advertisement services' label)

E) Classification:

1) Sufficient Positive Condition: If positive features exist, the traffic is marked as belonging to a certain IoT cloud platform
.
2) Sufficient Negative Condition: If negative features exist, the traffic is marked as not belonging to a certain IoT cloud platform.

3) Necessary Positive Condition: If positive features are absent, the traffic is marked as not belonging to a certain IoT cloud platform.

4) Necessary Negative Condition: If negative features are absent, the traffic is classified as belonging to a certain IoT cloud platform. Necessary conditions are prioritized to minimize conflicts

IV.IMPLEMENTATION OF IOTPF

The workflow of IoTPF as shown in fig.3.
It involves the following steps:

1) Packet analysis and feature extraction.
2) Feature labeling as positive or negative.
3) Application of classification conditions to determine platform affiliation.
4) Translation of features into scripts, often using regular expressions for efficient matching.
5) Integration of these scripts into IoTPF for automated traffic classification.

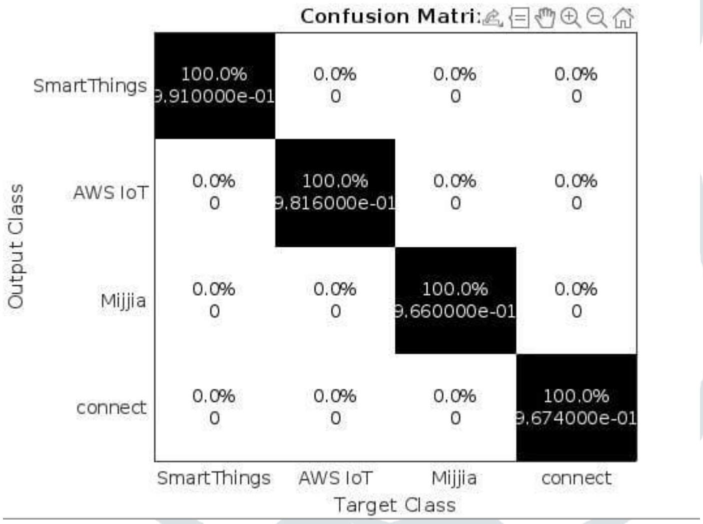
V. RESULTS

It's impressive to note the high accuracy rate of 98.9% achieved by IoTPF in distinguishing IoT platform traffic. However, it's also essential to understand the reasons behind the inaccurate results to further improve the tool's performance. The main factors contributing to these inaccuracies, as outlined in your description, include

1)Unclear Purpose of Public Service Traffic: Some public service traffic might not have a clear purpose, making it challenging to distinguish from IoT traffic during the preprocessing stage. This ambiguity could lead to misclassification.

2)Conflicts Among Platforms Sharing Web Servers: Platforms such as MiJia, Alink, and Ezviz, which utilize the same Web server, may exhibit conflicts on certain data points with limited features. Consequently, IoTPF might confuse data from these platforms.

3)Ambiguity in Traffic Originating from Specific Servers: The traffic originating from servers like "openresty" used by Samsung Connect could be confused with non-IoT traffic due to similarities or lack of distinguishing features.



	SmartThings	AWS IoT	Mijjia	connect
SmartThings	100.0% 9.910000e-01	0.0% 0	0.0% 0	0.0% 0
AWS IoT	0.0% 0	100.0% 9.816000e-01	0.0% 0	0.0% 0
Mijjia	0.0% 0	0.0% 0	100.0% 9.660000e-01	0.0% 0
connect	0.0% 0	0.0% 0	0.0% 0	100.0% 9.674000e-01

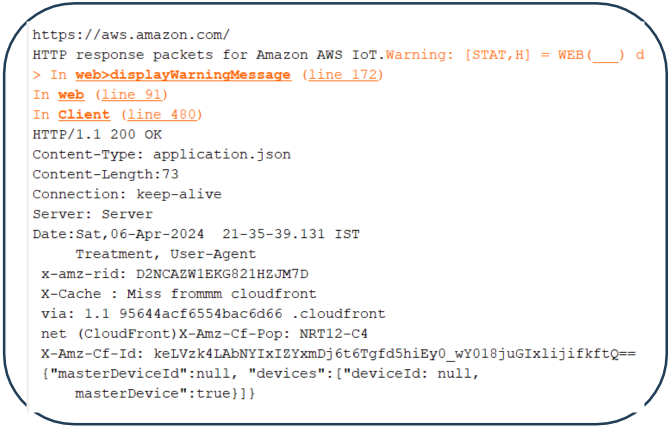
Fig.3. Confusion matrix of the result of marking experiments

Conclusion:

The efficiency and accuracy of the fingerprinting model by incorporating machine learning techniques such as deep learning algorithms. This could potentially allow for more robust identification of IoT platforms and services, as well as better differentiation between IoT and non-IoT traffic. Furthermore, exploring the integration of blockchain technology into our security framework could provide added layers of protection, particularly in terms of ensuring the integrity and immutability of data exchanged within IoT networks. Smart contracts deployed on a blockchain could enforce access control policies and facilitate secure transactions between IoT devices and platforms. Additionally, as the IoT ecosystem continues to expand and diversify, we anticipate the emergence of new types of devices and communication protocols.

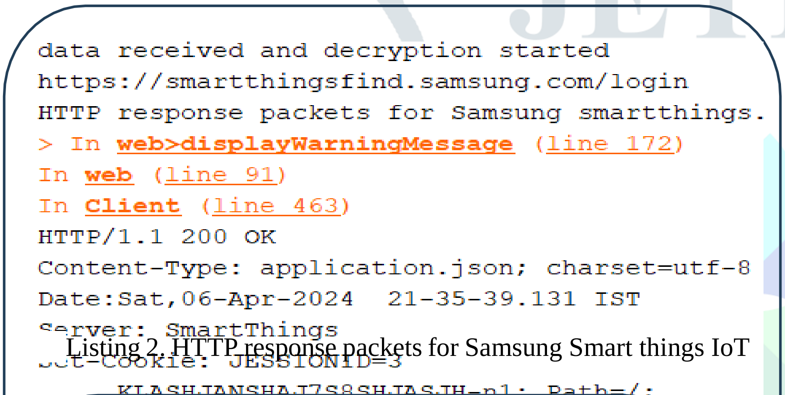
APPENDIX A TYPICAL PACKETS FOR FINGERPRINTING

See the Listings 1–3



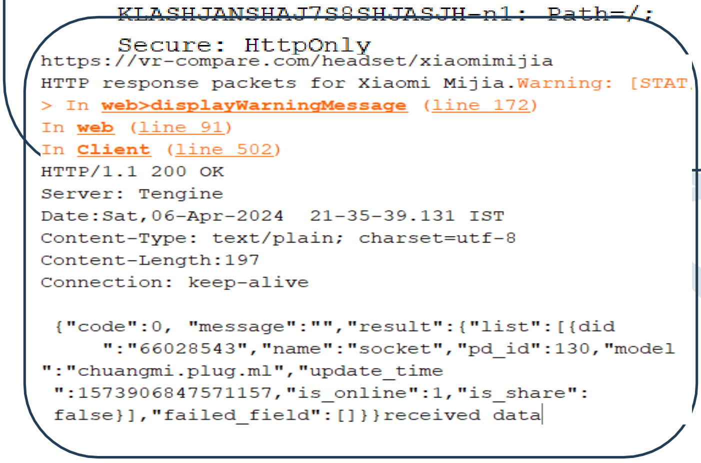
```
https://aws.amazon.com/
HTTP response packets for Amazon AWS IoT.Warning: [STAT,H] = WEB(___) d
> In web>displayWarningMessage (line 172)
In web (line 91)
In Client (line 480)
HTTP/1.1 200 OK
Content-Type: application.json
Content-Length:73
Connection: keep-alive
Server: Server
Date:Sat,06-Apr-2024  21-35-39.131 IST
     Treatment, User-Agent
 x-amz-rid: D2NCAZW1EKG821HZJM7D
 X-Cache : Miss frommm cloudfront
 via: 1.1 95644acf6554bac6d66 .cloudfront
 net (CloudFront)X-Amz-Cf-Pop: NRT12-C4
 X-Amz-Cf-Id: keLVzk4LAbNYIxIZYxmDj6t6Tgfd5hiEy0_wY018juGIxlijifkftQ==
 {"masterDeviceId":null, "devices":["deviceId: null,
     masterDevice":true}]}
```

Listing 1. HTTP response packets for Amazon Aws IoT



```
data received and decryption started
https://smartthingsfind.samsung.com/login
HTTP response packets for Samsung smartthings.
> In web>displayWarningMessage (line 172)
In web (line 91)
In Client (line 463)
HTTP/1.1 200 OK
Content-Type: application.json; charset=utf-8
Date:Sat,06-Apr-2024  21-35-39.131 IST
Server: SmartThings
Set-Cookie: JESSIONID=3
     KLASHJANSHAJ7S8SHJASJH-n1: Path=/;
     Secure: HttpOnly
```

Listing 2. HTTP response packets for Samsung Smart things IoT



```
https://vr-compare.com/headset/xiaomimijia
HTTP response packets for Xiaomi Mijia.Warning: [STAT
> In web>displayWarningMessage (line 172)
In web (line 91)
In Client (line 502)
HTTP/1.1 200 OK
Server: Tengine
Date:Sat,06-Apr-2024  21-35-39.131 IST
Content-Type: text/plain; charset=utf-8
Content-Length:197
Connection: keep-alive

 {"code":0, "message":"","result":{"list":[{did
     ":"66028543","name":"socket","pd_id":130,"model
":"chuangmi.plug.ml","update_time
 ":1573906847571157,"is_online":1,"is_share":
 false}],"failed_field":[]}}received data
```

Listing 3. HTTP response packets for XiaomiMijia IoT