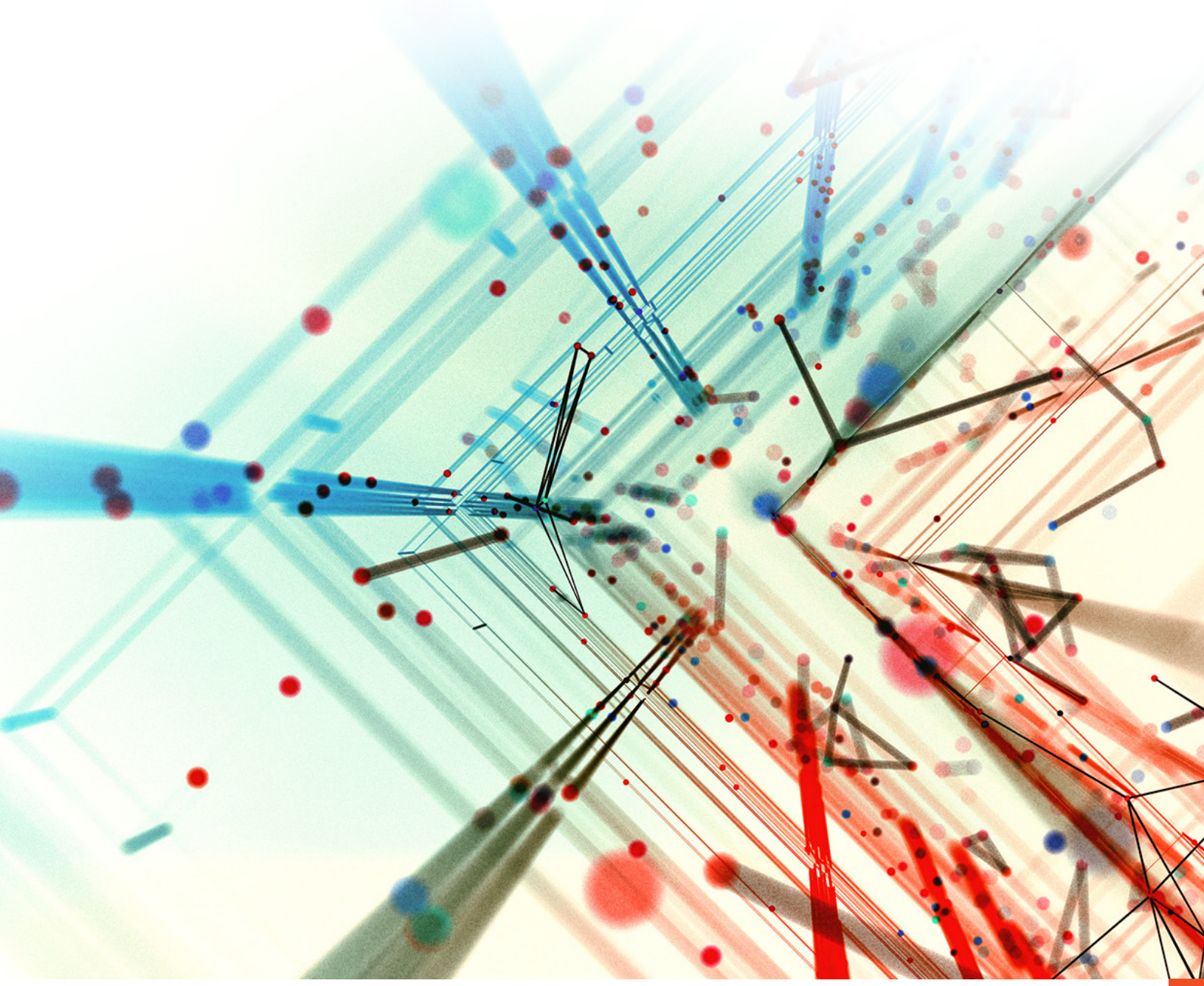
PALO ALTO NETWORKS | IoT

Impacts of Cyberattacks on IoT Devices



paloalto NETWORKS

Table of Contents

Introduction

The early days of the internet age were about connecting people, whereas today the focus is on connecting "things." In the broadest sense, the internet of things (IoT) is a vast matrix of numerous nonstandard devices and endpoints connected wirelessly over the internet. IoT device connections are enabled by advanced wireless cellular technologies, including 3G, 4G, 5G, and low-power wide-area (LPWA) cellular technologies, such as NB-IoT and LTE-M.

Connected IoT devices are capable of generating and transmitting real-time data from the nearest to the farthest edges of the network. The data is then monitored, managed, and analyzed to achieve desired business outcomes. According to a recent forecast by IDC, there will be 41.6 billion connected IoT devices, or "things," generating 79.4 zettabytes (ZB) of data in 2025.[1]

By leveraging analytics derived from large volumes of IoT device-generated data, organizations gain insights germane to the performance of their business systems, operational processes, and customer experiences. IoT-generated data has the unique potential of opening up market opportunities for one-of-a-kind products and services not previously thought possible. Due to its compelling value proposition as a major driver of business transformation, IoT adoption is growing rapidly across numerous industry verticals worldwide. Key findings from a recent report by Microsoft suggest that almost 88% of business decision makers feel the adoption of IoT-based technologies is becoming critical for business success.[2] However, despite the business transformation benefits IoT has to offer, organizations are confronted with barriers when it comes to adoption—97% of respondents to Microsoft's IoT Signals survey cited security concerns as a major challenge when implementing IoT.[3]

Modern IoT Attacks

All IoT devices are vulnerable to being weaponized with botnets for the purpose of carrying out distributed denial-of-service (DDoS) attacks. In 2016, the Mirai botnet targeted more than 600,000 IoT devices, such as routers and cameras, to launch a massive DDoS attack that reached 620 Gbits at its peak. Since that infamous incident, new malware families have come into existence to launch other types of aggravated attacks. Take Xbash for example, which has botnet, ransomware, cryptomining, and self-propagation capabilities. Data-destructive by nature, Xbash spreads by attacking weak passwords and unpatched vulnerabilities in Linux and Microsoft Windows® servers. Another example, a new variant of the Muhstik botnet, self-installs and infects Linux servers and IoT devices with its wormlike self-propagating capability to wreak havoc with cryptomining and DDoS attacks. To understand the reasons behind these attacks and their impact on IoT devices, let's move on to discuss IoT device attack surfaces.

The attack surfaces of IoT devices are broadly categorized into three main groups: hardware interfaces, communication channels, and applications/services. The first group, the physical **hardware interface** of the device, is the most obvious attack surface. Attackers get access to the shell or outermost layer of the operating system kernel of the device, modify the firmware, and go as far as to embed a backdoor through the hardware interface to bypass normal authentication mechanisms.

IoT devices connect and communicate through two types of channels: either short-range channels that include BLE, Zigbee/Z-Wave, and Wi-Fi or the long-range cellular network channel. These **communication channels** represent the second attack surface. Few IoT devices use secure and superior encrypted communications during initial configuration, thereby making it simple for attackers to make the IoT device vulnerable to IP spoofing where a hacker illicitly impersonates another IoT device in the network to spread malware and launch a man-in-the-middle attack, replay attack, or a denial-of-service (DoS) attack.

Applications/services are essential to IoT devices and represent the third attack surface. The administrative interface, the web API, the cloud server, system functionality components, and services delivered from the applications are all susceptible to attacks. Most modern IoT attacks are unleashed by compromising this attack surface because of its scalability, accessibility, and proximity to the edge of the internet. Even if the services are deployed on an enterprise's intranet, attackers can still infiltrate it through a vulnerable edge router. For example, the Xbash botnet has the ability to infiltrate the enterprise intranet to scan and attack multiple services, such as the Telnet application protocol and File Transfer Protocol (FTP).

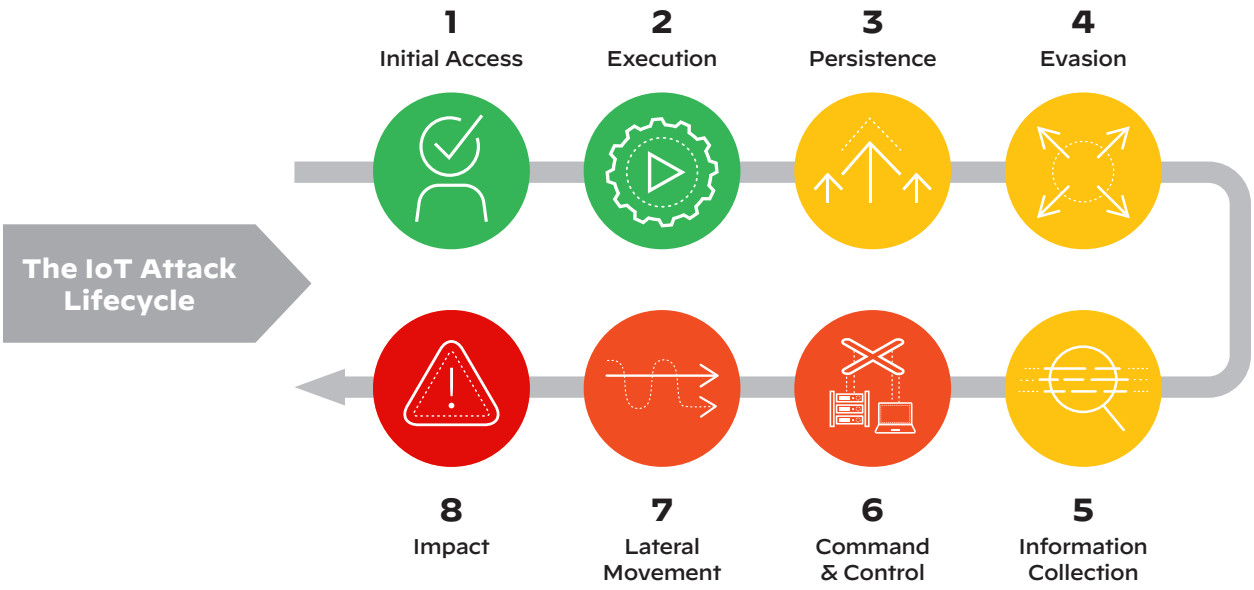


Figure 1: Stages of the IoT attack lifecycle

The IoT Attack Lifecycle

Now that we've reviewed the three main types of IoT attack surfaces, let's discuss how the IoT attack lifecycle works. The lifecycle comprises eight stages.

1. Initial Access

As the name suggests, this is the first stage in the IoT attack lifecycle. At this stage, the attack leverages the network scanning method to first locate IP addresses of vulnerable devices using fast port-scanning tools, such as ZMap or Masscan, to scan the internet.

2. Execution

At this stage, the attack executes payloads or commands in the vulnerable device. In order to do this, it either gets access to the shell of the device's operating system directly or injects the device with commands. To get direct access, brute force is used to attack weak or default passwords for services such as Telnet or SSH. To add to this, shell commands are then injected by exploiting the remote code execution (RCE) vulnerability or command injection vulnerability. When shell commands are executed, a general pattern follows in that a malicious executable file (such as the ELF binary or the shell script) gets downloaded, the executable permission to the payload file gets assigned, and the payload gets executed.

3. Persistence

In the third stage, the executed malware payload shows persistence on the device. The malware can show persistence by killing the watchdog process to avoid the system rebooting; insert itself in scheduled cron jobs, system booting initial jobs, and system daemons; and even create new accounts. The shell is, at times, left open as a redundant access channel in order to establish redundant access in the future.

4. Evasion

The use of evasion techniques ensures the attack is not discovered or detected. By being evasive, the attack can clear the system logs and the BASH command history, hide the payload file in the system folder with a masquerading filename, and even delete the original payload file. Advanced malware, such as the Xbash family for its ability to install host-based security monitoring tools. Lastly, to avoid detection, the attack also adopts anti-VM and anti-debugging techniques to detect the presence of automatic malware analysis systems, such as debuggers and virtual machines. Examples of these include the Tsunami variant and the Torii botnet.

5. Collection of Information

At this stage, device information and sensitive files, such as the private key and the cryptocurrency wallet, are collected. Take the VPNFilter malware as an example. As an advanced persistent threat (APT) infect- ing a number of IoT network routers and storage devices, VPNFilter steals sensitive data from the network traffic in compromised routers.

6. Command and Control

Next, the malicious payload also receives commands from the command-and-control (C2) server. For different C2 commands, the payload continues to launch different attack activities like TCP flooding, UDP flooding, and infiltration of additional devices. For C2 channels, HTTP, IRC, P2P, and other such protocols are used.

7. Lateral Movement

The lateral movement in IoT attacks is mainly to continue infecting a large number of new devices in the local network. For example, an edge router first gets infected and then continues to infect all IoT devices that are connected to it.

8. Impact

Malicious activities launched in the IoT device have multiple impacts on the device: encryption of data for a ransom, total wipe out of disk and data, and abuse for cryptomining. The BrickBot family of malware, for example, can "brick" an IoT device by corrupting its storage capability or by completely reconfiguring its kernel parameters.

While attack surfaces, threat vectors, and vulnerabilities in IoT devices are being widely researched, we haven't seen enough research on the impacts of successful cyberattacks on IoT devices from the perspective of the customer. A Project rIoT paper from UC Berkeley is the first one of its kind to have discussed this topic. However, the paper focuses more on network consumption and evaluates the impact by the cost of bandwidth. Research on BlackIoT[4] from USENIX demonstrated that an IoT botnet of high-wattage devices gives adversaries a unique ability to launch large-scale coordinated attacks on the power grid.

At Unit 42, the global threat intelligence team for Palo Alto Networks, we took a closer look at the impact of cyberattacks on IoT devices, including the overall device performance, device usability, and the services offered by the IoT devices.

Cyberattack Scenarios on IoT Devices and Impacts

Network Scanning Impact on Network Services

One of the most important attack stages in an IoT attack lifecycle is the initial access "network scanning" stage when an attacker starts probing available open ports with scanner IPs in compromised IoT devices. It is possible to launch the attack remotely by exploiting a remote code execution vulnerability, typically on internet-facing service port 37215.

It is evident that the network scanning in compromised IoT devices costs network bandwidth resources. So, an experiment was designed to explore the impacts of network scanning on application or network services in compromised IoT devices.

Experiment Environment and Test Methods

In most IoT attack cases, attackers have used many types of network scanner tools or programs, such as Nmap, ZMap, and Masscan. Here, we choose the Masscan tool, which is a mass IP port scanner. Masscan can scan the entire internet in fewer than six minutes, transmitting 10 million packets per second from a single machine.

We used an ASUS RT-AC87U router as the test device. To bypass the challenge of getting a shell access of ASUS RT-AC87U, we directly flashed a new ROM of Asuswrt-Merlin. We monitored the real-time bandwidth usage on the management page of Asuswrt-Merlin. We also did the comparison analysis on the service response time and device stability status under both the non-scanning and scanning environments.

CPU, Memory, and Bandwidth Usage

We tested different rates of Masscan for the port scanning, as shown in Table 1. We chose the 1 million packets per second rate in the following experiments.

Table 1: Bandwidth, CPU, and Memory Usage at Different Scanning Rates

Scanning Rate (packets/sec)	Bandwidth (KB/sec)	CPU (%)	Memory (%)
1000	55	0.7	26
2500	137	1.6	26
5000	268	3.8	26
...	...	...	...
1M	5140	51.5	26

Table 2: Response Delay of Management Plane Services		
Service Functionality	**Average Time During Normal Conditions (sec)**	**Average Time During Network Scan (sec)**
Log in on management page "index.html"	0.458	0.849 (185%)
Refresh the system information	0.0408	0.0481 (-)
Update clients	0.0127	0.0138 (-)
Ping	4.301	14.602 (340%)
Netstat	0.583	1.301 (223%)

Table 3: Response Delay of Dataplane Services		
Service Functionality	**Average Time During Normal Conditions (sec)**	**Average Time During Network Scan (1M packets/sec)**
Save router settings	0.196	0.273 (139%)
Get system logs	0.125	0.150 (120%)

Service Response Time Results

Management Plane Service

- **Log in on management page "index.html":** The device administrator can log in directly on the webpage "index.html" by typing the username and password.
- **Refresh the system information:** The device administrator can click the refresh button to obtain the latest health and status information of the router.
- **Update clients:** The device administrator can check the current connected clients.
- **Ping:** This utility tool is for the device administrator to check the network connection.
- **Netstat:** This utility tool is for the device administrator to know the current network status.

Data Plane Service

- **Save router settings**—used to download all the configuration information from the router.
- **Get system logs**—used to download all the system log files.

Service Availability

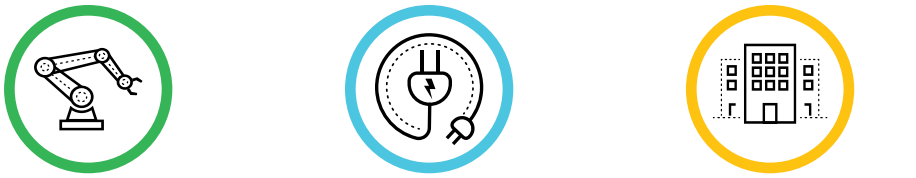
Service availability was not impacted, but service delay was experienced for the above services.

Conclusions

Network scanning had a subtle impact on most services, with the exception of those relying on the network functionality. For example, the "ping" and "netstat" management plane services are affected heavily. The "ping" service response was delayed since the ICMP packet was sent slowly due to the limited available bandwidth. The "netstat" service had to traverse each socket, but since the sockets were created and destroyed by the Masscan tool, this delayed the service response for "netstat" services.

Impacts

We noted that the post-exploitation behavior of the network scanning can cause network congestion, mainly affecting services that rely on the network. If the network delay exceeds a certain threshold, services such as Ping can produce inaccurate results. Other noted impacts include slow response to customer logins to the management webpage and sometimes browser timeout.



Smart Manufacturing
Machinery control
Factory automation
Industrial IoT

Critical Infrastructure
Smart metering for utilities (electric, water, gas)
Smart Grid

Smart Cities
Smart street lighting
Smart parking
Smart waste management

Figure 2: Sample industries with C-IoT applications

C-IoT Device Battery Exhaustion

Background and Attack Vectors

Cellular IoT (C-IoT) devices operate by leveraging existing cellular networks to connect to the internet. In the network layer, C-IoT relies on the low-power wide area network (LPWAN) technology, a category of wireless communication networks capable of providing improved long-range coverage while demanding low power usage. C-IoT is becoming an increasingly dominant LPWAN connectivity option to support low-cost, low-power sensors and IoT devices, enabling massive device connectivity. Cellular IoT comprises complementary technologies, LTE-M, NB-IoT, and EC-GSM-IoT, optimized and ideally suited for the needs of low-cost and low-power IoT applications.

Unlocking the Potential of C-IoT Connectivity

C-IoT has its applications in a variety of industries, such as smart manufacturing, farming, and smart cities, and plays a huge role in industry digitization. For example, IoT adoption is changing the paradigm of farming and the way things are monitored, especially tracking livestock, irrigation, and monitoring climate using connected greenhouse sensors.

IoT devices can become a pretty attractive target for ransomware. Vulnerable IoT sensors on a factory floor that are infected by ransomware could bring the production to a complete stop. Besides that, these attacks could result in expensive overhaul, involving maintenance costs running into the hundreds of thousands of dollars. Needless to say, the impact of attacks to the connected medical devices and autonomous vehicles could be devastating.

C-IoT sensors serving applications such as power-grid monitoring in smart farming, smoke detection in environment monitoring setup, and even smart meters serving the utility segment are often battery-constrained and rely on batteries as a power source. This requirement makes ransomware much more feasible, given the high cost of replacing all batteries in a geographically distributed remote locations setup.

Attackers have two main attack channels to achieve their purpose; in this case, it is battery exhaustion. If these C-IoT devices are connected to a vulnerable gateway, such as the Sierra LTE gateway, it becomes an easy target for the attackers. In situations where these C-IoT devices need to establish communication with the backend servers for control or updates, these remote servers can be targets for attack as well. Through these channels, attacks can push remote commands, push firmware OTA update, and modify task configurations to control the C-IoT devices.

Experiment Environment and Test Methods

In order to simulate potential attacks in a real-world cellular IoT product, we worked on an LTE CAT-M1-based environment sensor prototype, which monitors and updates temperature and humidity data readings to the cloud via cellular network. The environment setup included:

- Development board: Pycom's GPy board
- Environment sensor: Pycom's Pysense expansion board
- IoT SIM card: Hologram LTE CAT-M1/CAT-NB1 sim card
- Network provider: AT&T
- Battery: PKCELL Lithium Ion Polymer Battery, 3.7v 150mAh, PKCELL LP402025, purchased from Adafruit
- Energy consumption analysis tool: Otii Arc made by Qoitech
- IoT cloud: Pycom's Pybytes

The sensor was programmed in a way such that it wakes up at regular time intervals (the experiment is set up to awake once every 24 hours), to connect to the LTE network, send the temperature and humidity data readings to the cloud, and then go back to sleep again. We chose a low-volume battery for the experiment. Tests were conducted to evaluate the battery usage at different stages of the data upload cycle: device boot up, connecting to LTE, sending the data over LTE, disconnecting from LTE, and entering the deep sleep mode. The battery consumption is evaluated through a graph generated by the Otii Arc.

Test Results

Figure 3 shows the battery consumption by the non-malicious sensor program (programmed to sleep at various steps for better perception). We can note that the LTE connect, data upload procedures and LTE disconnect stages consume more battery power than other procedures.

We further developed a malicious sensor program to include only the LTE connect and data upload procedures. No sleep or deep sleep was introduced into the malicious program. Figure 4 depicts the battery consumption graph by the malicious program.

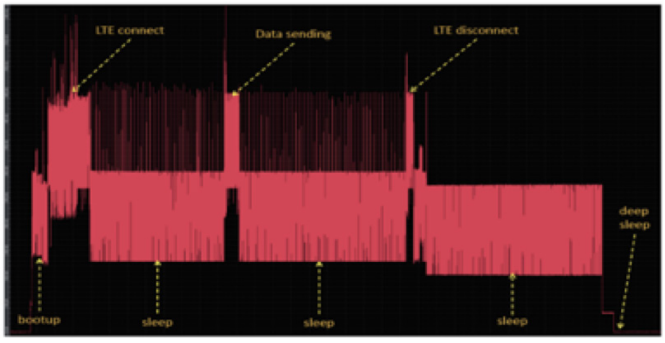


Figure 3: Battery usage in a cycle by the original sensor program

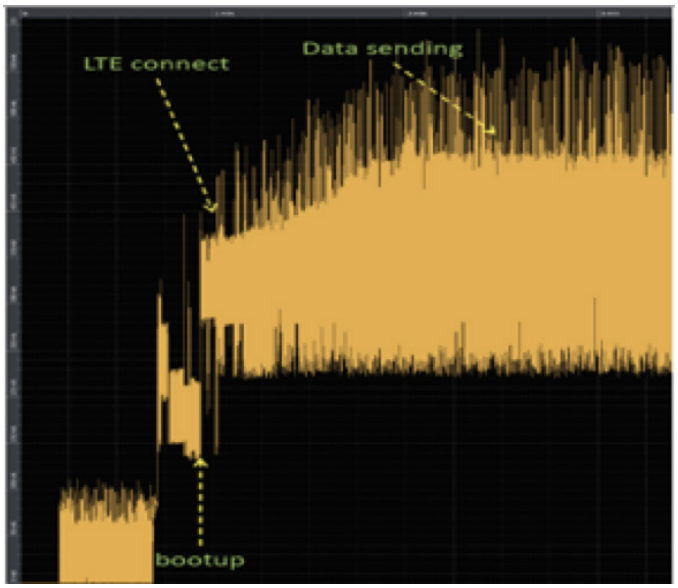


Figure 4: Battery usage by the malicious sensor program

Conclusions

We conclude that IoT devices that are compromised in an attack could result in battery drain, causing the C-IoT devices to power off and render the IoT services offline.

Impact

If the batteries of all C-IoT devices are drained quickly, this almost shuts down the C-IoT application and can cause accidents. For example, the farm could be unprotected and the smoke detection system out of commission. Also, the C-IoT owner would need to spend significant time and money to replace the batteries with new ones and potentially even pay more for the ransom.

Summary

Cyberattacks on IoT devices impact the overall device performance, device usability, and services offered by the IoT devices.

Negative Effect on Device Performance

The impact of modern IoT malware and attacks, such as botnet scanning and propagation, is that they could drain the CPU and memory, resulting in more than a 90% performance reduction, impacting the availability of legitimate services and average life expectancy of the devices.

Reduced Battery Life on Cellular IoT Devices

Certain novel attacks can drain device batteries in cellular IoT significantly in a matter of hours or minutes, which would otherwise sustain a 10-year battery, supporting normal data rates of up to 150 Mbps in normal conditions. Attacks of this nature could result in ransomware or DoS attacks that would likely incur a significant financial cost to mitigate.

1 Carrie MacGillivray and David Reinsel, "Worldwide Global DataSphere IoT Device and Data Forecast, 2019–2023," IDC, May 2019, https://www.idc.com/getdoc.jsp?containerId=US45066919.

2 "IoT Signals," Microsoft, last accessed October 9, 2019, https://azure.microsoft.com/en-us/iot/signals/.

3 Ibid.

4 Saleh Soltan, Prateek Mittal, and H. Vincent Poor, "BlackIoT: IoT Botnet of High Wattage Devices Can Disrupt the Power Grid," USENIX, last accessed October 9, 2019, https://www.usenix.org/conference/usenixsecurity18/presentation/soltan.

paloalto NETWORKS

3000 Tannery Way
Santa Clara, CA 95054

Main: +1.408.753.4000
Sales: +1.866.320.4788
Support: +1.866.898.9087

www.paloaltonetworks.com