Network Intrusion and Security

Hieu Do, Brian Kao, Theodore Kim, Nick Nguyen, and Weiwen Ying

1. Problem Statement

With the recent surge of consumer demand for "smart technologies," common household items such as light bulbs and refrigerators are now being networked and exposed to the public internet to provide extra functionality and connectivity to computers and smartphone **[1]**. This Internet-of-Things (IoT) technology presents an opportunity for increased consumer convenience, but also may serve as a vulnerability in previously secured networks. Still experimenting in a nascent field, IoT developers have yet to develop a thorough security model that can prevent intrusion and unauthorized use of the device and its network. As the IoT ecosystem grows, so do the the number of unsecured access points to the network. The 2016-2017 Mirai DDoS attack of major networks like Twitter and PlayStation, or the 2017 hack of a North American casino system through an Internet-connected fish tank are examples of network intrusion via an IoT device vulnerability. Therefore, it is imperative that a solution be determined for how to prevent such intrusions.

2. Solution Definition

2.1. Networked IoT devices

The Internet-of-Things technology is based on a device's ability to interface with one and other over a private network or public internet to allow for remote access and automatic control. Devices that are connected to the IoT, such as an internet enabled fish tank, communicate to either a cloud service or local server running on a PC or other IoT device through a dedicated API. This connection allows other devices to query the server through the same API (through a web browser or dedicated application) to be remotely monitored and control the system.

2.2. IoT devices and Network Vulnerabilities

As diverse IoT devices are currently available on the market, so too are the vulnerabilities they present for a network **[2]**; however, the most common penetration method involves IoT device's lack of complete mitigation of remote accesses. Attackers can easily compromise the traffic between the device and its server. In fact, 70% of IoT devices lack proper network encryption **[3]**, while others, like many Samsung Smart Refrigerators, inadequately (or lack thereof) verify the identity of the server it is communicating with (SSL) **[4]**. This provides attackers with the opportunity to send unauthorized requests via its API and often gain administrative access to the system through SSH. Once within the shell, the infiltrator has full trusted access within the device's local network.

2.3. Properties of the Solution

Given the unique nature of IoT devices (compared to more traditional network endpoints such as computers, servers, and even smartphones) as single purpose, user independent entities, network security solutions must take into account this platform and exhibit the following properties:

- Given the limited processing capabilities of IoT devices, the solution should achieve **low computational overhead.**
- A network is only as strong as its weakest device, therefore the solution should be **backwards compatible** and applicable to old devices. It should similarly be updatable as IoT

devices often have a lifespan longer than that of the security standards they implement.

- The solution should be **atomic**; an individual compromise or suspicion thereof within the network should not compromise other endpoints.
- The majority of breaches occur from unauthorized accesses to these devices, so the solution should **complete mitigate** all accesses.
- The solution must be **resilient to physical tampering**. Given the mobile and distributed nature of IoT devices, physically acquiring a device for the purpose of penetration is easier compared to traditional computing devices.

2.4. Applications of the Solution

- **Transportation:** IoT devices serve a variety of purposes in the transportation industry from safety (OnStar) to analysis of transit traffic patterns (Waze). A penetration of the private networks on which these systems communicate could violate individuals' privacy (location tracking) or threaten their safety **[5]**.
- **National Defense:** Although IoT devices would be very beneficial to national defense for automation, communication, and data analytics, there are many concerns regarding the security of these devices that are preventing their adoption **[12]**. By improving IoT network security, the potential seizure and penetration of these devices would not threaten the entire network.
- **Smart Homes:** Secure IoT devices encourage users to adopt the technology more readily. Companies like Nest offer a wide range of IoT products for homes **[6]**. In order for users to install these devices in their home, companies need to have a robust solution to protect the security of their users' private networks..
- **Academic Campus Infrastructure:** Many IoT devices can be found in universities, such as vending machines, lighting systems, and smart door locks. Universities host more individuals with the expertise to exploit vulnerabilities in the network, potentially causing disruption to the school's network.

2.5. Threat Model for the System

2.5.1. Default Credentials

Threat: Constrained and single-purpose devices often have a single privilege level (root) security. Control over the system can be granted to a user (often through the public internet as the device is intended to be used remotely) through these keys. If a malicious user gains access remotely, they now have control over the device and unencumbered access to within its private network

Vulnerability: Developers assign simple, default credentials with the implication that they will be changed by the end user later on (which they are not). Therefore, attackers can gain remote access to multiple devices through the use of these known, default credentials.

Risk: With access, the device can request data from other devices within the network or make alterations to the network configuration. Alternatively, the device can perform a DoS attack by flooding another networked device with legitimate requests (like pings). For example, Miller and Valasek demonstrated that the default hotspot password for Jeep automobiles were based on a refactoring of the manufacture date and were able to bruteforce entry onto the network **[7]**.

Solution: Developers should require that the user provide a password that conforms to a strict complexity policy *before* the device becomes active on a network. Alternatively, devices that are compromisable should not have access to other members of the network

2.5.2. Unconditional Trust of the Remote Server

Threat: Many IoT use cloud services to enable indirect, remote communication to and from other devices and / or other internet services. Hackers can intercept the data sent between the device and the server to gain information about the system. Moveover, the attacker can impersonate this remote server in order to send requests to the device in order to manipulate it.

Vulnerability: Many devices do not encrypt its outgoing network traffic, leaving packets readable to anyone with access to any stage of the network link. Furthermore, many devices do not adequately verify the authenticity of the server with which its communicating using standard mechanisms like SSL and HTTPS.

Risk: The attacker may gain meta-information about the network such as the IP / MAC addresses or serial numbers of devices involved in the network transaction (routers, switches, and servers) and keys utilized in calls to the API. All of this information can be used to spoof the identity of the attacker as a legitimate source like the remote cloud server **[8]**. In this capacity, an attacker could instruct the device to compromise the availability of the network through flooding (i.e. Samsung Smart Refrigerators were instructed to repeatedly send emails to users) **[4]**.

Solution: Add measures to encrypt the traffic to make it secure in an unsafe channels using a low overhead algorithm like SHA-3 **[9]** and authenticate the server using SSL.

2.5.3. Out-of-date Firmware

Threat: Attackers gain access to SSH and code execution capabilities in an IoT device via innocuous web queries.

Vulnerability: IoT developers are conscious of the security of the system that they are designing. However, they employ 3rd party libraries with vulnerabilities. When these vulnerabilities are discovered, either through exploitation or penetration testing, updating existing systems with resolutions is often foregone, late, or ignored by the consumer.

Risk: With SSH and code execution capabilities, the attacker has the ability to traverse the network

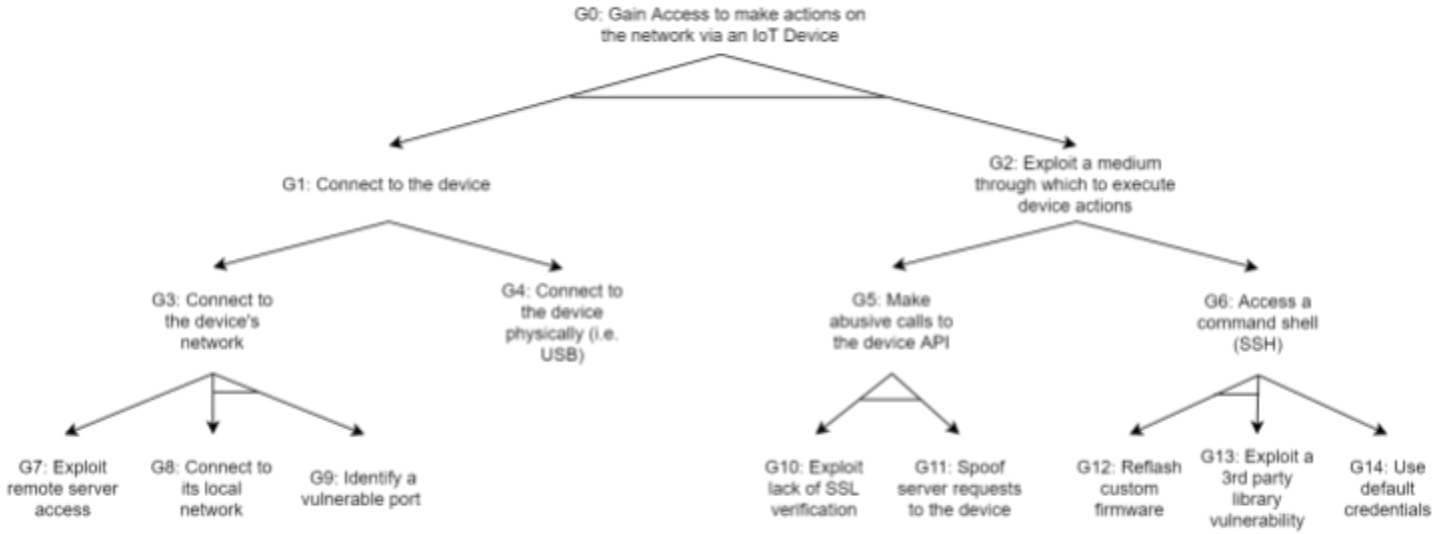

Figure 1: Threat Model - Attack Tree

as a trusted device or install programs on the infected device to enable attacks on the rest of the network (packet sniffing, man-in-the-middle attacks). The "Devil's Ivy" exploit of a 3rd party library on a variety of security cameras allowed a malicious party to take advantage of a known exploit on its connected router, circumvent the corporate VPN and infiltrate its network **[10]**.

Solution: Developers should thoroughly vet the 3rd party software they elect to use, especially those libraries that interface with the device's networking capabilities and be ready to push updates to address vulnerabilities discovered by the community. Consumers should keep their devices up-to-date to protect against discovered vulnerabilities.

2.5.4. Comprehensive Solution Alternatives

There exist a variety of solutions to address the previously outlined threats. Specifically, there are two vectors from which the solution can originate: the developers and the end user.

Developers arguably have the easiest means of resolving these vulnerabilities: building more security oriented systems. One such design alternative is to implement hardware security, such as Microchip's CEC1702 security IC, which prevents writes to a device's firmware and prevent threats such as stack overflow exploits and remote code execution. To ensure network security with low computational overhead, developers can employ hardware encryption such as STMicroelectronic's STM7007. One one hand, these solutions would be easy to implement and are impervious to software exploitation attempts. However, these solutions are nearly impossible to repair or patch in the event of a breach, do not apply to already manufactured products (although the developer could issue a recall, at their expense), and may be physically impossible to implement in small form factor devices.

Consumers also have an opportunity to protect themselves from network exploits through vulnerable IoT devices through careful IT management. This means implementing a network with a carefully designed security policy to defend against IoT specific exploits. A proposal from Cisco describes the steps that an IoT enabled network might implement to ensure its security, including network segmentation, utilization of a VPN to protect possibly unencrypted data, and MAC authentication **[11]**. These security steps can be implemented by an IT team (enterprise consumers) or automatically via hardware such as the Norton Core router. This implementation of user side network protection achieves low computational overhead as it reassigns security functionality to network hardware. It is also applicable to both new and old devices and can protect the rest of the network from a potentially compromised device. On the other hand, implementing such a policy is complex and expensive for the consumer to implement and therefore may be impractical.

The final solution is a cooperative effort between both developers and the product's users to keep devices up-to-date. Aside from wide scale attacks such as the Mirai Botnet attack in October, 2016, many breaches happen in isolation and can be patched quickly and pushed to other users to prevent their exploitation. However, there must be a concerted effort amongst IoT users to stay aware of possible software updates and install them timely. This has the potential to solve most of the aforementioned threats (except in the case of hardware breaches, in which users must be willing to upgrade their units and businesses make this alternative attractive) on a wide variety of devices and networks. However, to expect every user (including the casual luddite) to be aware of these problems is probably unreasonable and therefore this solution would be impractical to employ.

2.6. FURTHER EXPLORATION

While implementing a secure Service Management System is key to the security of a network with many unsecured endpoints (i.e. IoT devices), many consumers do not have access to the IT expertise or resources to employ these practices. Therefore, providing automatic enforcement of prebuilt security policies through a network device, specifically a wireless access point and router, should be further explored as an opportunity for the average user to secure their IoT devices.

3. References and Further Reading

[1]https://www.sciencedirect.com/science/article/pii/B9780128113738000045
[2]https://www.owasp.org/index.php/OWASP_Internet_of_Things_Project#tab=IoT_Attack_Surface_Areas
[3]https://www.networkworld.com/article/3217664/internet-of-things/how-to-improve-iot-security.html
[4]https://www.theregister.co.uk/2015/08/24/smart_fridge_security_fubar/
[5]https://www.wired.com/2015/07/hackers-remotely-kill-jeep-highway/
[6]https://www.forbes.com/sites/andrewweinreich/2017/12/18/the-future-of-the-smart-home-smart-homes-iot-a-century-in-the-making/#63437bc657a
[7]https://ericberthomier.fr/IMG/pdf/remote_car_hacking.pdf
[8]https://securelist.com/iot-hack-how-to-break-a-smart-home-again/84092/
[9]https://www.nist.gov/news-events/news/2012/10/nist-selects-winner-secure-hash-algorithm-sha-3-competition
[10]https://www.wired.com/story/elaborate-hack-shows-damage-iot-bugs-can-do/
[11]https://www.cisco.com/c/en/us/about/security-center/secure-iot-proposed-framework.html
[12]https://fedtechmagazine.com/article/2018/02/future-dods-plan-defend-against-iot-threats