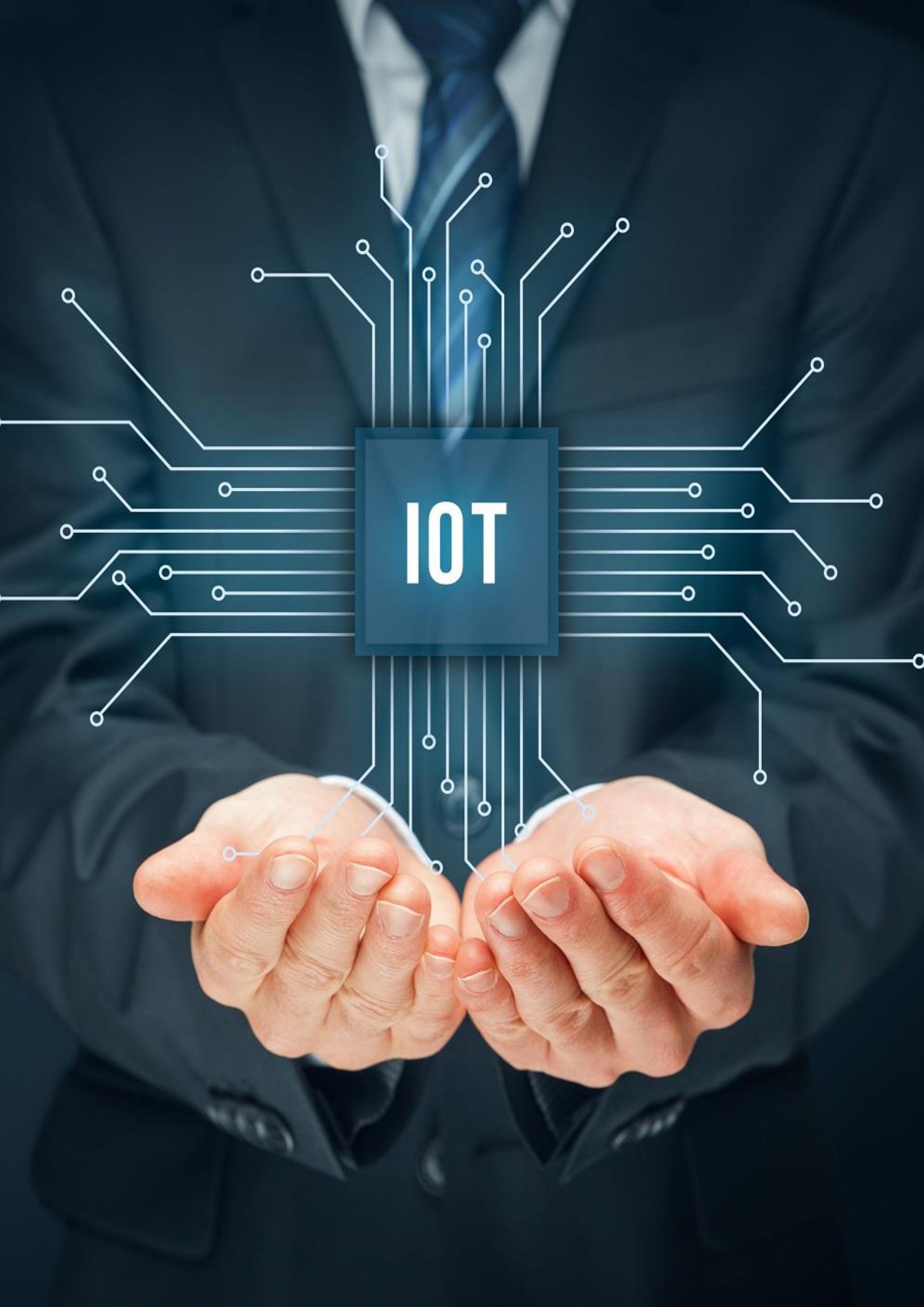

IOT

IoT and Hacking: Baby Monitor Exposures and Vulnerabilities

Baby monitors fulfill an intensely personal use case for IoT. They are usually placed near infants and toddlers, are intended to bring peace of mind to new parents, and are marketed as safety devices. Being Internet accessible, they also help connect distant family members with their newest nieces, nephews, and grandchildren, as well as allow parents to check in on their kids when away from home. They are also largely commodity devices, built from general purpose components, using chipsets, firmware, and software found in many other IoT devices. Video baby monitors make ideal candidates for security exploration; not only are they positioned as safety and security devices (and therefore, should be held to a reasonably high standard for security), but the techniques used in discovering these findings are easily transferable to plenty of other areas of interest.

The Challenge

Baby monitors, being a 'thing' that can be connected to 'internet' makes this case quite an 'Internet of Things'. However, once installed and connected over the internet it cannot be upgraded or have patches installed via the same network securely. Unlike traditional computers IoT devices often lack a reasonable update and upgrade path once the devices leave the manufacturer's warehouse. The absence of a fast, reliable, and safe patch pipeline is a serious and ongoing deployment failure for the IoT.

The presence of devices that are insecure by default, difficult to patch, and impossible to directly monitor by today's standard corporate IT security practices constitutes not only a threat to the IoT device and its data, but also to the network to which it's connected. As the IoT is made up of general purpose computers, attackers may be able to leverage an exposure or vulnerability to gain and maintain persistent access to an IoT device. That device can then be used to pivot to other devices and traditional computers by taking advantage of the unsegmented, fully trusted nature of a typical home network.

Another concern is the raw computing power available to attackers in the form of millions to billions of IoT devices. In total, the teraflops of processing power may be effectively harnessed by malicious actors to launch powerful distributed denial of service (DDoS) attacks against arbitrary Internet targets.

Vulnerabilities

Cleartext Local API: Devices built with commodity components and software often fail to use modern cryptographic standards for LAN-local communications. While it is "only the LAN," there are many passive and active network attacks which can be defeated simply by using common encrypted protocols, such as HTTPS and SSH.

Cleartext Cloud API: Major Internet brands, such as Facebook, Google, Twitter, and other household names are adopting encryption across the board in order to ensure the privacy and authenticity of communications routed over the public (and eaves droppable) Internet. However, services connected with IoT devices often fail to adhere to this increasingly common standard.

Unencrypted Storage: In addition to the cleartext implementations described above, an ideal IoT a recording device such as a video baby monitor should store all recordings in industry standard, encrypted formats, where only authorized users have access to the recorded data.

Backdoor Accounts: As these devices are developed, manufacturers occasionally include either default accounts or service accounts, which are either difficult or impossible to disable under normal usage. Furthermore, these accounts often use default or easily guessable passwords, and tend to share the same unchangeable password, SSH key, or other secret-but-universally-shared token. Finally, these accounts may be protected by a password unique to the device, but the password generating algorithm is easily deduced and the passwords for all devices can be guessed with low attacker effort.

Particular Cases of Exposed Vulnerabilities and Mitigations

Case 1: A particular product from a vendor's site was assessed. The website has a vulnerability by which any authenticated user to the website's service is able to view camera details for any other user, including video recording details, due to a direct object reference vulnerability.

The object ID parameter is eight hexadecimal characters, corresponding with the serial number for the device. This small object ID space enables a trivial enumeration attack, where attackers can quickly brute force the object IDs of all cameras.

Once an attacker is able to view an account's details, broken links provide a filename that is intended to show available "alert" videos that the camera recorded. Using a generic AWS CloudFront endpoint found via sniffing iOS app functionality, this URL can have the harvested filename appended and data accessed from the account. This effectively allows anyone to view videos that were created from that camera stored on the website's service, until those videos are deleted, without any further authentication.

Mitigation: Today, this attack is more difficult without prior knowledge of the camera’s serial number, as all logins are disabled on the website. Attackers must, therefore, acquire specific object IDs by other means, such as sniffing local network traffic. In order to avoid local network traffic cleartext exposure, customers should inquire with the vendor about a firmware update, or cease using the device

Case 2: The device ships with hard coded credentials, accessible from a telnet login prompt and a UART interface, which grants access to the underlying operating system.

Mitigations: In order to disable these credentials, customers should inquire with the vendor about a firmware update. UART access can be limited by not allowing untrusted parties physical access to the device. A vendor-provided patch should disable local administrative logins and in the meantime, end-users should secure the device’s housing with tamper-evident labels.

Case 3: The device ships with hard coded and statically generated credentials which can grant access to both the local web server and operating system. The operating system “admin” and “mg3500” account passwords are present due to the stock firmware used by this camera, which is used by other cameras on the market today. In addition, while the telnet service may be disabled by default on the most recent firmware, it can be re-enabled via an issue

Mitigations: In order to disable the hard-coded credentials, customers should inquire with the vendor about a firmware update. UART access can be limited by not allowing untrusted parties physical access to the device. A vendor-provided patch should disable local administrative logins, and in the meantime, end-users should secure the device’s housing with tamper-evident labels. In order to avoid the XSS and cleartext streaming issues with Philips’ cloud service, customers should avoid using the remote streaming functionality of the device and inquire with the vendor about the status of a cloud service update.