RAPID7

HACKING IoT: A Case Study on Baby Monitor Exposures and Vulnerabilities

#IoTsec

Written by **Mark Stanislav** and **Tod Beardsley** | September 2015*

**Last updated September 29, 2015*

HACKING IoT: A Case Study on Baby Monitor Exposures and Vulnerabilities

Contents

RAPID7

Executive Summary

This is especially relevant today, as employees increasingly blur the lines between home networks and business networks.

The term "Internet of Things" (IoT) is used to describe a galaxy of wildly different devices, from twenty dollar children's toys to airliners that cost hundreds of millions of dollars. While this paper focuses on the consumer end of the IoT spectrum, we believe that the findings can inform how security researchers look at undiscovered vulnerabilities affecting expensive, industrial devices as well.

While Rapid7 is not aware of specific campaigns of mass exploitation of consumer-grade IoT devices, this paper should serve as an advisory on the growing risk that businesses face as their employees accumulate more of these interconnected devices on their home networks. This is especially relevant today, as employees increasingly blur the lines between home networks and business networks through routine telecommuting and data storage on cloud resources shared between both contexts.

Several video baby monitors from a cross-section of manufacturers were subjected to in-depth security testing, and all of the devices under test exhibited several of these common security issues.

This paper focuses specifically on ten new vulnerabilities which were disclosed to the individual vendors, to CERT, and to the public, in accordance with Rapid7's Disclosure Policy[1]. CVE-2015-2880 through CVE-2015-2889 (inclusive) were assigned by CERT. Typically, these newly disclosed vulnerabilities are only effectively mitigated by disabling the device and applying a firmware update when one becomes available, or with updates to centralized vendor cloud services.

The vulnerabilities explored and disclosed in this paper are broken down according to the "reach" of the attack, that is, if the issues are exploitable only with physical access to the device; if they are exploitable via the local network; or if they are exploitable from the Internet.

It is important to stress that most of the vulnerabilities and exposures discussed in this paper are trivial to exploit by a reasonably competent attacker, especially in the context of a focused campaign against company officers or other key business personnel. If those key personnel are operating IoT devices on networks that are routinely exposed to business assets, a compromise on an otherwise relatively low-value target – like the video baby monitors covered in this paper – can quickly provide a path to compromise of the larger, nominally external, organizational network.

Finally, this paper also discusses the insecure-by-default problems inherent in the design of IoT devices, the difficulty for vendors to develop and deliver patches, the difficulties end-users face in learning about, acquiring, and applying patches once developed, and the friction involved in reporting issues to vendors in a way that is beneficial to end-users. Only one vendor cited in this report, Philips N.V., responded with an expected timeline for producing fixes for the issues described.

[1] https://www.rapid7.com/disclosure.jsp

01

THE INTERNET OF THINGS

For our purposes, we can think of a "Thing" with "Internet" as simply any device, regardless of size, use, or form factor, that contains a CPU and memory, runs software, and has a network interface which allows it to communicate to other devices, usually as a client, sometimes as a server. In addition, these Things tend not to resemble traditional computers. They lack a typical keyboard and mouse interface, and they often have a user interface not centered around a monitor or other text-filled screen. Finally, these devices are marketed and treated as if they are single purpose devices, rather than the general purpose computers they actually are.

This last distinction is often the most dangerous one to make when it comes to deploying IoT devices. In his keynote address to the Chaos Computer Club, *Lockdown: the coming war on general-purpose computing*[2], Cory Doctorow makes the case that with today's technology and current computer science thinking, we cannot yet create a computer that is anything other than a general purpose computer. End users may have devices that are nominally prohibited from performing certain actions according to the manufacturer, and those manufacturers sometimes go to great lengths to foil modification efforts. In the end, though, it is not possible to build and sell a computing device that cannot be coerced into rebelling against a manufacturer's intentions.

The classic example of a manufacturer-imposed prohibited action is media playback restrictions based on a digital rights management (DRM) system. The strategies employed for blocking some kinds of media, while allowing others, are proven to be fundamentally flawed, time and time again.

Self-identified hackers and tinkerers have been compromising DRM systems for decades, coercing media data files and media playback devices into a form more useful for the end-user. Such efforts merely require time, materials, and ingenuity, and are based on a foundational realization that there is truly no such thing as a single-purpose computer. Efforts to evade DRM may ultimately be too costly in terms of time and materials, and may require expertise beyond that of the end-user. While such DRM-evading efforts tend to violate local intellectual property laws, they do not violate the principles of computer science or engineering.

Security systems, like DRM, are for controlling access. Users rely on these systems to prevent unauthorized adversaries from viewing, altering, or destroying data on the secured system. Also like DRM, such systems are not foolproof, since again, the barriers to defeating security systems are time, materials, and expertise, and not the fundamental design of the computing platform. Because IoT devices do not normally appear to be, or behave like, the traditional computers we are familiar with, it is easy for the designers and vendors of these systems to forget this general-purpose property. As a result of this oversight, basic precautions to thwart even casual attackers can fail to make it into production.

IoT devices are actually general purpose, networked computers in disguise, running reasonably complex network-capable software. In the field of software engineering, it is generally believed that such complex software is going to ship with exploitable bugs and implementation-based exposures. Add in external components and dependencies, such as cloud-based controllers and programming interfaces, the surrounding network, and other externalities, and it is clear that vulnerabilities and exposures are all but guaranteed.

[2] https://boingboing.net/2012/01/10/lockdown.html

02
NO EASY FIXES

With traditional computers, we understand that access controls are required in order to satisfy basic security requirements. We also know that these controls will contain bugs, or may simply be rendered obsolete in the face of a novel new attack. Such circumstances are inevitable, and require a configuration change, a patch, or an entirely new design.

IoT devices, unlike traditional computers, often lack a reasonable update and upgrade path once the devices leave the manufacturer's warehouse. Despite the fact that the network is what makes the Internet of Things so interesting and useful, that network is rarely, if ever, used to deliver patches in a safe and reasonably secure way.

The absence of a fast, reliable, and safe patch pipeline is a serious and ongoing deployment failure for the IoT. A sub-one hundred dollar video baby monitor, a five hundred dollar smart phone, a thirty-five thousand dollar connected car, and a four hundred million dollar jet airliner are all difficult to patch, even when vulnerabilities are identified, known, and a fix is in hand. This situation is due to a confluence of factors, ranging from the design of these devices, through the regulatory environment (or lack thereof) in which these components and devices exist. Today, a commonly accepted (or truly acceptable) way to effect a rapid rollout of patches simply does not exist.

Unpatchable devices are coming online at an unprecedented rate, and represent a tsunami of unsecurable-after-the-fact devices. According to a 2014 Gartner report[3], the IoT space will be crowded with over 25 billion devices in five years, by 2020. The devices being built and shipped today are establishing the status quo of how these Things will be designed, assembled, commoditized, and supported, so we must take the opportunity, now, to both learn the details of the supply chain that goes into producing and shipping IoT devices, the vulnerabilities and exposures most common to these computers in disguise, and how we can work across the entire manufacturing space to avoid an Internet-wide disaster caused by the presence of these devices on the nervous system of Planet Earth.

Compounding these patching problems is the fact that the use of commodity, third-party hardware, software, and cloud-based resources is prevalent in the IoT industry. While reusing off-the-shelf technologies is critical in keeping costs of production low, it introduces an ambiguity of ownership for developing and deploying patches and other upgrades.

If a vulnerability's root cause is traced to a third-party software library, for example, the more correct fix would be to patch that library. However, this decision can lead to a "pass the buck" mentality for the vendors involved in the supply chain, ultimately delaying effective patching for the particular device in which the vulnerability was first discovered.

This patchwork of common components leads to confusing amalgamations of interdependencies, and can leave end-users exposed while the details of remediating vulnerabilities are worked out between vendors.

[3] https://www.gartner.com/newsroom/id/2905717

03

WHY BABY MONITORS?

The research presented focuses on the security of retail video baby monitors for a number of reasons. 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. By 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. Other products of direct interest to commercial and industrial consumers and security researchers (commercial security systems, home automation systems, on-premise climate control systems) share many of the insecure design and deployment issues found in video baby monitors.

04

WHAT IS THE BUSINESS IMPACT?

While video baby monitors are vastly more commonplace in a home environment and uncommon in an office environment, office environments and home environments are, increasingly, literally the same environment.

The percentage of employees and contractors who are working from home on at least a part time basis continues to rise across every modern economy. New parents are traditionally at the core of this trend, though it is increasingly common across all genders, ages, and family statuses[4]. These employees are, as a matter of necessity, connecting to their workplace virtually, either through VPN connections or through the use of cloud services shared by colleagues.

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.

Today, employees' home networks are rarely, if ever, "in scope" for organizational penetration testing exercises, nor are they subject to centralized vulnerability scanners.

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.

Given the lack of home network and on-board monitoring, remediating such attacks may prove extremely difficult once underway, and short-term solutions will tend to deny service to large chunks of residential network space. This, in turn, can knock sizable percentages of the aforementioned stay-at-home workforce offline, with little recourse for employers not prepared to offer alternative workplace accommodations.

[4] http://www.nytimes.com/2014/03/08/your-money/when-working-in-your-pajamas-is-more-productive.html

05

COMMON VULNERABILITIES AND EXPOSURES FOR IoT DEVICES

The items below describe the common vulnerabilities and exposures for IoT devices. Not all IoT devices suffer from all of these software, firmware, and hardware issues, but it is rare to find an IoT device that doesn't exhibit at least one critical failing. Of the devices under test, all exhibited several common vulnerabilities and exposures.

KNOWN VULNERABILITIES	OLD VULNERABILITIES THAT SHIP WITH NEW DEVICES
Cleartext Local API	Local communications are not encrypted
Cleartext Cloud API	Remote communications are not encrypted
Unencrypted Storage	Data collected is stored on disk in the clear
Remote Shell Access	A command-line interface is available on a network port
Backdoor Accounts	Local accounts have easily guessed passwords
UART Access	Physically local attackers can alter the device

Table 1, Common Vulnerabilities and Exposures

Known Vulnerabilities

Brand-name manufacturers of IoT devices tend to implement much of the technology used by their products as embedded systems subcomponents, sourced from third party suppliers. The upstream vendors of these sub-components tend to run extremely large operations, producing millions of units in a given year, and any change in this supply chain is both time consuming and expensive. Due to the nature of this time-lagged supply chain, individual software components may be months to years old before being assembled into the final product, bringing old and commonly known software vulnerabilities along with them.

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 eavesdroppable) 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 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.

Remote Shell Access

IoT devices often ship with default or otherwise unconfigured portable operating systems, and are often host to a Linux or other POSIX kernel with a set of stock utilities, such as BusyBox. While these are quite useful for developing and tinkering with hardware, they should not be made available on production systems where shell access is never desired or required.

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.

UART Access

Universal Asynchronous Receiver/Transmitter (UART) interfaces often enable a physically close attacker to access and alter IoT devices in ways that bypass the normal authentication mechanisms via a serial cable connection. In addition, UART interfaces tend to grant root access, far exceeding the permissions of regular users. UART access is both a useful diagnostic tool and an excellent means of "rooting" or "jailbreaking" consumer devices. Such activities on a device specifically made for safety and security can lead to some very sneaky persistent attacks. IoT devices such as these should at least be tamper-evident, and give the owner or investigator some obvious indication that it has been altered, if UART access is intended at all.

Newly Discovered Vulnerabilities and Exposure Summary

This report is primarily focused on newly discovered vulnerabilities, rather than exhaustively detailing the expected and typical vulnerabilities found across the IoT space. Table 2 summarizes the new vulnerabilities discovered and disclosed to the vendors and CERT.

CVE-2015-2886	Remote	R7-2015-11.1	Predictable Information Leak	iBaby M6
CVE-2015-2887	Local Net, Device	R7-2015-11.2	Backdoor Credentials	iBaby M3S
CVE-2015-2882	Local Net, Device	R7-2015-12.1	Backdoor Credentials	Philips In.Sight B120/37
CVE-2015-2883	Remote	R7-2015-12.2	Reflective, Stored XSS	Philips In.Sight B120/37
CVE-2015-2884	Remote	R7-2015-12.3	Direct Browsing	Philips In.Sight B120/37
CVE-2015-2888	Remote	R7-2015-13.1	Authentication Bypass	Summer Baby Zoom Wifi Monitor & Internet Viewing System
CVE-2015-2889	Remote	R7-2015-13.2	Privilege Escalation	Summer Baby Zoom Wifi Monitor & Internet Viewing System
CVE-2015-2885	Local Net, Device	R7-2015-14	Backdoor Credentials	Lens Peek-a-View
CVE-2015-2881	Local Net	R7-2015-15	Backdoor Credentials	Gynoii
CVE-2015-2880	Device	R7-2015-16	Backdoor Credentials	TRENDnet WiFi Baby Cam TV-IP743SIC

Table 2, Newly Identified Vulnerabilities

06

VULNERABILITY REPORTING AND HANDLING

One of the goals of this research is to practice reasonable, coordinated disclosures with vendors of IoT equipment. So, as a matter of course, the vulnerabilities discovered as part of this research were reported in accordance to Rapid7's Vulnerability Disclosure Policy. According to this policy, vendors are contacted once the findings are verified, then after 15 days, CERT is contacted. 45 days after that (60 days after the initial disclosure attempt), the findings are published.

During the course of the vulnerability disclosure process, we saw vendors exhibit the entire range of possible responses. One vendor was impossible to contact, having no domain or any other obvious Internet presence beyond an Amazon store listing. Some vendors did not respond to the reported findings at all. Others responded with concerns about the motives behind the research, and were wondering why they should be alerted or why they should respond at all.

On the exemplary side, one vendor, Philips N.V., had an established protocol for handling incoming product vulnerabilities, which included using a documented PGP key to encrypt communications around this sensitive material. Philips was also able to involve upstream vendors in pursuing solutions to those technologies provided by others. Weaved, a provider of an IoT-in-the-cloud framework for Philips, was especially open with and responsive to the authors of this paper.

The range of responses itself is worrying, and representative of the IoT industry as a whole. While it is possible for an organization to maintain a flexible, mature process for handling unsolicited vulnerability reports, it is far from the norm. It is hoped that the publication of these findings will help IoT vendors establish reasonable, effective vulnerability handling practices.

07
DISCLOSURES

What follows are the ten vulnerabilities reported to the vendors (when the vendor could be reached), to CERT, and ultimately, disclosed at the High Technology Crime Investigation Association (HTCIA) conference on September 2, 2015. Each vendor was provided with an opportunity to address their product vulnerabilities in advance of this public disclosure, in accordance with Rapid7's Disclosure Policy.

Vendor: iBaby Labs, Inc.

The issues for the iBaby devices were disclosed to CERT under vulnerability note VU#745448.

Device: iBaby M6

The vendor's product site for the device assessed is https://ibabylabs.com/ibaby-monitor-m6

Vulnerability R7-2015-11.1: Predictable public information leak (CVE-2015-2886)

The web site ibabycloud.com has a vulnerability by which any authenticated user to the ibabycloud.com 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 ibabycloud.com service, until those videos are deleted, without any further authentication.

Relevant URLs

Access a camera's details, including video-recording filenames: http://www.ibabycloud.com/cam/index/camid/{serial_number}/camtype/{cam_type} [any authenticated user]

Access a camera's video recording: http://d3a9yv3r4ycsw2.cloudfront.net/monitor/alert/{serial_number}/{filename}[no authentication required]

Additional Details

The ibabycloud.com authentication procedure has been non-functional as of at least June 2015, continuing through the publication of this paper in September 2015. These errors started after testing was conducted for this research, and today, do not allow for logins to the cloud service. That noted, it may be possible to still get a valid session via the API and subsequently leverage the site and API to gain these details.

Mitigations

Today, this attack is more difficult without prior knowledge of the camera's serial number, as all logins are disabled on the ibabycloud.com 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.

Device: iBaby M3S

The vendor's product site for the device assessed is https://ibabylabs.com/ibaby-monitor-m3s

Vulnerability R7-2015-11.2, Backdoor Credentials (CVE-2015-2887)

The device ships with hardcoded credentials, accessible from a telnet login prompt and a UART interface, which grants access to the underlying operating system. Those credentials are detailed below.

Operating System (via Telnet or UART)
Username: admin
Password: admin

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.

Disclosure Timeline

Sat, Jul 04, 2015: Initial contact to vendor

Mon, Jul 06, 2015: Vendor reply, requesting details for ticket #4085

Tue, Jul 07, 2015: Disclosure to vendor
Tue, Jul 21, 2015: Disclosure to CERT
Fri, Jul 24, 2015: Confirmed receipt by CERT

Wed, Sep 02, 2015: Public disclosure

Wed, Sep 02, 2015: iBaby Labs communicated that access token expiration and secure communication channels have been implemented.

Note: According to iBaby Labs, it contacted Rapid7 by email on August 8 to let us know that access token expiration and secure communication channels had been implemented. We did not receive the message, and therefore did not learn about the changes until we received a communication on September 2, after this report was first published.

Vendor: Philips Electronics N.V.

The issue for the Philips device was disclosed to CERT under vulnerability note VU#569536.

Device: Philips In.Sight B120/37

The vendor's product site for the device assessed is http://www.usa.philips.com/c-p/B120_37/in.sight-wireless-hd-baby-monitor

Vulnerability R7-2015-12.1, Backdoor Credentials (CVE-2015-2882)

The device ships with hardcoded 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.

The web service "admin" statically-generated password was first documented by Paul Price at his blog[5].

In addition, while the telnet service may be disabled by default on the most recent firmware, it can be re-enabled via an issue detailed below.

Operating System (via Telnet or UART)
Username: root
Password: b120root

Operating System (via Telnet or UART)
Username: admin
Password: /ADMIN/

Operating System (via Telnet or UART)
Username: mg3500
Password: merlin

Local Web Server
Reachable via http://{device_ip}/cgi-bin/{script_path}
Username: user
Password: M100-4674448

Local Web Server
Reachable via http://{device_ip}/cgi-bin/{script_path}
Username: admin
Password: M100-4674448

- *A recent update changes this password, but the new password is simply the letter 'i' prefixing the first ten characters of the MD5 hash of the device's MAC address.*

Vulnerability R7-2015-12.2, Reflective and Stored XSS (CVE-2015-2883)

A web service used on the backend of Philips' cloud service to create remote streaming sessions is vulnerable to reflective and stored XSS. Subsequently, session hijacking is possible due to a lack of an HttpOnly flag.

When accessing the Weaved cloud web service[6] as an authenticated user, multiple pages have a mixture of reflective and stored XSS in them, allowing for potential session hijacking. With this access, a valid streaming session could be generated and eavesdropped upon by an attacker.

Two such examples are:

1. https://developer.weaved.com/portal/members/deviceSettings.php?id={mac_address}&name={base64_encoded_xss_string}

2. https://developer.weaved.com/portal/members/shareDevice.php?id={mac_address}&name={base64_encoded_xss_string}

Vulnerability R7-2015-12.3, Direct Browsing via Insecure Streaming (CVE-2015-2884)

The method for allowing remote viewing uses an insecure transport, does not offer secure streams protected from attackers, and does not offer sufficient protection for the the camera's internal web applications.

Once a remote viewing stream has been requested, a proxy connection to the camera's internal web service via the cloud provider Yoics[7] is bound to a public hostname and port number. These port numbers appear to range from port 32,000 to 39,000 as determined from testing.This bound port is tied to a hostname with the pattern of proxy[1,3-14].yoics.net, limiting the potential number of port and host combinations to an enumerable level. Given this manageable attack space, attackers can test for an HTTP 200 response in a reasonably short amount of time.

Once found, administrative privilege is available without authentication of any kind to the web scripts available on

the device. Further, by accessing a Unicode-enabled streaming URL (known as an "m3u8" URL), a live video/audio stream will be accessible to the camera and appears to stay open for up to one hour on that host/port combination. There is no blacklist or whitelist restriction on which IP addresses can access these URLs, as revealed in testing.

Relevant URLs

Open audio/video stream of a camera: http://proxy{1,3-14}.yoics.net:{32000-39000}/tmp/stream2/stream.m3u8 [no authentication required]

Enable Telnet service on camera remotely: http://proxy{1,3-14}.yoics.net:{32000-39000}/cgi-bin/cam_service_enable.cgi [no authentication required]

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.

Additional Information

Prior to publication of this report, Philips confirmed with Rapid7 the tested device was discontinued by Philips in 2013, and the current manufacturer and distributor is Gibson Innovations. Gibson has developed a solution for the identified vulnerabilities, an expects to make updates available by September 4, 2015.

Disclosure Timeline

Sat, Jul 04, 2015: Initial contact to vendor

Mon, Jul 06, 2015: Vendor reply, requesting details

Tue, Jul 07, 2015: Philips Responsible Disclosure ticket number 15191319 assigned

Tue, Jul 17, 2015: Phone conference with vendor to discuss issues

Tue, Jul 21, 2015: Disclosure to CERT

Fri, Jul 24, 2015: Confirmed receipt by CERT

Thu, Aug 27, 2015: Contacted by Weaved to validate R7-2015-12.2

Tue, Sep 01, 2015: Contacted by Philips regarding the role of Gibson Innovations

Wed, Sep 02, 2015: Public disclosure

Sat, Sep 05, 2015: Affected cloud services updated

Fri, Sep 11, 2015: Insight firmware updated to version 7.4

Sat, Sep 12, 2015: Insight Android app updated

Thu, Sep 17, 2015: Insight iOS app updated

Vendor: Summer Infant

The issues for the Summer Infant device was disclosed to CERT under vulnerability note VU#837936.

Device: Summer Baby Zoom WiFi Monitor & Internet Viewing System

The vendor's product site for the device assessed is http://www.summerinfant.com/monitoring/internet/babyzoomwifi.

Vulnerability R7-2015-13.1, Authentication Bypass (CVE-2015-2888)

An authentication bypass allows for the addition of an arbitrary account to any camera, without authentication.

The web service MySnapCam[8] is used to support the camera's functionality, including account management for access. A URL retrievable via an HTTP GET request can be used to add a new user to the camera. This URL does not require any of the camera's administrators to have a valid session to execute this request, allowing anyone requesting the URL with their details against any camera ID to have access added to that device.

After a new user is successfully added, an e-mail will then be sent to an e-mail address provided by the attacker with authentication details for the MySnapCam website and mobile application. Camera administrators are not notified of the new account.

Relevant URL

Add an arbitrary user to any camera: https://swifiserv.mysnapcam.com/register/?fn={first_name}&ln={last_name}&email={email}&userType=3&userGroup={id} [no authentication required]

Vulnerability R7-2015-13.2, Privilege Escalation (CVE-2015-2889)

An authenticated, regular user can access an administrative interface that fails to check for privileges, leading to privilege escalation.

A "Settings" interface exists for the camera's cloud service administrative user and appears as a link in their interface when they login. If a non-administrative user is logged in to that camera and manually enters that URL, they are able to see the same administrative actions and carry them out as if they had administrative privilege. This allows an unprivileged user to elevate account privileges arbitrarily.

Relevant URL

Access administrative actions as an unprivileged, but valid, user: https://www.summerlinkwifi.com/settings_users.php [a user account for the camera is required]

Mitigations

In order to avoid exposure to the authentication bypass and privilege escalation, customers should use the

device in a local network only mode, and use egress firewall rules to block the camera from the Internet. If Internet access is desired, customers should inquire about an update to Summer Infant's cloud services.

Disclosure Timeline

Sat, Jul 04, 2015: Initial contact to vendor

Tue, Jul 21, 2015: Disclosure to CERT

Fri, Jul 24, 2015: Confirmed receipt by CERT

Tue, Sep 01, 2015: Confirmed receipt by the vendor

Wed, Sep 02, 2015: Public disclosure

Wed, Sep 02, 2015: Summer Infant tweeted that all reported issues have been resolved

Vendor: Lens Laboratories(f)

The issues for the Lens Laboratories(f) device was disclosed to CERT under vulnerability note VU#931216.

Device: Lens Peek-a-View

The vendor's product site for the device assessed is http://www.amazon.com/Peek---view-Resolution-Wireless-Monitor/dp/B00N5AVMQI/

Of special note, it has proven difficult to find a registered domain for this vendor. All references to the vendor point at Amazon directly, but Amazon does not appear to be the manufacturer or vendor.

Vulnerability R7-2015-14, Backdoor Credentials (CVE-2015-2885)

The device ships with hardcoded credentials, accessible from a UART interface, which grants access to the underlying operating system, and via the local web service, giving local application access via the web UI.

Due to weak filesystem permissions, the local OS 'admin' account has effective 'root' privileges.

Operating System (via UART)
Username: admin
Password: 2601hx

Local Web Server
Site: http://{device_ip}/web/
Username: user
Password: user

Local Web Server
Site: via http://{device_ip}/web/
Username: guest
Password: guest

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.

Disclosure Timeline

Sat, Jul 04, 2015: Attempted to find vendor contact

Tue, Jul 21, 2015: Disclosure to CERT

Fri, Jul 24, 2015: Confirmed receipt by CERT

Wed, Sep 02, 2015: Public disclosure

Vendor: Gynoii, Inc.

The issues for the Gynoii devices was disclosed to CERT under vulnerability note VU#738848.

Device: Gynoii

The vendor's product site for the device assessed is http://www.gynoii.com/product.html

Vulnerability R7-2015-15, Backdoor Credentials (CVE-2015-2881)

The device ships with hardcoded credentials, accessible via the local web service, giving local application access via the web UI.

Local Web Server
Site: http://{device_ip}/admin/
Username: guest
Password: guest

Local Web Server
Site: http://{device_ip}/admin/
Username: admin
Password: 12345

Mitigations

In order to disable these credentials, customers should inquire with the vendor about a firmware update.

Disclosure Timeline

Sat, Jul 04, 2015: Initial contact to vendor

Tue, Jul 21, 2015: Disclosure to CERT

Fri, Jul 24, 2015: Confirmed receipt by CERT

Wed, Sep 02, 2015: Public disclosure

Wed, Sep 02, 2015: Gynoii acknowledged the above research shortly after publication and are assessing appropriate patch strategies.

Vendor: TRENDnet

The issue for the TRENDnet device was disclosed to CERT under vulnerability note VU#136207.

Device: TRENDnet WiFi Baby Cam TV-IP743SIC

The vendor's product site for the device under test is http://www.trendnet.com/products/proddetail.asp?prod=235_TV-IP743SIC

Vulnerability R7-2015-16: Backdoor Credentials (CVE-2015-2880)

The device ships with hardcoded credentials, accessible via a UART inter-

face, giving local, root-level operating system access.

Operating System (via UART)
Username: root
Password: admin

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.

Disclosure Timeline

Sat, Jul 04, 2015: Initial contact to vendor

Mon, Jul 06, 2015: Vendor reply, details disclosed to vendor

Sun, Jul 16, 2015: Clarification sought by vendor

Mon, Jul 20, 2015: Clarification provided to vendor

Tue, Jul 21, 2015: Disclosure to CERT

Wed, Sep 02, 2015: Public disclosure

Thu, Sep 03, 2015: TRENDnet reports updated firmware available here (version 1.0.3), released on Sep 02, 2015.

[5] http://www.ifc0nfig.com/a-close-look-at-the-philips-in-sight-ip-camera-range/

[6] http://www.weaved.com/

[7] https://www.yoics.net

[8] http://www.mysnapcam.com/

08

WORKING TO IMPROVE IoT SECURITY

It is the authors' hope that everyone who reads this paper has a better sense of security issues facing the current generation of the Internet of Things. While we take great pride in performing research on individual IoT devices that have real-world benefits to consumers and businesses, we also realize that those efforts alone don't scale to the massive size and growth of IoT.

In February 2014, Mark Stanislav co-founded the IoT security initiative, BuildItSecure.ly.[9] Through vendor outreach efforts, BuildItSecure.ly not only provides curated information security guidance to IoT vendors of all sizes, but also pairs those vendors with highly regarded information security researchers. Through this pro bono, coupled approach, BuildItSecure.ly is able to translate research and knowledge transfer into real security improvements that will impact the entire product line of participating vendors.

Additionally, Mark also participates in the Online Trust Alliance's IoT Working Group[10], which is developing the "IoT Trust Framework" to provide clear guidance to vendors on expectations of both privacy and information security features for their products. Vendors that utilize this framework will have a set of minimum boundaries for how their products and related services should handle the data and trust being provided to them by their customers. By establishing this framework, vendors can be confident in how to approach tough design and implementation choices that produce high quality, secure, and affordable products.

[9] http://builditsecure.ly/

[10] https://otalliance.org/initiatives/internet-things

09
ABOUT RAPID7

Rapid7 is a leading provider of security data and analytics solutions that enable organizations to implement an active, analytics-driven approach to cyber security. We combine our extensive experience in security data and analytics and deep insight into attacker behaviors and techniques to make sense of the wealth of data available to organizations about their IT environments and users. Our solutions empower organizations to prevent attacks by providing visibility into vulnerabilities and to rapidly detect compromises, respond to breaches, and correct the underlying causes of attacks. Rapid7 is trusted by more than 4,150 organizations across 90 countries, including 34% of the Fortune 1000. To learn more about Rapid7 or get involved in our threat research, visit www.rapid7.com.