Retrofitting Security and Privacy Measures to Smart Home Devices

Citation for published version:
Ye, C, Indra, PP & Aspinall, D 2019, Retrofitting Security and Privacy Measures to Smart Home Devices. in *2019 Sixth International Conference on Internet of Things: Systems, Management and Security (IOTSMS).* IEEE Xplore, pp. 283-290, 6th International Conference on Internet of Things: Systems, Management and Security , Granada, Spain, 22/10/19. https://doi.org/10.1109/IOTSMS48152.2019.8939272

Digital Object Identifier (DOI):
10.1109/IOTSMS48152.2019.8939272

Link:
Link to publication record in Edinburgh Research Explorer

Document Version:
Peer reviewed version

Published In:
2019 Sixth International Conference on Internet of Things: Systems, Management and Security (IOTSMS)

Retrofitting Security and Privacy Measures to Smart Home Devices

Chenghao Ye
School of Informatics
The University of Edinburgh
Edinburgh, United Kingdom
s1786987@ed-alumni.net

Praburam Prabhakar Indra
School of Informatics
The University of Edinburgh
Edinburgh, United Kingdom
s1788279@ed-alumni.net

David Aspinall
School of Informatics
The University of Edinburgh
Edinburgh, United Kingdom
David.Aspinall@ed.ac.uk

***Abstract*—There is a current trend for Internet of Things (IoT) technology in the home. However, device vendors provide no guarantees of security or privacy of their gadgets, nor can such things be measured by consumers. By now, there have been many incidents of vulnerable devices being sold and real-world attacks. Despite proposals for improving the quality of consumer devices, vulnerable devices are likely to remain in use, with it being highly difficult to replace or patch their hardware or software.**

In this paper, we set out to design a mitigation framework so that home networks can be made resilient to vulnerable devices. First, we select a representative collection of home IoT devices with different functions, and investigate their security and privacy, discovering a range of exploitable flaws. Then we design a framework based on a dedicated router, firewall, an IoT control platform and other mechanisms, which allows mitigation of current and potential future vulnerabilities. The framework is designed to be adaptable and extensible for all kinds of devices. We implement this framework and evaluate it against the sample devices, finding that it can indeed prevent most of the known exploits and the new exploits we found. Based on this study, we make some design suggestions for the future enhanced home cyber-security platforms.

***Index Terms*—Computer network security, Information Privacy, Internet-of-Things**

I. Introduction

Smart home Internet of Things (IoT) devices like cameras, lights, sockets and doorbells are increasingly used to improve the quality of daily life. While consumers are enjoying the convenience brought by smart home devices, the security and privacy aspects have not been well designed from their inception. Arguably, such consumer devices and their networks should be seen as part of future Critical National Infrastructures, for several reasons. First, consumer devices control and influence citizens' everyday lives and impact safety, security, and privacy in their own homes; violations like the TRENDnet webcam hack are alarming [1]. Second, their reach has extended beyond their anticipated scope: homes are becoming healthcare environments and connected health devices rely on home networks; conversely, consumer-grade IoT devices are being used by enterprise and industry (due to price, availability or accessibility) [2]. Finally, as demonstrated by the Mirai Botnet [3], home networks are a frighteningly effective attack vector on global Internet services.

Future homes clearly need more sophisticated cyber-security architecture. Security must be "designed in", assisted by emerging guidelines, tests, and standards for consumer connected devices [4], [5]. But security must also be "designed around", to protect devices that pre-date security standards, fail to have patches applied or built [6], or simply become targets of previously unknown attacks.

This paper addresses the "designed around" need. Of course, the working principles between one device and another vary quite widely, due to different functionalities, design choices and vendors. Consequently, the creation of universal patches is impossible, and it is hard to come up with a fully general solution to automatically mitigate all possible flaws. However, we believe it is possible to create a general framework outside the gadgets which can solve most of the problems with a small number of configurable parameters, which in the future could be controlled remotely or via a database of known IoT repairs.

To investigate this idea, we first take a set of common IoT devices and test them to find vulnerabilities and exploits. Then a security infrastructure for in effect "retrofitting" the devices is designed and implemented, along with several general and device-specific measures to ensure the security and privacy of the users are not compromised. Our contributions are:

- Discovery of a number of previously unreported vulnerabilities in some common devices;
- As far as we know, the first heterogeneous IoT mitigation framework designed for a suite of different devices combining active firewalling and an IoT control platform;
- A mitigation strategy for our test suite of 7 different device types.

The mitigation framework was built and tested in a lab environment. For future real deployment, the techniques could be built into a commodity home router or security appliance product. The controls require configuration which is too difficult for ordinary end-users, we expect that future security solutions will allow automatic configuration from a range of recognised devices, perhaps controlled remotely by security service providers or ISPs.

The rest of this paper is as follows. In Section II we introduce the devices and describe the security testing performed and its results. In Section III we introduce the infrastructure that makes up our mitigation framework, motivating its design. Section IV then explains how the framework is configured for each of the test devices. The results in Section V show that by re-testing security, most of the previously identified problems are mitigated. Finally, Sections VI and VII summarise some of the related and future work.

II. Device analysis

A. Threat modelling

A complete smart home device consists of a physical device, often a smartphone app or a control terminal as the user interface, a remote server and connections between endpoints. For a smart home system, threats can come from two origins:

- **Wide-area network (WAN)** An attacker can capture communication between the smart home and remote locations. A remote adversary may interfere with normal traffic to exfiltrate information, mount a man-in-the-middle attack or control devices remotely.
- **Local-area network (LAN)** A local attacker can moreover eavesdrop LAN traffic or launch unrestricted network attacks on devices directly.

Some devices may furthermore be exploitable via physical methods. We omit discussion of physical countermeasures in this paper.

Most attacks on IoT devices can be associated with one of the following groups, according to their scope and purpose [7]:

- **Denial of service (DoS)** The attacker aims to stop legitimate users from accessing the service (e.g., stop a fire alarm from responding).
- **Data exfiltration** The attacker aims to capture traffic at different points of the transmission to extract valuable information (e.g., get the hardware information by analysing the traffic).
- **Software exploitation** The attacker aims to alter the function of the device firmware and execute malicious orders (e.g., the attack of Mirai Botnet [3]).
- **Cyber-physical remote control** The unauthorised attacker aims to get the privileges of the device and control remotely (e.g., turn on the victim's light at midnight).

We analysed and tested several IoT devices step by step, with a focus on the top 10 most-seen vulnerabilities listed in the OWASP Internet of Things project [8]. Seven different types of common smart home devices are chosen to represent a typical smart home environment. Information about the devices is listed in Table I.

The attacks to these devices can cause actual loss or damage. Here are some typical threat scenarios which can be achieved via exploiting the vulnerabilities we found (described later in Section II-C and Table IV):

- **Hive hub and light** The adversary can find out the geolocation of the device by sniffing IP addresses.
- **Feed and Go pet feeder** The attacker can potentially brick the pet feeder and starve the pet, by crashing the internal control command parser or alter the commands sent from the application. The built-in camera could be compromised to spy on the owner's home.
- **iKettle 1.0** The attacker can use forged commands to keep the kettle heating up even if the water boils dry. This can produce immense heat and potentially cause fire.
- **Ring Doorbell** The attacker can insert malicious code to the device during the insecure firmware updating process.

TABLE I
Device name and firmware version

Device name	Firmware version
WeMo Insight Switch and WeMo iOS app [9]	WeMo WW 2.00.3007.PVT and 1.19.1(491000)
Hive Hub and Hive Active Light [10]	1.0.0-5927-35.0 and 11340002
Ring Video Doorbell 1.0 [11]	Unidentifiable Latest Version
Feed and Go Pet Feeder [12]	Unidentifiable
iKettle 1.0 and iKettle app [13]	Unidentifiable and 3.0
BT Smart Home Cam 100 [14]	Earlier than 0.0.9.1
LeFun C2 Wireless Camera [15]	v4.3.1.1703291555

B. Methods used for analysis

Several methods are used to find and test the potential vulnerabilities of the devices:

- **Compare to known attacks**: Some of the devices or parts of the devices (e.g., a specific model of the chip) have known vulnerabilities already made public.
- **Port scanning, packet-sniffing and analysing**: The major way to understand the traffic flow of devices and find new flaws.
- **Simulate the attack**: Launch the known attack in a lab environment and try to understand the device's security mechanisms from the attacker's perspective.
- **Reverse engineering**: Use reverse engineering methods to understand the principles of the device. For those devices that use an app, decompiling the app provides a chance to look into the logic behind the device implementation and protocol. (Modifying the firmware itself is beyond the scope of this project.)

The tools we used were standard tools such as Wireshark, Nmap and Android Studio alongside using scripting tools to automate exploits and vulnerability discovery to act as baseline security tests to use after our framework was implemented.

C. Result of analysis

First, we selected 7 representative smart devices including a switch, light, doorbell, pet feeder, kettle and smart cameras. The devices were chosen to cover a range of very different functions and also exhibit a range of working principles. Then, we evaluated the devices according to three major attributes out of thirteen in the "Secure by Design" [4] guidelines, the UK government's code of practice for consumer Internet of Things (IoT) Security for manufacturers. These three are the most technical and intrinsic attributes which can be evaluated; others include advice concerning the service provided around

TABLE II
SELECTED SECURE BY DESIGN GUIDELINES COMPLIANCE

	WeMo Insight Switch	Ring Video Doorbell	Feed and Go Pet Feeder
No default passwords	✓	✓	✓
Keep software updated	Left to the user	✓	Company out of business
Communicate securely	Insecure communication between the device and the app	Insecure communication for call with the doorbell	Insecure communication over HTTP with the cloud

	iKettle 1.0	BT smart home camera 100	LeFun camera C2
No default passwords	Default password 000000 not changed	✓	✓
Keep software updated	No automatic firmware update nor user notification	Left to the user	Left to the user
Communicate securely	Insecure between the device and the app	✓	Insecure communication over RTP with the cloud

IoT gadgets, such as ensuring secure storage on manufacturer cloud servers, gathering telemetry or implementing vulnerability disclosure procedures. The results are shown in Table II.[1]

Overall, we found 17 vulnerabilities found in the 7 devices, including the vulnerabilities found in the penetration testing or reported in previous studies and news [16]–[20]. Most of them can be considered as common vulnerabilities among IoT devices listed by the OWASP IoT list [8].

Some major vulnerabilities found are briefly listed below:

- **WeMo Insight Switch.** The TLSv1.0 protocol used for communication between the device and the cloud is no longer considered a secure protocol. Moreover, information is not fully encrypted between the app and the device.
- **Hive Hub and Hive Active Light.** One API call will send location information in HTTP and DNS queries are not encrypted.
- **Ring Video Doorbell.** Firmware updating, STUN messages and DNS queries are not encrypted. The attacker may be able to view or alter the traffic.
- **Feed and Go Pet Feeder.** No traffic is encrypted between the device, the app and the cloud.[2]
- **iKettle 1.0.** The default PIN of the device (000000) is not changed and sensitive information (home Wi-Fi username and passphrase) is stored in the device in the format of plain text. Anyone in the LAN can access and change the information, settings or the PIN itself stored in the kettle using the default PIN. Attackers can also send forged commands repeatedly to keep the kettle boiling.
- **BT Smart Home Cam 100.** The adversary may be able to detect user activities by analysing the traffic because the motion detection function gives feedback to the server every few seconds. (The network pattern of the feedback is different according to the result of the feedback)
- **LeFun C2 Wireless Camera.** The video stream uses RTP protocol and DNS queries are not encrypted. Data in portable storage is not encrypted.

More detailed information of the vulnerabilities is summarised in Table IV in Section V.

In Table III, we classify all the vulnerabilities we found and match them to the taxonomy we devised in Section II-A.

TABLE III
CLASSIFICATION OF THE VULNERABILITIES

	Denial of service	Data exfiltration	Software exploitation	Remote control
WeMo Insight Switch		✗		
Hive Hub and light		✗		
Ring Video Doorbell		✗	✗	
Feed and Go Pet Feeder	✗	✗		✗
iKettle 1.0	✗	✗		✗
BT Smart Home Cam 100		✗		
LeFun C2 Wireless Camera		✗		

D. Disclosure and response

According to responsible disclosure, the new vulnerabilities were reported to the vendors. Responses from the corresponding vendors vary significantly. The manufacture for the Feed and Go Pet feeder went out of business. Vulnerabilities related to the BT smart home camera were fixed in later firmware updates. The iKettle company was informed of the problems but decided not to take any actions since the model is considered as an older generation and the vendor is more focused on developing new products. Some of the companies like the LeFun did not reply to our email.

Despite these cases, the life expectancy for an IoT device can be more than 10 years [22], often significantly longer than the support end date provided by the vendor. Devices will run on an outdated firmware will face high risks during that time. Hence, infrastructures to stop potential problems and patch existing flaws will be necessary for a smart home system.

III. INFRASTRUCTURE DESIGN

Based on the threat taxonomy in Section II-A for IoT and combined with some common IoT security and privacy flaws found in sample devices discussed in Section II-C, we propose a security and privacy infrastructure. The main goal of building this infrastructure is to isolate the vulnerable devices from

[1] HIVE hub and light satisfies all three requirements so is not listed.

[2] Shortly after our work, the official cloud server was shutdown. Based on the YouTube views of the instruction video [21], it is estimated that 5000 products have been sold. However, other IoT devices may have a similar structure and platform.

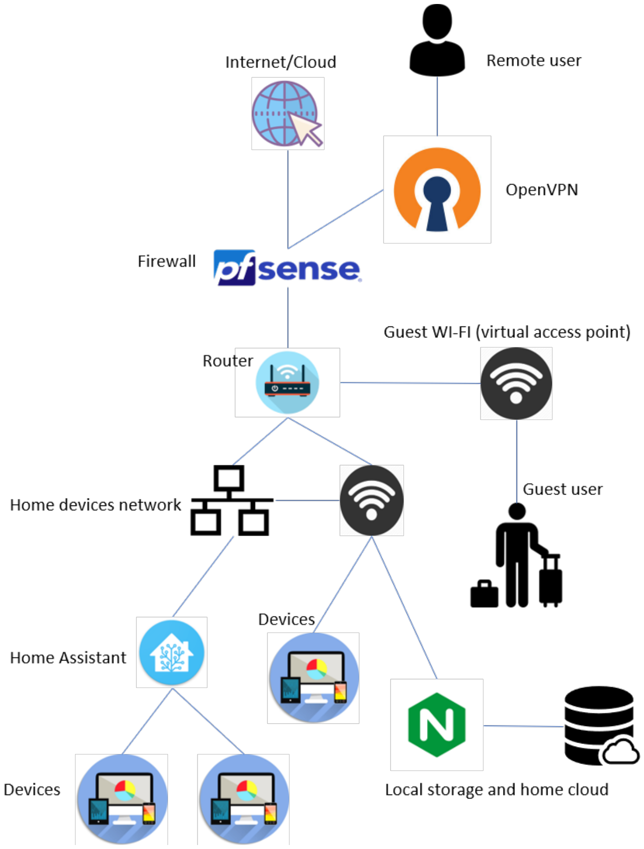
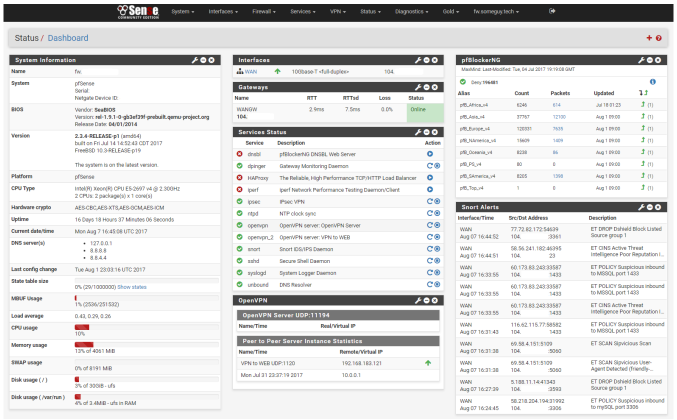


Fig. 1. Designed infrastructure

attackers and retain as many functionalities as possible. It also provides an alternative way to control some of the devices so that they will not be restricted to official clouds only to avoid insecure cloud or the case of cloud goes offline or provide after-market support for neglected devices.

The prototype framework is designed using a mixture of easily available hardware and software components as shown in Fig. 1. Configuration of these components requires considerable understanding of security and networking. The aim of our project is to evaluate the feasibility of the framework; in a realisation of this framework for consumers, configuration would not be performed by the end users, it could be provided as a service or a library of pre-configured devices in a security knowledgebase.

- **pfSense Firewall [23]**
  pfSense is a highly configurable open-source free firewall software based on FreeBSD. In our setup, it is placed between the home LAN and WAN. All the network flow to the cloud servers have to go through the pfSense.
  It has many relevant features such as rate limits and alarms, multi-user support, OpenVPN to provide remote secure access, DHCP server as well as traditional filtering firewall functions.
  In our prototype, pfSense is installed on an HP workstation between the router and the WAN as a dedicated firewall and gateway. The LAN interface is connected to the LAN port of the Wi-Fi router. Customised rules and logging are configured for each of the connected devices. Fig. 2 shows a configuration page of pfSense in its web UI.



Status / Dashboard

System Information

Name	fw.
System	pfSense Serial: Netgate Device ID:
BIOS	Vendor: **SeaBIOS** Version: **rel-1.9.1-0-gb3ef39f-prebuilt.qemu-project.org** Release Date: **04/01/2014**
Version	**2.3.4-RELEASE-p1** (amd64) built on Fri Jul 14 14:52:43 CDT 2017 FreeBSD 10.3-RELEASE-p19 The system is on the latest version.
Platform	pfSense
CPU Type	Intel(R) Xeon(R) CPU E5-2697 v4 @ 2.30GHz 2 CPUs: 2 package(s) x 1 core(s)
Hardware crypto	AES-CBC,AES-XTS,AES-GCM,AES-ICM
Uptime	16 Days 18 Hours 37 Minutes 06 Seconds
Current date/time	Mon Aug 7 16:45:08 UTC 2017
DNS server(s)	• 127.0.0.1 • 8.8.8.8 • 8.8.4.4
Last config change	Tue Aug 1 23:03:16 UTC 2017
State table size	0% (29/1000000) Show states
MBUF Usage	1% (2536/251532)
Load average	0.43, 0.29, 0.26
CPU usage	10%
Memory usage	13% of 4061 MiB
SWAP usage	0% of 8191 MiB
Disk usage (/)	3% of 30GiB - ufs
Disk usage (/var/run)	4% of 3.4MiB - ufs in RAM

Interfaces

WAN	↑	10Gbase-T <full-duplex>	104.

Gateways

Name	RTT	RTTsd	Loss	Status
WANGW 104.	2.9ms	7.5ms	0.0%	Online

Services Status

	Service	Description	Action
☒	dnsbl	pfBlockerNG DNSBL Web Server	▶
✓	dpinger	Gateway Monitoring Daemon	↻ ◉
☒	HAProxy	The Reliable, High Performance TCP/HTTP Load Balancer	▶
☒	iperf	iperf Network Performance Testing Daemon/Client	▶
✓	ipsec	IPsec VPN	↻ ◉
✓	ntpd	NTP clock sync	↻ ◉
✓	openvpn	OpenVPN server: OpenVPN Server	↻ ◉
✓	openvpn_2	OpenVPN server: VPN to WEB	↻ ◉
✓	snort	Snort IDS/IPS Daemon	↻ ◉
✓	sshd	Secure Shell Daemon	↻ ◉
✓	syslogd	System Logger Daemon	↻ ◉
✓	unbound	DNS Resolver	↻ ◉

OpenVPN

OpenVPN Server UDP:11194

Name/Time	Real/Virtual IP

Peer to Peer Server Instance Statistics

Name/Time	Remote/Virtual IP	
VPN to WEB UDP:1120	192.168.183.121	↑
Mon Jul 31 23:37:19 2017	10.0.0.1	

pfBlockerNG

MaxMind: Last-Modified: Tue, 04 Jul 2017 19:19:08 GMT

✓ Deny:**196481**

Alias	Count	Packets	Updated	
pfB_Africa_v4	6246	614	Jul 18 01:23	↑ (1)
pfB_Asia_v4	37767	12100	Aug 1 09:00	↑ (1)
pfB_Europe_v4	120331	7635	Aug 1 09:00	↑ (1)
pfB_NAmerica_v4	15609	1409	Aug 1 09:00	↑ (1)
pfB_Oceania_v4	8238	86	Aug 1 09:00	↑ (1)
pfB_PS_v4	80	0	Aug 1 09:00	↑ (1)
pfB_SAmerica_v4	8205	1398	Aug 1 09:00	↑ (1)
pfB_Top_v4	1	0	Aug 1 09:00	↑ (1)

Snort Alerts

Interface/Time	Src/Dst Address	Description
WAN Aug 07 16:44:52	77.72.82.172:54639 104. :3361	ET DROP Dshield Block Listed Source group 1
WAN Aug 07 16:44:51	58.56.241.182:46395 104. 23	ET CINS Active Threat Intelligence Poor Reputation I...
WAN Aug 07 16:33:55	60.173.83.243:33587 104. 1433	ET POLICY Suspicious inbound to MSSQL port 1433
WAN Aug 07 16:33:55	60.173.83.243:33587 104. 1433	ET POLICY Suspicious inbound to MSSQL port 1433
WAN Aug 07 16:33:55	60.173.83.243:33587 104. 1433	ET CINS Active Threat Intelligence Poor Reputation I...
WAN Aug 07 16:31:43	116.62.115.77:58582 104. 1433	ET POLICY Suspicious inbound to MSSQL port 1433
WAN Aug 07 16:31:38	69.58.4.151:5109 104. :5060	ET SCAN Sipvicious Scan
WAN Aug 07 16:31:38	69.58.4.151:5109 104. :5060	ET SCAN Sipvicious User-Agent Detected (friendly-...
WAN Aug 07 16:27:39	5.188.11.14:41343 104. :3593	ET DROP Dshield Block Listed Source group 1
WAN Aug 07 16:24:45	58.218.204.194:31992 104. :3306	ET POLICY Suspicious inbound to mySQL port 3306

Fig. 2. pfSense user interface

- **WiFi and LAN Router**:
  All the devices in the smart home are connected to a simple home router. A virtual access point (VAP) is configured to separate between trusted smart home devices, users and other users (guest users). For the smart home devices, each device's MAC address is mapped to a static IP address. Network isolation is enabled on the guest network so devices cannot find other guest users or route smart home devices.
- **OpenVPN [24]**
  OpenVPN is open-source software that implements virtual private network (VPN) to create secure point-to-point between the user and the infrastructure. An OpenVPN server on the WAN interface of the firewall is configured for remote secure access.
- **Home Assistant platform [25]**
  Home Assistant is a platform for smart home automation and integration. The Home Assistant community supports over 1000 types of IoT device. Supported devices can be added and controlled via the platform so that the user does not need to install dedicated software. Moreover, the platform can avoid some known problems of the device control apps (especially ones which are not maintained or updated).
  We chose this platform for its convenience and facilities; other platforms are available. In our setup, Home Assistant is installed on a Raspberry Pi 3B. The Raspberry Pi together with other home devices are connected to the router under the same LAN.
  In Fig. 3 we can see control options on Home Assistant like the WeMo Insight switch and Hive Active Light. The web dashboard can be customised and configured by the user and is accessible either from LAN devices or from WAN through pfSense using VPN.

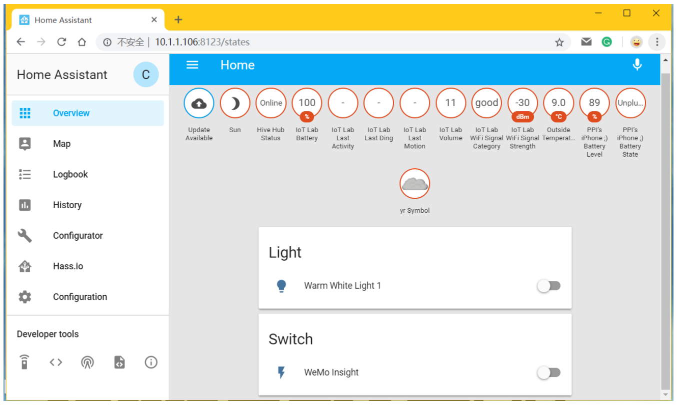


Fig. 3. Home Assistant dashboard

- **Customised local server**
  An Nginx server on a Raspberry Pi is attached to the network to work as a customised local server for storing user data. The purpose of this local server is to replace some of the insecure official servers or avoid unencrypted traffic flow to the remote server (e.g., the LeFun camera RTP video stream described in Section II-C).

IV. Configurations and Retrofitting Measures

To apply the framework, we use several features to improve the security and privacy aspects of the smart home.

- **Proper firewall configuration**
  A firewall is an essential part of network security and the main component to defend against attackers.
  Mandatory rules are made according to the vulnerabilities to block unnecessary data or redirect risky traffic. Some of the data, for instance, the video stream of the LeFun camera, is blocked at the firewall level and redirected to the local data server for storage, replacing the official video server to avoid information leaks.
  The DHCP server is configured on pfSense. We use MAC access control both on the WiFi router and in the trusted device list in the pfSense DHCP configurations. So only authentic connected devices may join the home network VAP and receive an IP lease.
  Another feature of pfSense we use extensively is the OpenVPN server endpoint. Certificates and user credentials are used for the VPN and UDP used as tunnel protocol. Multiple users can connect using their own private keys.
  Finally, we configured a cron job to kill the firewall status for the purpose of "port shuffling": after a period, communication is re-established with new ports over the NAT. This can cause minor momentary disconnections but makes it harder for attackers to scan the devices or find patterns of activities [26].

- **Using secure DNS service**
  Sending insecure DNS queries leaks information about user activity. In the setup, all the DNS queries made by smart home devices are sent to port 53 via UDP to the pfSense firewall. Then the DNS resolver for pfSense is configured to query 1.1.1.1 (Cloudflare DNS [27]). The connection to this server is encrypted with SSL/TLS to prevent information leakage.

- **Use virtual access points**
  In the smart home environment, separating the device network and guest network can help reduce the risk from guest devices. To achieve this, we enabled the VAP function in the setup. A virtual access point "Guest" is created and separated from the AP for smart home devices. The guest network is protected with WPA2-PSK (AES) encryption. Usually, a guest connects to the Wi-Fi is to surf the Internet, so wireless isolation is enabled. The SSID of the home device Wi-Fi is hidden.

- **Use virtual LAN to group devices**
  VAP helps isolate threats from a guest device. To defend against an infected smart home device already in the LAN, more steps are needed. To mitigate potential threats from inside of the LAN such as the iKettle or the WeMo case described in Section II-C or the situation of malware-infected devices in the LAN infected by malware, we use VLAN for network segmentation.
  The IEEE 802.1Q-2011 standard states [28]: A virtual local area network (VLAN) is a logical group of network devices that which are considered on the same LAN. VLAN is partitioned and isolated in a computer network at the data link layer (OSI layer 2).
  Devices on different VLANs use the same switch but cannot see each other. Thus, proper VLAN setup can prevent exploited devices from attacking other devices to spread malware or cause service problems. Devices are grouped and tagged using VLANs, so devices that need to communicate share a VLAN group, others are kept separate. For example, Home Assistant, WeMo switch and Hive light are allocated in VLAN group 1; iKettle and the phone installed with controlling app are in VLAN group 2.

- **Replace the official server/application**
  Any downtime of the official server may cause the device to be unavailable. In cases like the Feed and Go Pet Feeder, the official service may shut down altogether (the vendor went out of business and closed the website three years after the product came to the market, rendering the devices inoperative). Design flaws of the product or cloud app vulnerabilities may cause potential damage to the user; in general cloud servers may not do the proper job of securing user data [29].
  A solution to these issues is to build a centralised control mechanism which is controlled by the user. In our prototype, the Home Assistant platform is chosen to work as the control platform.
  We tested WeMo switch and Hive light controlled by

the platform instead of official apps. In both cases, the platform provides full functionality and solved known problems caused by flaws in the official app.

- **Device PIN**
  Many devices use a simple default PIN for authentication which might not be changed. For example, the iKettle has a default PIN 00000 used for the internal server, which is not user-configurable. To exploit the iKettle we built and tested an Android app prototype based on the original official app, but also allowing PIN modification. We then provided a secure route for sharing the PIN across multiple owners. This prototype solution could be extended to different devices for further development.

- **Apply latest patches**
  Finally, after identifying out-of-the-box vulnerabilities, we updated devices to the latest available official versions, where available, to test which problems have been addressed since product release. We also re-tested firmware released after we had disclosed vulnerabilities to vendors. Although conceptually a simple step, research has shown that many real-world users fail to apply firmware patches if it requires manual intervention.

V. Results and evaluation

We implemented the mitigation platform shown in Fig. 1 and configured our lab prototype as described in Section IV above. We then attached the vulnerable devices and re-tested if the vulnerabilities were mitigated.

The result is listed in Table IV. Among the 17 vulnerabilities, 13 of them are fully addressed (marked in the colour green), 2 of them are mitigated with limited loss of function (marked in yellow), and 5 of the vulnerabilities cannot be feasibly fixed without changing the device firmware or hardware, due to design flaws. The impacts of the vulnerabilities before retrofitting are assessed using our own assignment of CVSS v3.0 vectors to derive the base scores shown [30]. The number of vulnerabilities categorised in low severity, medium severity and high severity are 8, 7 and 2 respectively. Some of the vulnerabilities between devices are common, for example, many devices use insecure DNS communications between themselves and their servers, so could be subjected to man-in-the-middle attacks.

In the end, the success of retrofitting depends on the particular device and vulnerability concerned. It is seen that different devices follow different behaviour when interacting with the cloud or the app, and use different kinds of protocols. This is the reason that each device requires bespoke treatment in the firewall rule configuration that matches the protocols and ports used. Saying this, more than half of the identified problems were fixed by general solutions including general firewall rules and VLAN grouping. These methods can be applied to any type of new device attached to the network, regardless of the working principles, device specific features or whether the potential vulnerabilities are disclosed. So our prototype is suited to extend to more devices. Finally, some of the problems can only be solved by firmware, such as the issues fixed in BT Smart Home Cam 100's latest update.

VI. Related work

A very recently published SoK paper by Alrawi et.al [31] gives a useful survey of much of the past research in the last 10 years or so. Alongside, they evaluated a number of common IoT devices and applied some of the same approaches as this project. They discussed several security measures as we have but didn't deploy a comprehensive platform for device retrofitting.

Previous work before this survey includes Gupta et al. [32], who designed a cost-effective firewall solution based on Raspberry Pi for IoT devices. In their approach, the communications are filtered and only connections in the white list are allowed. This measure can prevent attacks which leveraged on open device ports and default credentials but lacks the full network segmentation and control aspects of our framework.

Apthorpe et al. [33] discuss four strategies to prevent smart home network monitoring by observers: blocking outgoing connections, encrypting DNS queries, using VPN tunnelling, and traffic shaping methods. Their threat modelling informed the approach we followed and traffic shaping goes beyond the measures we employed (although port shuffling is a simplified approach).

Sivaraman et al. [34] propose the idea of device-level protection augmented with network-level security solutions and software designed networks (SDN) to provide customised solutions. In future, SDN control of home devices appears as a strategy that router or security appliance vendor may adopt, to provide "Home Cyber Security as a Service".

Finally, Vincentius et al. [26] designed a comprehensive IoT defence to raise an attackers uncertainty about devices in the home network and enable the home network to monitor traffic, detect anomalies, and filter malicious packets. This is an example of extending tools for monitoring for unexpected attacks to home cybersecurity management, which is also likely to feature in future fully-formed solutions.

VII. Conclusion and future work

Our results on penetration testing show that security and privacy features of common IoT devices are far from satisfying. We reported the findings to the vendors, but so far only one of them published patches to solve the problems. Other companies either refused to make patches for their products, did not respond to our report or just went out of business. Four out of seven flawed devices remain exploitable and may have a widely installed base. Thus it is vital to use some framework to mitigate the problems at home level or ISP level.

In future work, we plan to experiment with further novel mitigation techniques. For example, by adding noise to traffic to communications from insecure devices which must communicate with unsafe clouds, we may be able to recover some amount of user privacy while keeping the device running as designed.

TABLE IV
RESULTS AFTER THE INFRASTRUCTURE APPLIED

Device name	Potential vulnerability	CVSS score	Notes	Type of solution
WeMo Insight Switch	Connection to the cloud server is using weak TLSv1.0, STUN protocol is used without any encryption	Medium	All the insecure connection to the official cloud is blocked	General solution: Firewall rules
	Clear HTTP connection with sensitive details during the initial setup stage between the device and the app	Low	The switch is now controlled by Home Assistant Clear HTTP with sensitive details will not be sniffed by the attacker since VLAN is applied	General solution: Home Assistant VLAN
HIVE Hub + Active Light	One of the APIs will reveal the device location in clear HTTP	Low	The insecure API is blocked Replace the officel app with Home Assistant	General solution: Firewall rules Home Assistant
	DNS query not encrypted and can be seen by an adversary	Low	Redirect all DNS queries to 1.1.1.1 instead	General solution: Router setup Firewall rules
Ring Video Doorbell	Firmware updating file sent by the server is using clear HTTP. Injection attack might be possible	Medium	The device does not support traffic encryption for downloading firmware	N/A
	DNS query not encrypted and can be seen by an adversary	Low	Redirect all DNS queries to 1.1.1.1 instead	General solution: Router setup Firewall rules
	Using unencrypted STUN over UDP for multimedia traffic	Medium	The device does not support traffic encryption	N/A
Feed and Go pet feeder	Clear HTTP communication between the device, the app and the cloud. The attacker can viewvideo stream, serial number of the device or the location information.	High	Device does not support traffic encryption	N/A
iKettle 1.0	Default PIN of the device 000000 will never be changed by the app during initialization stage or after finishing the setup process due to a logical fault of the app code	Medium	Decompiled the application and fixed the existing bug of the app. Add functions for generating randomized password and send to the key-sharing database	Customized solution: Change the offical app
	Home Wi-Fi password is stored in the device in plaintext	Medium	This cannot be changed only if the vendor releases a new version of firmware update	N/A
	The device will not testify if the commands received are sent from realible sources. Thus it is possble to flood the port with forged commands	High	Block all the traffic between LAN device and the kettle. Only allow white listed devices to send commands to the kettle	General solution: VLAN Traffic controll
BT Smart Home Cam 100	Port 53 runs an outdated version of dnsmasq, which will cause poteneial system logs leakage	Low	The problem is fixed by the later version of the official firmware	General solution: Firmware updating
	Accessible setup page which might expose router credential, with default username and password ”admin” and ”admin”	Low	The problem is fixed by the later version of the official firmware	General solution: Firmware updating
	Potential user activity leak: the camera will send different number of packets depending on whether motion is detected	Medium	Block the packets which caused the number difference Will lose part of the function. This problem is fixed by a later version of the official firmware	General solution Firmware updating or Packet filtering
LeFun C2 Wireless Camera	The video stream is using real-time transport protocol (RTP) which is not encrypted	Medium	Block the upstream to the official server and save in data in the local server	General solution: Replace the offical server using local server
	DNS query not encrypted and can be seen by an adversary	Low	Redirect all DNS queries to 1.1.1.1 instead	General solution: Router setup Firewall rules
	The local video stored on the SD card is not encrypted	Low	Can not feasibly be fixed	N/A

The Internet Engineering Task Force (IETF) has proposed the idea of manufacturer usage description RFC 8520 [35] which allows IoT vendors to publish their device technical specifications such as the intended communication patterns and endpoints in with the Internet. This could be combined into the framework for configuring firewall policy automatically when a new device is attached to the system.

The framework we designed showed a good promise on mitigating flawed devices. However, this framework is a technical prototype and involve a lot of configuration work. Thus, it is unlikely to be applied to consumers' home directly. Perhaps the most important line of future research will be to enable more pervasive automatic security controls, alongside automatic software and firmware updates.

Acknowledgements: The work reported here was conducted in the undergraduate and MSc project dissertations of the first two authors respectively. We're grateful to the School of Informatics and other sponsors for funding the purchase of the devices used.

REFERENCES

[1] K. Zetter, "Flaw in home security cameras exposes live feeds to hackers," 2012, https://www.wired.com/2012/02/home-cameras-exposed/, [Accessed July, 2019].

[2] F. Shrouf, J. Ordieres, and G. Miragliotta, "Smart factories in industry 4.0: A review of the concept and of energy management approached in production based on the Internet of Things paradigm," in *2014 IEEE International Conference on Industrial Engineering and Engineering Management*, Dec 2014, pp. 697–701.

[3] G. Kambourakis, C. Kolias, and A. Stavrou, "The Mirai botnet and the IoT zombie armies," in *MILCOM 2017 - 2017 IEEE Military Communications Conference (MILCOM)*. IEEE, 2017, pp. 267–272.

[4] GOV.UK, "Secure by design," 2019, https://www.gov.uk/government/collections/secure-by-design, [Accessed July,2019].

[5] U. D. of Homeland Security, "Strategic principles for securing the Internet of Things (IoT)," 2016, https://www.dhs.gov/sites /default/files/publications/Strategic_Principles _for_Securing_the_Internet_of_Things-2016-1115-FINAL_v2-dg11.pdf, [Accessed July,2019].

[6] R. Hackett, "TRENDnet cameras still have gaping security holes, 3 years after FTC settlement," 2017, http://fortune.com/2017/11/15/security-camera-hack-ftc-trendnet-dahua-belkin/, [Accessed March, 2019].

[7] S. Babar, P. Mahalle, A. Stango, N. Prasad, and R. Prasad, "Proposed security model and threat taxonomy for the Internet of Things (IoT)," in *Recent Trends in Network Security and Applications*, N. Meghanathan, S. Boumerdassi, N. Chaki, and D. Nagamalai, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 420–429.

[8] O. Community, "OWASP Internet of Things project," 2018, https://www.owasp.org/index.php/OWASP_Internet_of_Things_Project, [Accessed July, 2019].

[9] WeMo, "Wemo insight switch," 2019, https://www.belkin.com/uk/p/P-F7C029/, [Accessed July, 2019].

[10] Hivehome, "Hive hub," 2019, https://www.hivehome.com/products/hive-hub, [Accessed July, 2019].

[11] RingUK, "Ring video doorbell," 2019, https://en-uk.ring.com/products/video-doorbell, [Accessed July, 2019].

[12] Amazon, "Feed and go smart pet feeder," 2019, https://www.amazon.com/Feed-Go-Feeder-Android-Webcam/dp/B00UNQZ2QI/, [Accessed July, 2019].

[13] Smarter, "ikettle 1.0," 2019, https://support.smarter.am/hc/en-us/categories/115000615949-iKettle-1-0, [Accessed July, 2019].

[14] BTGroup, "Bt smart home cam 100," 2019, https://shop.bt.com/products/bt-smart-home-cam-100-077232-9C5D.html, [Accessed July, 2019].

[15] LeFun, "Lefun wireless camera," 2019, https://www.lefunsmart.com/collections/wireless-cameras/products/lefun-c2-wifi-camera, [Accessed July,2019].

[16] R. Alharbi and D. Aspinall, "An IoT analysis framework: An investigation of IoT smart cameras' vulnerabilities," in *Living in the Internet of Things: Cybersecurity of the IoT - 2018*, March 2018, pp. 1–10.

[17] S. Notra, M. Siddiqi, H. Habibi Gharakheili, V. Sivaraman, and R. Boreli, "An experimental study of security and privacy risks with emerging household appliances," in *2014 IEEE Conference on Communications and Network Security*, Oct 2014, pp. 79–84.

[18] NVD, "CVE-2015-4400," 2019, https://nvd.nist.gov/vuln/detail/CVE-2015-4400, [Accessed July,2019].

[19] K. Munro, "Hacking kettles & extracting plain text wpa psks. yes really!" 2015, https://www.pentestpartners.com/security-blog/hacking-kettles-extracting-plain-text-wpa-psks-yes-really/,[Accessed July, 2019].

[20] M. Schneider, "Belkin WeMo switch communications analysis," 2016, https://www.scip.ch/en/?labs.20160218,[Accessed July, 2019].

[21] Feedandgo, "Feed and Go YouTube channel," 2019, https://www.youtube.com/user/feedandgo/videos,[Accessed July, 2019].

[22] M. Vedomske, "Without device longevity, the Internet of Things will never be," 2019, https://medium.com/achieving-the-grand-vision-of-the-internet-of/without-device-longevity-the-internet-of-things-will-never-be-58c904703abb, [Accessed July,2019].

[23] pfSense, "pfSense," 2019, https://www.pfsense.org/, [Accessed July,2019].

[24] OpenVPN, "OpenVPN," 2019, https://openvpn.net/, [Accessed July,2019].

[25] Home Assistant, "Home Assistant," 2019, https://www.home-assistant.io/, [Accessed July,2019].

[26] V. Martin, Q. Cao, and T. Benson, "Fending off IoT-hunting attacks at home networks," in *Proceedings of the 2Nd Workshop on Cloud-Assisted Networking*, ser. CAN '17. New York, NY, USA: ACM, 2017, pp. 67–72. [Online]. Available: http://doi.acm.org/10.1145/3155921.3160640

[27] Cloudflare, "Cloudflare DNS," 2019, https://1.1.1.1/dns/,[Accessed July, 2019].

[28] IEEE, "Ieee standard for local and metropolitan area networks–media access control (mac) bridges and virtual bridged local area networks," 2011, https://standards.ieee.org/standard/802_1Q-2011.html , [Accessed July, 2019].

[29] W. Ashford, "Orvibo data leak puts security spotlight on IoT back end," 2019, https://www.computerweekly.com/news/252466128/Orvibo-data-leak-puts-security-spotlight-on-IoT-back-end, [Accessed July,2019].

[30] First.org, "CVSS v3.0," 2019, https://www.first.org/cvss/calculator/3.0, [Accessed July, 2019].

[31] O. Alrawi, C. Lever, M. Antonakakis, and F. Monrose, "Sok: Security evaluation of home-based IoT deployments," in *IEEE S&P*, 2019, pp. 208–226.

[32] N. Gupta, V. Naik, and S. Sengupta, "A firewall for Internet of Things," in *2017 9th International Conference on Communication Systems and Networks (COMSNETS)*, Jan 2017, pp. 411–412.

[33] N. Apthorpe, D. Reisman, and N. Feamster, "Closing the blinds: Four strategies for protecting smart home privacy from network observers," *arXiv preprint arXiv:1705.06809*, 2017.

[34] V. Sivaraman, H. H. Gharakheili, A. Vishwanath, R. Boreli, and O. Mehani, "Network-level security and privacy control for smart-home IoT devices," in *2015 IEEE 11th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*, Oct 2015, pp. 163–167.

[35] E. Lear, R. Droms, and D. Romascanu, "Manufacturer Usage Description Specification," RFC 8520, Mar. 2019. [Online]. Available: https://rfc-editor.org/rfc/rfc8520.txt