Zoë Henderson
v00855200
Neil Ernst
SENG401
February 15, 2019

SENG480 A1; Home Assistant

Table of Contents

Introduction

Home Assistant is home automation (or, *domotics)* software, which as a category is an important component of the Internet of Things ecosystem, designed so a user may control, monitor and regulate the devices and functions of their home. The first fully formed smart house, featured in December 1950's Popular Mechanics, was the Push-Button Manor by creator Emil Mathias. This home had a multitude of functions seen as futuristic for its time, with almost everything controllable by a button. Some of these include push button draperies and windows, clocks to turn radios on in the morning and off at night (and intermittently throughout the day), opening his wife's dresser to turn on a lamp, elevators, burglary alarms that pipe sound from the affected area to the bedroom, among more improvements unique for the time (*Railton A, 1950).* Other early examples were moreso exhibits, retro-futuristic spectacles, including the New York World's Fair of 1939's *Futurama*, or Disney and Monsanto's *House of the Future* c 1957.

The first example of a general purpose, networked home automation technology is the X10 protocol, developed by Pico Electronics. From the partnership of Pico and BSR came the venture Accutrac Ltd., to develop a vinyl record changer Accutrac-2000 (X9; X1 through X8 were calculator ICs). The X9 had unique features like non-IR remote controls, relatively uncommon in the mid-1970s. From this came the idea to remotely control appliances and lights through existing AC wiring; by 1978 the technology was released, first sold to RadioShack and Sears (*Rye D., 1999).* Early alternative systems include *Timer* (automation based around rotary and digital timers for predictable events), and an unknown system given the pseudonym "RightSchedule," for Orthodox Jews to observe Sabbath among other functions (*Woodruff, A et. al, 2007).* Since then smart home technology has expanded to encompass a variety of different uses and needs, including helping those with accessibility issues, all of the tasks in Emil's early explorations and more.

The global market volume for Smart Home software and hardware is around $69 billion, with an expected annual growth rate of 20.3% through 2023 (*Statista, 2019).* With the development of new technologies comes many difficulties - in the rush to capture an emerging market, there have been many cases of pervasive insecurity from IoT devices. From relatively simple attacks such as LIXIL's smart toilet, where a hardcoded bluetooth pin would allow attackers to repeatedly flush a toilet (*Trustwave Advisories, 2013*) to 2016's massive Mirai botnet, infecting at a peak 600,000 lightbulbs, security cameras, and other such internet connected devices (who proceeded to DDOS websites and infrastructure) (*Antonakakis et al, 2017),* there is good reason to be

concerned about this technology. In my research, Home Assistant itself appears to have only 1 CVE (CVE-2017-16782), an XSS attack into a persistent notification, and 1 CVE for dependant technology aiohttp (CVE-2018-1000519), a session fixation vulnerability in the “load_session” function for redis storage. Both of these have since been addressed. From my observations, the developers appear to straddle the concerns of security between themselves and the user. That is, by default the software is run in ideal conditions (offline and accessible via a user’s LAN), and they note that functionality like remote access adds inherent complexity with the potential for vulnerability, offering the details and appropriate steps to harden such a system.

For the first Quality Attribute Scenario, I will be considering a case of intrusion detection in the most literal sense, as this is home automation software, with the “ios” component.

For the growth case scenario, I will be investigating components as a factor of growth, and intrusion detection software-side.

Use Case Quality Attribute Scenario

From the main site, Home Assistant states to “put local control and privacy first.” To “allow you to control all your devices without storing any of your data in the cloud,” to “keep track of all the devices in your home, so you don’t have to.” Since this copy is front and centre we can infer that these are business goals. Looking at some of the software’s functionality, there are “Actionable notifications” which allow us to receive push notifications while away when motion is detected, a door is opened, and so forth, and allows us to respond with an associated action such as “sound_alarm.”

Quality Attribute: *Security* ***Refinement*** **:** Intrusion Detection

Scenario **:** A user sets up Home Assistant for their house, and wants to be able alerted when there is suspicious behaviour at home.
Source: A user of Home Assistant away from home, at work.
Stimulus: The source receives a notification that there is motion in the living room, but no one is home.
Response: The user clicks the “sound alarm” button, sending event “ios.notification_action_fired” to be emitted on the Home Assistant event bus.
Response Measure: Alarm is sound, metadata of the action and notification saved.

Growth Case Quality Attribute Scenario

Considering that local and offline control are fundamental design principles, the base software itself is secure and strives to be; however there is still risk, notably in the use cases where a user wishes to access the software from outside of their network (thereby exposing it to the internet), and/or through the use of third party extensions ("components"). Since these components are included in the source code of the software, but disabled by default, they must be considered as a part of Home Assistant architecturally; most of the interesting things that you can do with this software relies on the interoperation of Home Assistant with a user's sensors, cameras, devices &c. So, when considering growth, we must consider the development and integration of these components and their associated vulnerabilities.

From the Shodan report for Home Assistant I generated (*Shodan, 2019)*, there are 1,804 results viewable to the internet of which 12 have exposed Samba shares, 10 Mosquitto instances (the latest release from 7 days ago address 3 CVEs (*Mosquitto, 2019)*, but the software and MQTT protocol have a history of vulnerability and vulnerability from misconfiguration), 9 instances using TLSv1 (which is no longer accepted for PCI compliance after June 30th, 2018 (*PCI Security Standards Council, 2016*), having been affected by BEAST, POODLE, POUND, and 2 expired SSL certificates. At least, there is also a Shodan component, which allows users to query for specific instances.

Growth with respect to this refinement introduces unique challenges and considerations. From the developer's perspective there is a delineation between the core system and components, and an acknowledgement that "due to the lack of resources we are not able to review all of our dependencies and inspect them for malicious behavior, leakage of information or compliance with GDPR," (*Home Assistant)*, which holds the implication that components are not necessarily audited either. I will concede that the case of misconfiguration is solely outside of the responsibility the developer's purview, but the possibility of incorporation of malicious or vulnerable components is a sort of "user-beware"; of course, anyone may read the source, but can they? For this aspect to change would require the developers to come into more funding, but this is a sort of double edge for dealing with open source software.

Quality Attribute: *Security* ***Refinement*** **:** Intrusion Detection

Scenario **:** A user sets up Home Assistant for their business, adding third party components, and wants digital intrusion attempts to be logged.

Source: A user of Home Assistant at work.

Stimulus: The source receives notifications of failed login attempts via fail2ban.

Response: The user sets policy to lock out users after “x” attempts.

Response Measure: A user blocks low-bar automated login attempts, and prompts a more serious audit of the system, finding other insecurity.

Diagram, Key and Explanation

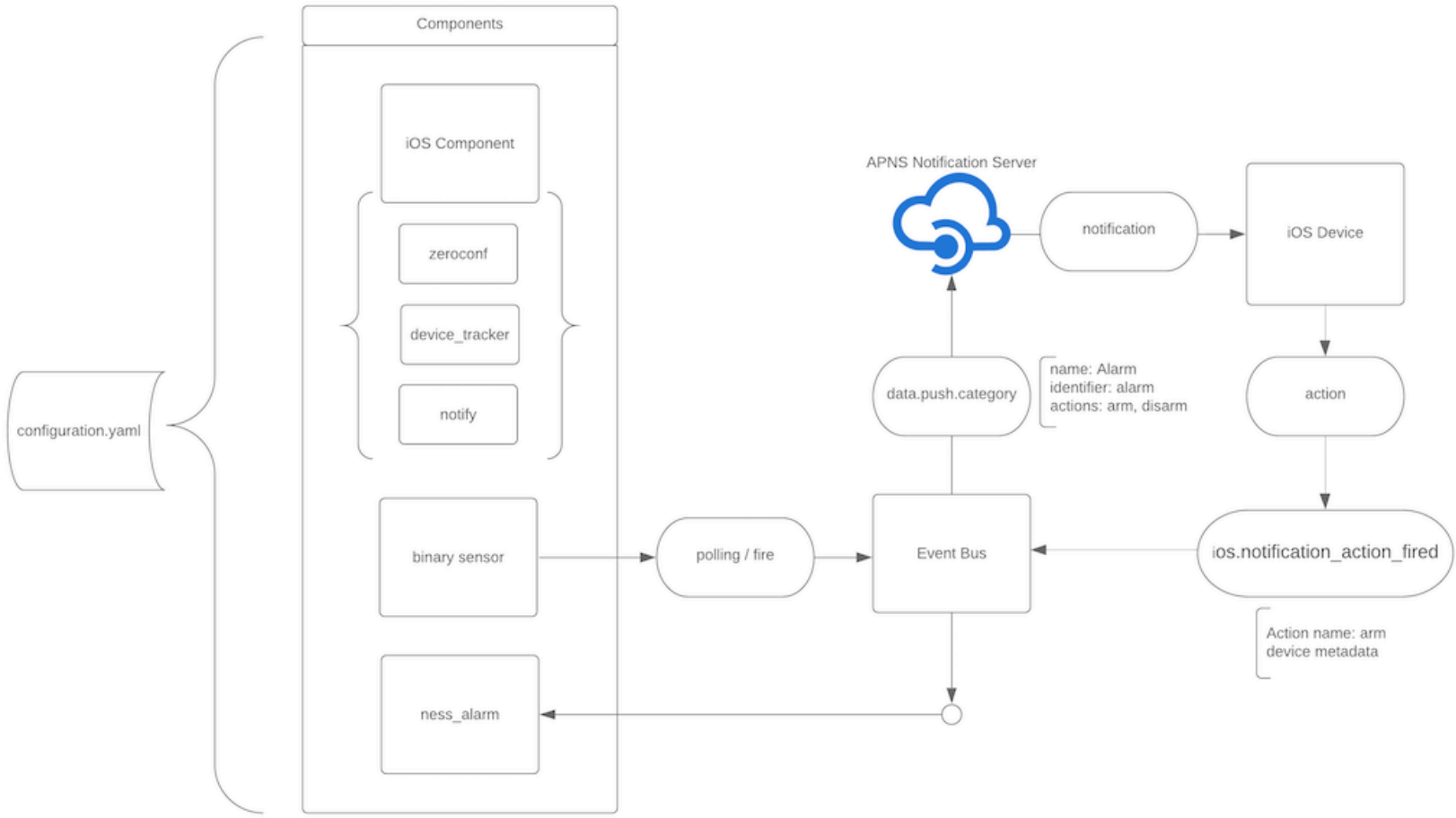

Figure 1, A simplified architecture and state diagram for Home Assistant, looking at the use case "a user's motion sensor is set off, notifying a user over iOS."

The polygon to the left, encompassing “configuration.yaml” is for saved data, angled so as to be seen as “read in” to the components.

The swimmer’s lane adjacent is Home Assistant, abstractly. It holds the components of Home Assistant that make up this scenario.

The rectangles inside the swimmer’s lane are components, as found as subfolders within the source code. The iOS component loads with it zeroconf, device_tracker and notify components, shown underneath paired angle brackets.

Outside of the swimmer’s lane, rectangles are system components; the event bus and user’s iOS device.

Arrows represent the movement of state, data, actions.

Unpaired ovals are actions (“polling / fire, notification, action”). Polling is the event bus’ polling of components for events, and firing off an action if one is received. Notification is the notification as received from APNS. Action is defined as either, “arm the alarm,” or “disregard.”

Ovals paired with text boxes are data sent over wire.

The cloud is Apple’s notification server.

This diagram is a mishmash of view-scales, starting with configuration’s relation to component, a simplified representation of core architecture, and a flow of state through the diagram as a directed graph pertaining to the scenario at hand, with nothing else.

…

“configuration.yaml” $\Longrightarrow$ contains passwords, API tokens, configuration and network information. example fields: api_password, server_port, ssl_certificate, ssl_key, trusted_networks, ip_ban_enabled, login_attempts_threshold

Figure 2, the architecture is on fire. Result of an insecure system open to Shodan.

Sources

Antonakakis, M., April, T., Bailey, M., Bernhard, M., Bursztein, E., Cochran, J., ... & Kumar, D. (2017). Understanding the mirai botnet. In *26th {USENIX} Security Symposium ({USENIX} Security 17)* (pp. 1093-1110).

Home Assistant. Security of Home Assistant. Retrieved from https://www.home-assistant.io/docs/security/

Mosquitto. (2019). Version 1.5.6 release notes. *Mosquitto blog.* Retrieved from: https://mosquitto.org/blog/2019/02/version-1-5-6-released/

Railton, A. (1950, December). Push-Button Manor. In *Popular Mechanics* (pp. 85-88). Retrieved from: https://bit.ly/2EcuVo2

PCI Security Standards Council. (2016, April). Migrating from SSL and Early TLS. Retrieved from: https://www.pcisecuritystandards.org/documents/Migrating-from-SSL-Early-TLS-Info-Supp-v1_1.pdf

Rye, Dave (October 1999). "My Life at X10". *AV and Automation Industry eMagazine*. AV and Automation Industry eMagazine. Retrieved from: https://www.hometoys.com/content.php?url=/htinews/oct99/articles/rye/rye.htm

Shodan. (2019, February). Search for Home Assistant. Generated on: https://www.shodan.io/report/VA8XMUNT

Statista. (2019). Smart Home Market, Worldwide. Retrieved from: https://www.statista.com/outlook/279/100/smart-home/worldwide

Trustwave Advisories. (2013, August). TWSL2013-020: Hard-Coded Bluetooth PIN Vulnerability in LIXIL Satis Toilet. Retrieved from: https://seclists.org/fulldisclosure/2013/Aug/18

Woodruff, A., Augustin, S., & Foucault, B. (2007, April). Sabbath day home automation: it's like mixing technology and religion. In *Proceedings of the SIGCHI conference on Human factors in computing systems* (pp. 527-536). ACM.