IoT-Watchdog

Endbericht | Call 14 | Projekt ID 4430

Inhalt

1 Einleitung

Der IoT-Watchdog hatte coronabedingt, wie viele andere Community-Projekte auch, unter den Einschränkungen und Absagen von Community-Veranstaltungen zu leiden. Der so wichtige persönliche Austausch mit der Community kam dadurch viel zu kurz. Wir freuen uns schon, die versäumten Community-Veranstaltungen im Jahr 2021 nachzuholen!

Für UnravelTEC hatte die Entwicklung des IoT-Watchdogs einen ungeahnten, positiven Nebeneffekt: Die Erkenntnisse aus dem IoT-Watchdog beschleunigten die Entwicklung der in der Corona-Pandemie vor allem von Schulen dringend benötigten CO_2-Ampeln.

Trotz aller Widrigkeiten konnte die geplante Beta-Phase des IoT-Watchdogs noch heuer starten, wir freuen uns auf viele Rückmeldungen!

2 Projektbeschreibung

Das Ziel des Projektes „IoT-Watchdog“ ist es, das Vertrauen in IoT-Geräte zu steigern – und für unvertrauenswürdige Geräte einen Netzwerkschutz bieten.

Verwender von Smartphones, IoT- und anderen „smarten“ Geräten können mit dem IoT-Watchdog die Sicherheit von Geräten und den darauf laufenden Apps im eigenen Netz überprüfen.

Um den IoT-Watchdog einzusetzen, benötigt man einen Raspberry Pi 4 (ein Kleincomputer mit Netzwerkschnittstelle, das einen WLAN-Hotspot öffnen kann). Das IoT-Watchdog Betriebssystem wird aus dem Internet heruntergeladen, und auf der SD-Karte für den Raspberry Pi installiert. Der IoT-Watchdog wird per Netzwerkkabel mit dem Heim-Internet-Router verbunden und öffnet einen WLAN-Hotspot, an den die zu überwachenden IoT-Geräte verbunden werden.

Im Projekt wurde die Software für den IoT-Watchdog soweit entwickelt, um die Beta-Version mit Dezember 2020 zu veröffentlichen. Die Beta-Version kann Netzwerk-Verbindungen die IoT-Geräte ins Internet aufbauen grafisch darstellen, und liefert professionelle Tools zur tiefergehenden Analyse mit.

3 Verlauf der Arbeitspakete

3.1 Arbeitspaket 1 - Backend

Die Recherche zur Tauglichkeit des Netzwerkscanners Wireshark hat folgende Ergebnisse gebracht:

- Die technische Funktionalität ist gegeben, die ersten Ergebnisse des manuellen Tests wurden bereits als Blogpost[1] veröffentlicht.
- Leider ist das eigentlich zur Verwendung geplante Frontend Webshark mit großen Hürden in der Integration in Raspbian verbunden, daher muss wohl auf die c't-Lösung gtk-build + broadwayd zurückgegriffen werden – siehe Blogpost[2].
- Ein neues Tool ist aufgetaucht: ntop-ng[3].

Ntopng verbessert den Analyse-Teil des IoT-Watchdogs signifikant. Es analysiert und klassifiziert ganze Verbindungen. Es unterstützt zwei Datenbanken, in die es die analysierten Verbindungen speichern kann: InfluxDB und MariaDB. Zuerst wurde InfluxDB getestet – hier wurden leider nur die kumulierten Werte aller Server gespeichert, nicht einzelner verbindungen. Schlussendlich wurde MariaDB als Backend verwendet.

Die MariaDB wurde in die Backend-API integriert, welches die Daten aus der MariaDB nun fürs Frontend bereitstellt.

Ein Nachteil von MariaDB ist (im Gegensatz zu InfluxDB) eine geringere Stabilität der Datenbank gegenüber plötzlichem Stromausfall. Da dieser Fall leider bei Raspberry Pi-Systemen zur Regel gehört, musste ein Weg gefunden werden, die bei einem Neustart nach Stromausfall in den meisten Fällen korrupte Datenbank zu resetten. Da jedoch Daten aus früheren Reboots für den Anwendungsfall der Analyse von Verbindungen selten von Wert sind, wird nun die Datenbank einfach bei jedem Reboot resettet.

Für die Hotspot-Funktionalität am Raspberry Pi wurden in der Backend-API die Möglichkeit geschafften, via Frontend die SSID sowie Passwort umzustellen.

Die Konfiguration der Netzwerkschnittstellen wird aus dem Backend via MQTT an das Frontend gepusht – hier wurde der Push-Weg gewählt, um Änderungen an der LAN-Schnittstelle, oder wenn sich ein neues IoT-Gerät via Bluetooth verbindet live im Frontend mitzubekommen.

Im Backend wurde der Teil für die Geolocation der Server-IP-Adressen mithilfe der Maxmind-Datenbank implementiert. Die Maxmind-Datenbank gibt es in mehreren Lizenzvarianten – die freie Variante ist mit einer persönlichen Lizenz versehen, unter der die Datenbank zwar verwendet, aber nicht weitergegeben werden darf. Hier befinden wir uns in einem Dilemma – es gibt leider keine Open Source-Datenbanken zur Geolocation von IP-Adressen, und wir wollen aus Datenschutzgründen keine IP-Adressen-Abfragen an irgendwelche zentralen Dienste wie Google schicken. Wir haben uns dazu entschlossen, dass jeder Nutzer die Maxmind-Datenbank selbst nachinstallieren muss, das ist die einzige 100% rechtskonforme und privacy-kompatible Lösung.

Die restlichen Funktionalitäten der Backend-API wie Neustart, Systemdienstübersicht und -Steuerung sind in PHP implementiert.

---

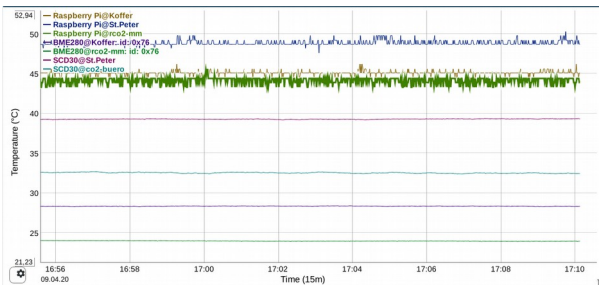
1 https://netidee.at/iot-watchdog/netzwerkanalysen-ziel-android

2 https://www.netidee.at/iot-watchdog/aktuelle-raspian-images-mit-wireshark-bauen

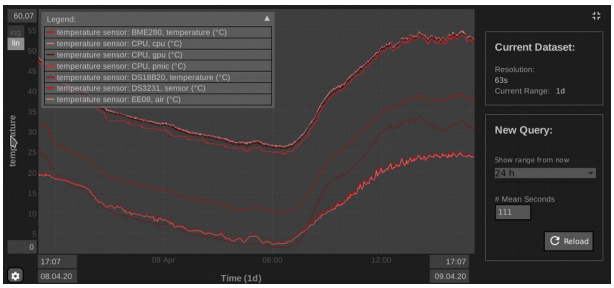
3 https://www.ntop.org/products/traffic-analysis/ntop

3.2 Arbeitspaket 2 - Frontend

Basierend auf dem von UnravelTEC schon für die Visualisierung von Zeitreihen-Sensordaten entwickelten Frontend ng-unrvl wurde ein neues, ansprechenderes Design entwickelt.



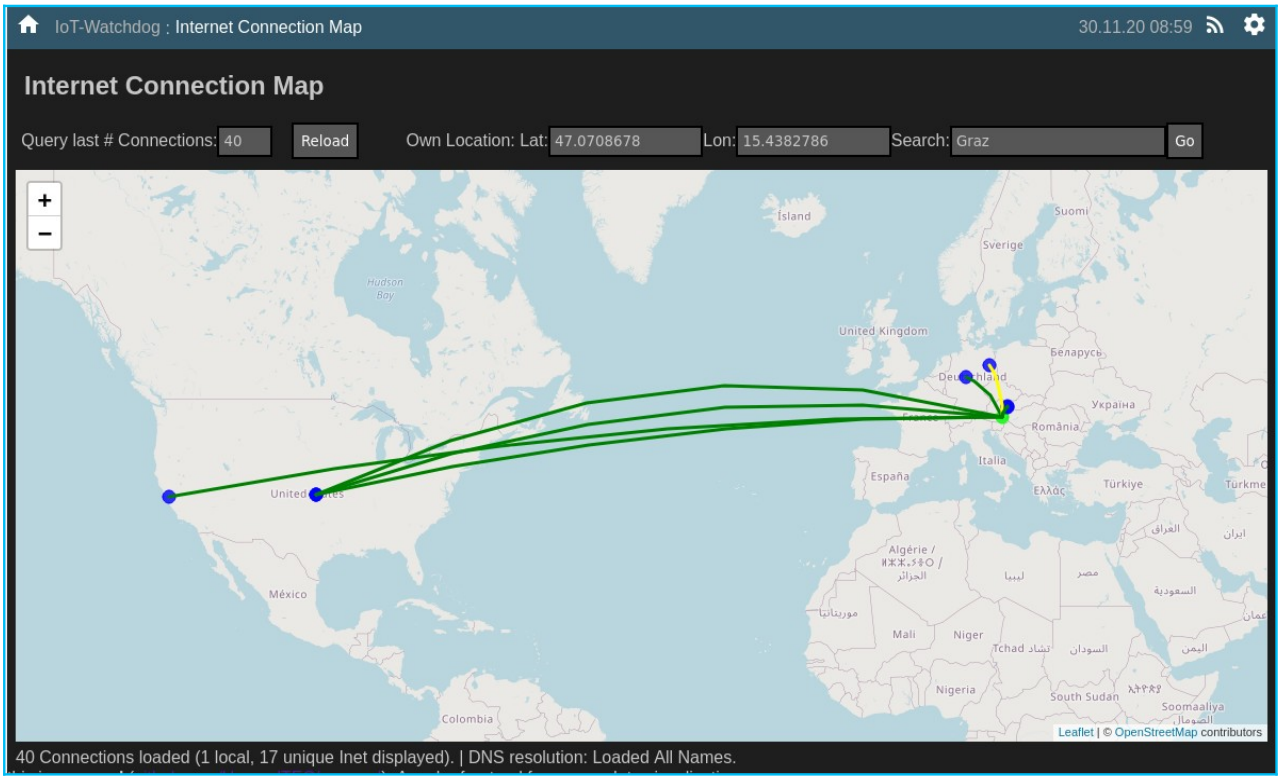
ng-unrvl: Basis-Design 2019



ng-unrvl: neues Design 2020

Für den IoT-Watchdog wurden 3 Apps geschrieben, paketiert und ins image integriert.

1. App: Kartenansicht aller *Internet*-Verbindungen



Hier werden die letzten Verbindungen ins Internet auf einer Karte dargestellt.

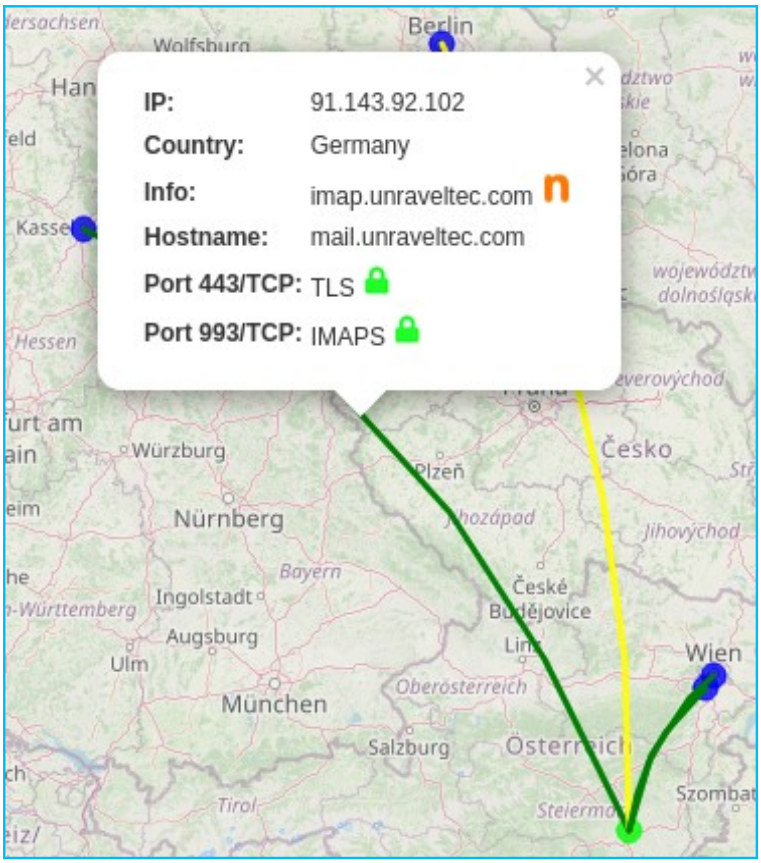
Der eigene Standort wird als Grüner Punkt dargestellt.
Der Zielserver jeder Verbindung wird als blauer Punkt dargestellt. Oft stehen viele Server an einem Punkt, oder werden einem Land (in dem Fall dem Flächenschwerpunkt) zugeordnet.

Die Verbindungen werden als Linien dargestellt, wobei die Farbe einen ersten Blick auf die Abhörsicherheit der Verbindungen zu diesem Server (Über die Art der Verschlüsselung) erlaubt:

- Grün: Die Verbindung ist verschlüsselt (HTTPS, TLS etc.)
- Gelb: Fraglich - Es werden Ports verwendet, wo verschlüsselter oder unverschlüsselt Verkehr möglich ist
- Rot: Verbindungen über unverschlüsselte Protokolle (wie zB HTTP über Port 80)

In einer Linie werden alle Verbindungen zu diesem Server aggregiert - mehrere Linien zu einem Punkt bedeuten Verbindungen zu unterschiedlichen Servern in diesem Gebiet.

Ein Klick auf die Linien öffnet ein Popup mit einer Übersicht auf die Verbindungen zu diesem Server.



In dem Popup sieht man zB, dass zu diesem Server mehrere Verbindungen (auf Port 443, 993) aufgebaut wurden. Detail-Statistiken zu diesem Server können mit einem Klick aufs orange "n" in ntopng geöffnet werden.

2. App: Detail-Ansicht aller Internet-Verbindungen, Liste

Die Listenansicht bietet eine tabellarische Übersicht über die offenen Verbindungen, zeitlich sortiert - die zuletzt aktiven oben.

Hosts overview

Query last # Connections: 50 Reload

Internet Connections

	Source IP	Dest. IP	Info / Hostname	Remote Location	L4 Prot	Src. Port	Dest. Port	L7 Protocol	Last Transfer	Duration (s)	In	Out
🔒	192.168.42.50	129.27.2.234	sbox.tugraz.at	AT	TCP	43187	993	IMAPS	30.11.20 11:00:14	300	6.3 KB	27.8 KB
🔓	192.168.42.50	45.33.7.16	www.httpvshttps.com/	US, Richardson, Texas	TCP	38083	80	HTTP	30.11.20 11:00:01	23	154.6 KB	780.4 KB
🔒	192.168.42.50	76.223.92.165	textsecure-service.whispersystems.org	US	TCP	40279	443	TLS	30.11.20 10:59:33	138	16.5 KB	24.8 KB
🔒	192.168.42.50	142.250.74.202	fra24s02-in-f10.1e100.net	US	TCP	44871	443	TLS.Google	30.11.20 10:59:08	0	156B	54B
🔒	192.168.42.50	172.217.22.42	fra15s16-in-f42.1e100.net	US	TCP	49884	443	TLS.Google	30.11.20 10:59:03	0	326B	108B
🔒	74.125.28.188	192.168.42.50	pc-in-f188.1e100.net	US	TCP	5228	41871	TLS.Google	30.11.20 10:58:33	3	2.3 KB	944B
🔒	192.168.42.50	142.250.74.202	android.googleapis.com	US	TCP	44871	443	G-DNS	30.11.20 10:58:09	241	1.7 KB	5 KB
🔒	192.168.42.50	172.217.22.42	play.googleapis.com	US	TCP	49884	443	G-DNS	30.11.20 10:58:03	241	6.8 KB	3.4 KB
🔒	91.143.92.102	192.168.42.50	mail.unraveltec.com	DE	TCP	993	49102	IMAPS	30.11.20 10:57:23	15	37.2 KB	3.3 KB
🔒	192.168.42.50	80.241.60.199	imap.mailbox.org	DE, Berlin, Land Berlin	TCP	42914	143	IMAPS	30.11.20 10:56:45	334	1.1 KB	1.2 KB
🔒	192.168.42.50	91.143.92.102	imap.unraveltec.com	DE	TCP	49142	993	IMAPS	30.11.20 10:56:44	336	2.6 KB	12.6 KB
?	80.241.60.199	192.168.42.50	imap.mailbox.org	DE, Berlin, Land Berlin	TCP	143	42841	IMAP	30.11.20 10:56:26	0	103B	66B
?	80.241.60.199	192.168.42.50	imap.mailbox.org	DE, Berlin, Land Berlin	TCP	143	42840	IMAP	30.11.20 10:56:26	0	103B	66B
?	80.241.60.199	192.168.42.50	imap.mailbox.org	DE, Berlin, Land Berlin	TCP	143	42836	IMAP	30.11.20 10:56:25	0	103B	66B
?	80.241.60.199	192.168.42.50	imap.mailbox.org	DE, Berlin, Land Berlin	TCP	143	42838	IMAP	30.11.20 10:56:25	0	103B	66B
🔒	192.168.42.50	91.143.92.102	cloud.unraveltec.com	DE	TCP	44510	443	TLS	30.11.20 10:55:37	8	9.6 KB	23.5 KB
🔒	192.168.42.50	216.58.208.42	fra15s12-in-f10.1e100.net	US, Mountain View, California	TCP	46893	443	TLS.Google	30.11.20 10:55:21	2	742B	162B
🔒	192.168.42.50	216.58.208.42	www.googleapis.com	US, Mountain View, California	TCP	46895	443	G-DNS	30.11.20 10:54:20	240	19.8 KB	53.2 KB
🔒	192.168.42.50	192.229.233.50	cdn.syndication.twimg.com	US	TCP	39732	443	DNS.Twitter	30.11.20 10:54:05	171	2.6 KB	7.2 KB
🔒	192.168.42.50	104.244.42.72	syndication.twitter.com	US	TCP	45955	443	DNS.Twitter	30.11.20 10:54:04	171	3 KB	5.5 KB
🔒	192.168.42.50	192.229.233.25	platform.twitter.com	US	TCP	45676	443	DNS.Twitter	30.11.20 10:53:02	109	2.8 KB	6.9 KB

50 Connections loaded (3 local, 21 unique Inet displayed). | DNS resolution: Loaded All Names.

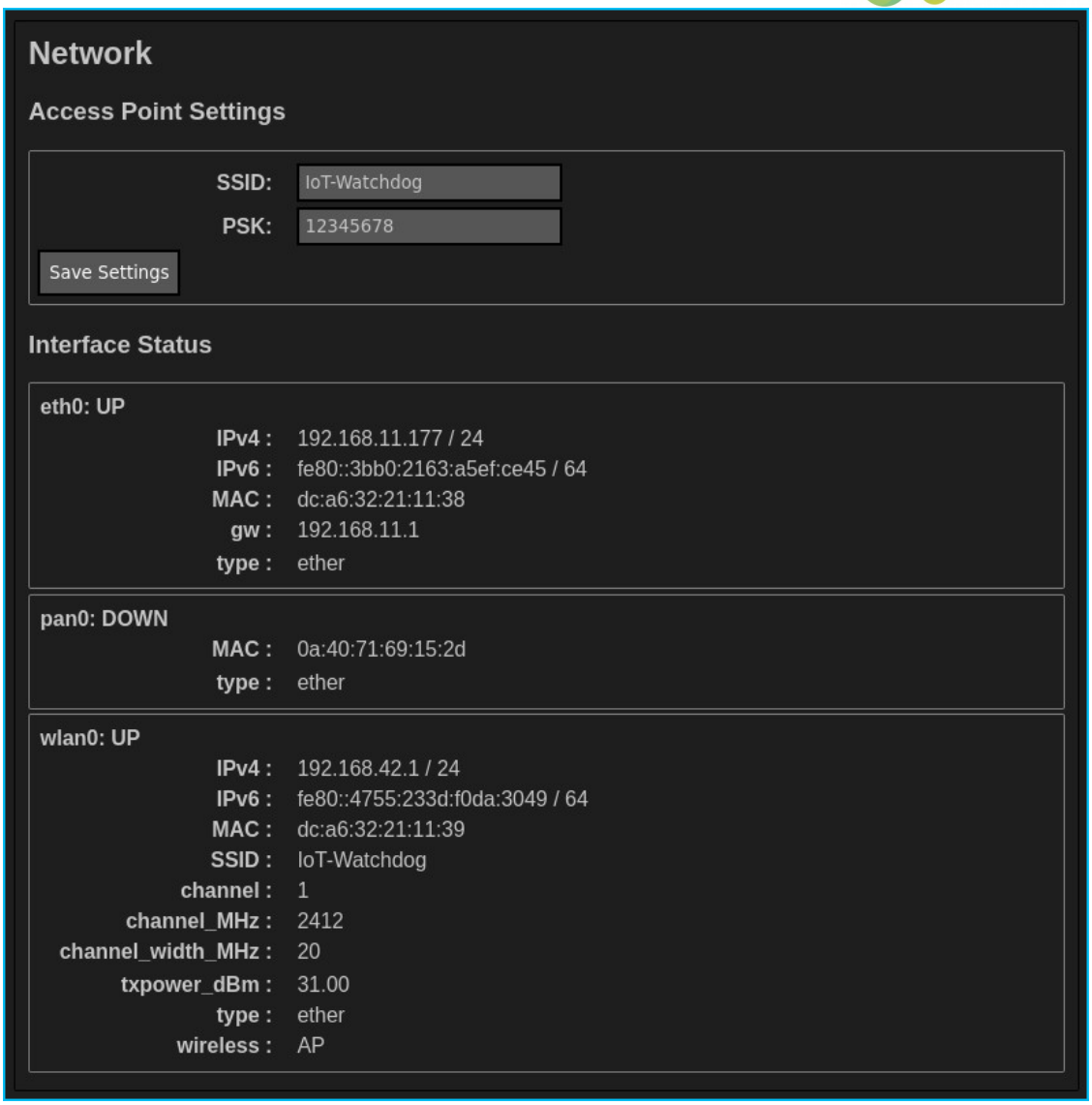
Erklärung der Spalten:

- Das bunte Symbol links zeigt die Abhörsicherheit der Verbindungen (Abgeleitet von den verwendeten Protokollen/Ports) an.
- *Source IP* und *Dest. IP* sind die IP-Adressen von Start und Ziel der Verbindung. Die mit 192.168. beginnende ist dabei diejenige vom Gerät im lokalen Netzwerk des IoT-Watchdogs.
- Mit Klick aufs orange "n" können Details des Zielservers via ntopng geöffnet werden.
- *Info/Hostname* zeigt den Namen des Servers im Internet an.
- *Remote Location* zeigt den Standort (Land oder wenn bekannt Stadt) des Servers im Internet an.
- *L4 Prot* zeigt das grundlegende Netzwerkprotokoll (meist TCP oder UDP) an.
- *Src. Port* und *Dest. Port* geben Quell- und Zielport der Verbindung an.
- *L7 Protocol* gibt das Protokoll auf Anwendungsebene an. Die meisten Verbindungen laufen heutzutage verschlüsselt über TLS oder HTTPS ab.

3. App: „Local Connection List“,

zeigt ähnlich der Internet-Verbindungs-Liste die Verbindungen an, die die Endgeräte ins Lokale Netzwerk (LAN) aufbauen.

Im Frontend wird auch die Konfiguration des IoT-Watchdogs ermöglicht, ein Ausschnitt aus den „Settings“:



3.3 Arbeitspaket 3 - Updates/Paketsystem

Die Toolchain zum bauen von Debian-Paketen für Updates wurde erfolgreich eingerichtet, siehe Blogpost[4].

Auf GitHub wurde in Kombination mit dem Blogpost ein Beispielpaket veröffentlicht:

https://github.com/IoT-Watchdog/Example-Deb-pkg

Die folgenden Pakete wurden als Debian-Paket paketiert:

- ng-unrvl-iotw (Frontend)
- lcars-api (Backend)
- mqtt2influx (Systemüberwachngsdaten in Influx-Datenbankformat konvertieren)

Die anderen verwendeten Pakete wie wireshark oder ntopng sowie die für den IoT-Watchdog angepasste Konfiguration wurden direkt ins image integriert.

4 https://www.netidee.at/iot-watchdog/debian-pakete-fuer-raspbian-bauen

3.4 Arbeitspaket 4 – Images

Seit Anfang 2020 läuft die Integration der nötigen Pakete in pi-gen. Die build-Toolchain wurde inzwischen erfolgreich von Debian Stretch (9) auf die neue Version Buster (10) umgestellt, und alle daraus resultierenden Bugs behoben.

Inzwischen können erfolgreich angepasste images für den Raspberry Pi Zero sowie den Raspberry Pi 4 gebaut werden.

Wie funktioniert unser build-system?

Aufbauend auf pi-gen werden weitere Stufen (stages) eingebaut. Unser repository nutzt das originale pi-gen als git submodule und verlinkt die originalen Stufen.

Die von uns aufgesetzte weitere Stufe (4) ändert folgende Dinge:

- Hinzufügen der offiziellen InfluxDB-Paketquellen und installieren von InfluxDB
- hinzufügen und installieren der .debs aus AP3
- Setzen eines Passworts für den pi-user aus Sicherheitsgründen
- Setzen der Zeitzone Mitteleuropa/Wien
- Unbenutzte Services wie zB Audio, den alten rsyslog, nfs deaktivieren
- Umstellen des Netzwerkstacks auf NetworkManager
- Installieren von Network-Dispatcher und skripten, die Netzwerkänderungen via MQTT ans Frontend weitergeben
- Bluetooth aktivieren und Bluetooth-Netzwerk konfigurieren
- Installieren eines Services zum resetten der MariaDB bei boot
- hostname setzen
- Webserver Apache konfigurieren und nötige php-module aktivieren
- diverse configfiles für apache, InfluxDB, MQTT-broker, MariaDB, ntopng

Die verwendenten Backend-Tools ntopng und Wireshark werden in der sub-Stufe „02-raspion“ mit angepasster Konfiguration installiert. Die ntopng-Konfiguration wird dabei als redis-Datenbank während der Installation eingespielt.

3.5 Arbeitspaket 5 – Doku

1. Entwickler-Doku: Für die Repositories der Einzelteile (Distribution, Frontend, Paketbau-System) ist die Entwickler-Dokumentation in den jeweiligen Repositories als README-Datei veröffentlicht:

   - https://github.com/IoT-Watchdog/pi-gen-iot-watchdog,
   - https://github.com/IoT-Watchdog/ng-unrvl,
   - https://github.com/IoT-Watchdog/Example-Deb-pkg

2. Anwender-Doku: Die Bedienungsanleitung befindet sich als Markdown direkt im README des Repositories, und zeigt mittels Screenshots die Bedienung der Beta-Version. Siehe:

- https://github.com/IoT-Watchdog/documentation

3.6 Arbeitspaket 6 – Tests

Alpha-Tests wurden im 1:1-Betreuungsverhältnis durch das IoT-Watchdog-Teams bei einzelnen Test-“Kunden“ durchgeführt – durch die Covid19-Pandemie leider nicht im geplanten Umfang.

Die öffentliche Beta-Phase startet Anfang Dezember 2020.

Für Bug-Reports und Feature-Requests wurden folgende Anlaufstellen eingerichtet:

- https://github.com/IoT-Watchdog/ng-unrvl/issues
- https://github.com/IoT-Watchdog/pi-gen-iot-watchdog/issues
- https://github.com/IoT-Watchdog/Example-Deb-pkg/issues

Von dem in der Beta-Testphase geplante Rückmelden von Verbindungs-Metadaten ans Entwicklerteam musste leider aus Datenschutzgründen abgesehen werden – nach eingehenden Diskussion konnte leider kein Weg gefunden werden, der Datenschutzrechtlich in Ordnung wäre, um Metadaten zentral in einer Datenbank für uns Entwickler zu übertragen/speichern, ohne dass dabei auf einzelne Nutzer rückgeschlossen werden könnte.

3.7 Arbeitspaket 7 – Marketing

Aufgrund der Corona-Pandemie wurden leider alle Open Source Community-Veranstaltungen, auf denen der IoT-Watchdogs präsentiert werden sollte abgesagt:

- Grazer Linuxtage 2020
- Maker Faire Wien 2020
- Wiener Linuxwochen

Auf der Netidee-Projektseite wurden 6 Blogposts veröffentlicht, die besondere Herausforderungen / Projektergebnisse präsentieren. Die Blogeinträge wurden auf Social Media geteilt.

Auf der UnravelTEC.com–Webseite wurde eine Projekt-Seite für den IoT-Watchdog eingerichtet.

Das Beta-Release des IoT-Watchdogs soll bei der verschobenen Cryptoparty Graz am 11.1.2020 präsentiert werden.

Bei den Grazer Linuxtagen 2021 ist geplant, den IoT-Watchdog-Stand aus Jahr 2020 nachzuholen.

4 Umsetzung Förderauflagen

1. Eigenmittel von 15% - Der Eigenanteil wurde aus dem laufenden Umsatz der UnravelTEC OG finanziert.

2. Die 3 Testgeräte (Raspberry Pi 4) für den Förderbeirat wurden mit „IoT-Watchdog“ 3D-Druck-Gehäuse, Speicherkarte mit IoT-Watchdog-Image sowie Enduser-Anleitung verschickt.

5 Liste Projektendergebnisse

1	**Projektzwischenbericht**	CC-BY-3.0 AT	www.netidee.at/iot-watchdog
2	**Projektendbericht**	CC-BY-3.0 AT	www.netidee.at/iot-watchdog
3	**Entwickler_innen-DOKUMENTATION:** im README jedes Repositories	GPL-3, BSD-3	github.com/IoT-Watchdog/
4	**Anwender_innen-DOKUMENTATION** "Bedienungsanleitung" Sie befindet sich als Markdown direkt im README des Repositories, und zeigt mittels Screenshots die Bedienung der Beta-Version.	CC-BY-3.0 AT	github.com/IoT-Watchdog/documentation
5	**Veröffentlichungsfähiger Einseiter**	CC-BY-3.0 AT	www.netidee.at/iot-watchdog
6	**Dokumentation Externkommunikation** * Auf der Netidee-Projektseite wurden 6 Blogposts veröffentlicht, die besondere Herausforderungen / Projektergebnisse präsentieren. * Die Blogeinträge wurden auf Social Media geteilt * Auf der UnravelTEC.com – Webseite wurde eine Projekt-Seite für den IoT-Watchdog eingerichtet. Aufgrund der Corona-Pandemie wurden leider alle Open Source Community-Veranstaltungen, auf denen die Beta-Version des IoT-Watchdogs präsentiert werden sollte abgesagt, wie zB: * Grazer Linuxtage 2020 (2021 nachholen) * Maker Faire Wien 2020 * Wiener Linuxwochen Das Beta-Release des IoT-Watchdogs soll bei der verschobenen Cryptoparty Graz am 11.1.2020 präsentiert werden.	CC-BY-3.0 AT	www.netidee.at/iot-watchdog unraveltec.com/iot-watchdog/
7	**Backend** ***Backend-Pakete sind ins image integiert.*** * API für Frontend * Netzwerkanalyse-Tools: wireshark, ntopng ins image integriert * Firewall-backend iptables	BSD-3	https://github.com/IoT-Watchdog/pi-gen-iot-watchdog
8	**Frontend** * Zugriff auf alle Funktionen via Webfrontend in Angular * Visualisierung aller Verbindungen ins Internet als	GPL-3.0	https://github.com/IoT-Watchdog/ng-unrvl

	Kartenansicht * Visualisierung aller Netzwerkverbindungen (Internet und Lokal) als Listenansicht * Regeln/Blockieren von Verbindungen und Geräten via Easywall umgesetzt		
9	**Debian-Paketierung** * Beispiel-Repository zum Bau eines Debian-Paketes Folgende Pakete werden derzeit paketiert: * ng-unrvl-iotw (Frontend) * lcars-api (Backend) * mqtt2influx (Systemüberwachungsdaten in Influx-Datenbankformat konvertieren)	GPL-3.0	https://github.com/IoT-Watchdog/Example-Deb-pkg
10	**IoT-Watchdog-Distribution** * Die IoT-Watchdog Distribution basierend auf Raspbian * Verwendung von pi-gen sorgt für die Generierung von aktuellen Images fertig zum Download	BSD-3	https://github.com/IoT-Watchdog/pi-gen-iot-watchdog

6 Verwertung der Projektergebnisse in der Praxis

Das parallel zum IoT-Watchdog laufenden UnravelTEC-Projekt „CO_2-Ampel“ hat von den Errungenschaften im IoT-Watchdog profitiert.

Die CO_2-Ampel ist ein kleines Gerät, das beim richtigen Lüften hilft – Sie wurde inmitten der Corona-Pandemie auf schnellstem Wege entwickelt, um insbesondere in Schulen zu helfen, das Infektionsrisiko in den Innenräumen zu senken.

Insbesondere die im IoT-Watchdog gewonnenen Erkenntnisse mit pi-gen zum schnellen Bau von Images halfen uns bei der Prototypen-Entwicklung und Kleinserien-Fertigung der CO_2-Ampeln.

Die im IoT-Watchdog ständig mitlaufenden Sicherheitsüberlegungen haben unseren Blick für die Sicherheit von IoT-Geräten geschärft, und helfen uns, die IoT-Geräte in unserem Produktportfolion besonders sicher und Datenschutzkonform zu gestalten.

7 Öffentlichkeitsarbeit / Vernetzung

Aufgrund der Corona-Pandemie wurden leider alle Open Source Community-Veranstaltungen, auf denen der IoT-Watchdogs präsentiert werden sollte abgesagt:

- Grazer Linuxtage 2020
- Maker Faire Wien 2020

- Wiener Linuxwochen

Wir wollen die Beta-Release des IoT-Watchdogs bei der verschobenen Cryptoparty Graz im Jänner 2021 präsentieren.

Bei den Grazer Linuxtagen 2021 ist geplant, den IoT-Watchdog-Stand aus Jahr 2020 nachzuholen.

8 Eigene Projektwebsite

Eine eigene Projektwebsite wird nicht betrieben.

Auf der UnravelTEC-Webseite wird das Projekt gezeigt:
https://www.unraveltec.com/iot-watchdog/

Als Open Source-Anlaufpunkt dient die Organisation auf Github:

https://github.com/IoT-Watchdog

9 Geplante Aktivitäten nach netidee-Projektende

Das Feedback aus der Betatest-Phase soll sukzessive eingearbeitet werden.

Wenn im nächsten Jahr wieder Community-Veranstaltungen stattfinden können, soll dort nach neuen Contributoren im Open Source-Umfeld für den IoT-Watchdog gesucht werden.

10 Anregungen für Weiterentwicklungen durch Dritte

Als sinnvolle Weiterentwicklungen würden wir sehen:

- VLAN-Funktionalität im WLAN – so dass mehrere verbundene Client-Geräte sich gegenseitig nicht sehen/erreichen können
- Integration von Pi-Hole – der Werbungs-Blocker war optional im Projekt geplant, und konnte aus Zeitgründen nicht umgesetzt werden.
- Verbindungs-Anzeige: Einführen von „Allowlists“, um als unbedenklich bekannte Verbindungen ausblenden zu können, und nur neue fragliche Verbindungen anzuzeigen.
- Verbindungs-Datenbank: MariaDB durch etwas gegen Korruption durch Stromausfall robusteres ersetzen/erweitern.