

Cotopaxi

IoT protocols
testing toolkit



About the author

Jakub Botwicz

- Principal Security Engineer
at Samsung R&D Institute in Warsaw, Poland
- Leads a team of security researchers / pentesters
- PhD and MSc at Warsaw University of Technology
- 16+ years experience - previously worked as:
 - Developer/architect for vendor of encryption devices
 - Security advisor at credit card payment company
 - Security consultant and manager at Big4 company
- Over 35 CVEs reported in recent 3 years
- Big enthusiast of rock climbing and active volcanoes



Cotopaxi – contributors

- Anonymous Cat
- liyansong2018
- Mariusz Księżak
- Michał Radwański
- Wiktor Gerstenstein



Source:

<https://twitter.com/mmatp/status/1151851535439159299/photo/1>

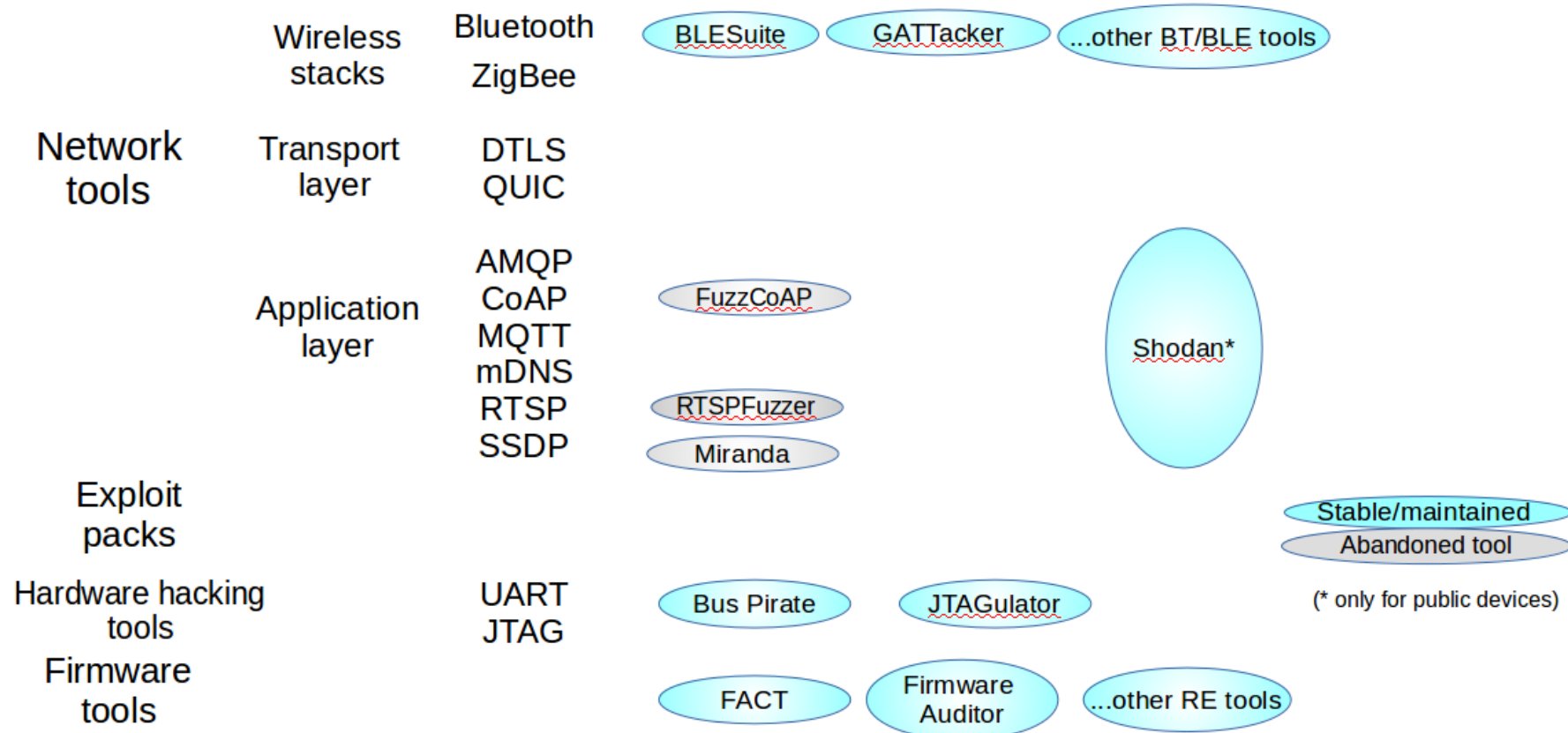
Cotopaxi – origin / idea

- IoT introduced **new protocols**: CoAP, DTLS, MQTT and refurbished old protocols: UPnP, SSDP
- Lack of **security testing tools for IoT protocols** (except for MQTT)
- Low level of **security measures** of IoT components and devices
- Our team performed assessment of multiple IoT components – results:
 - ideas for tools
 - corpus of malformed messages
 - 20+ new vulnerabilities

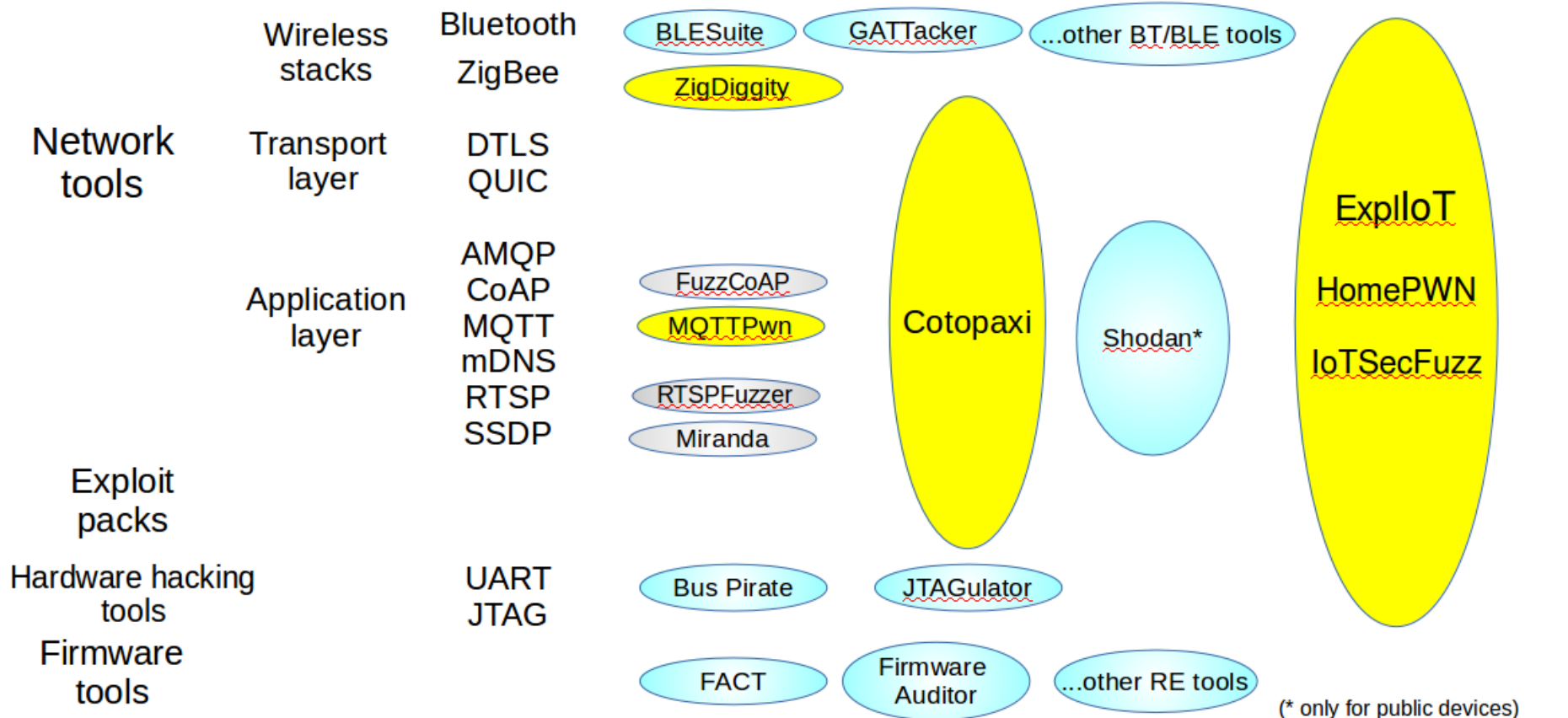


Source: Redbubble sticker:
<https://ih1.redbubble.net/image.534691174.7627/sticker,375x360-bg,ffffff.u3.png>

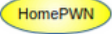
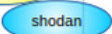
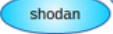
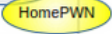
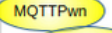
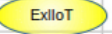
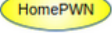
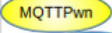
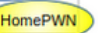
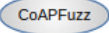
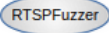
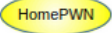
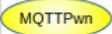
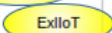
IoT security tool – landscape (end of 2017)

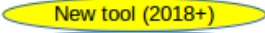


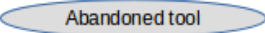
IoT security tool – landscape (end of 2019)



IoT protocols supported by free IoT tools

Tool	AMQP	CoAP	DTLS	HTCPCP	mDNS	MQTT	QUIC	RTSP	SSDP
service_ping		 				 			 
server_fingerprinter									
credential_cracker						 			
resource_listing		 				 			 
protocol_fuzzer									
client_proto_fuzzer									
vulnerability_tester						 			
client_vuln_tester									
amplifier_detector									
active_scanner									





nmap scan on CoAP / DTLS servers

```
sudo nmap 1xx.1xx.1xx.1xx -sU -p 10001-10005,20000-20010 -A
```

Starting Nmap 7.70 (<https://nmap.org>) at 2019-09-09 09:09 CET
Nmap scan report for 1xx.1xx.1xx.1xx
Host is up (0.00059s latency).

PORT	STATE	SERVICE	VERSION
10001/udp	open filtered	scp-config	
10002/udp	open	documentum?	
10003/udp	open	documentum_s?	
10004/udp	open	unknown	
10005/udp	open	unknown	
20000/udp	open filtered	dnsp	
20001/udp	open filtered	microsan	
20002/udp	open	commtact-http	
20003/udp	open	commtact-https	
20005/udp	open filtered	openwebnet	
20006/udp	open filtered	unknown	



Source: Bilal Farooqui, Independent, <https://www.independent.co.uk/news/world/americas/suicidal-robot-did-not-kill-itself-fountain-accident-loose-brick-surface-a7866856.html>

Wireshark view on CoAP traffic

62	108.72...	106.1...	106.1...	UDP	54	48443	→	10002	Len=12
63	111.35...	106.1...	106.1...	UDP	54	48443	→	10002	Len=12
64	118.01...	106.1...	106.1...	UDP	54	57186	→	10003	Len=12
65	120.32...	106.1...	106.1...	UDP	54	57186	→	10003	Len=12
70	132.58...	106.1...	106.1...	UDP	54	49034	→	10001	Len=12
71	135.20...	106.1...	106.1...	UDP	54	49034	→	10001	Len=12
80	144.32...	106.1...	106.1...	UDP	54	60966	→	10001	Len=12
81	147.26...	106.1...	106.1...	UDP	54	60966	→	10001	Len=12
86	157.34...	106.1...	106.1...	UDP	54	45486	→	1001	Len=12
87	162.27...	106.1...	106.1...	UDP	54	34418	→	10001	Len=12
88	166.10...	106.1...	106.1...	UDP	54	44498	→	10101	Len=12

▶ User Datagram Protocol, Src Port: 48443, Dst Port: 10002

▼ Data (12 bytes)

Data: 4001736a7227124474657374

[Length: 12]

0000	00 08 e3 ff fd 90 50 65 f3 2a 28 5b 08 00 45 00	Pe.*([...E.
0010	00 28 3b 89 40 00 40 11 e9 a2 6a 78 b8 11 6a 78	.(;@.@. .jx..jx
0020	88 97 bd 3b 27 12 00 14 15 bf 40 01 73 6a 72 27	...;'... ..@.sjr'
0030	12 44 74 65 73 74	.Dtest

Wireshark view on CoAP traffic – “Decode As” CoAP

62 108.72... 106.1... 106.1... CoAP		54 CON, MID:29546, GET, :10002/test
63 111.35... 106.1... 106.1... CoAP		54 CON, MID:29546, GET, :10002/test
▶ User Datagram Protocol, Src Port: 48443, Dst Port: 10002		
▼ Constrained Application Protocol, Confirmable, GET, MID:29546		
01.. = Version: 1		
..00 = Type: Confirmable (0)		
.... 0000 = Token Length: 0		
Code: GET (1)		
Message ID: 29546		
▼ Opt Name: #1: Uri-Port: 10002		
Opt Desc: Type 7, Critical, Unsafe		
0111 = Opt Delta: 7		
.... 0010 = Opt Length: 2		
Uri-Port: 10002		
▼ Opt Name: #2: Uri-Path: test		
Opt Desc: Type 11, Critical, Unsafe		
0100 = Opt Delta: 4		
.... 0100 = Opt Length: 4		
Uri-Path: test		
[Uri-Path: :10002/test]		
0000	00 08 e3 ff fd 90 50 65 f3 2a 28 5b 08 00 45 00	Pe .*([.E.
0010	00 28 39 fd 40 00 40 11 eb 2e 6a 78 b8 11 6a 78	.(9.@.@. .jx.jx
0020	88 97 bd 3b 27 12 00 14 15 bf 40 01 73 6a 72 27	...;'.... @.sjr'
0030	12 44 74 65 73 74	.Dtest

Cotopaxi – origin / idea

- IoT introduced **new protocols**: CoAP, DTLS, MQTT and refurbished old protocols: UPnP, SSDP
- Lack of **security testing tools for IoT protocols** (except for MQTT)
- Low level of **security measures** of IoT components and devices
- Our team performed assessment of multiple IoT components – results:
 - ideas for tools
 - corpus of malformed messages
 - 20+ new vulnerabilities



Source: Redbubble sticker:
<https://ih1.redbubble.net/image.534691174.7627/sticker,375x360-bg,ffffff.u3.png>

Cotopaxi – the toolkit

- Set of tools for security testing of Internet of Things devices using network IoT protocols
- License: GPL-2.0
- Repository:
<https://github.com/samsung/cotopaxi>
- Releases:
 - 1st release in March 2019 (before Black Hat Asia)
 - **4th release in August 2020 (before Black Hat USA and DEFCON)**



Source: <https://www.amazon.com/Wenger-16999-Swiss-Knife-Giant/dp/B001DZTJRQ>

Cotopaxi – the name

- Active stratovolcano in Ecuador
- Elevation:
19,347 ft / 5,897 m



Author: [Gerard Prins](#) [CC BY-SA 3.0](#)

Cotopaxi – usage scenarios

- For **pentesters** to:
 - analyze environments using IoT, Smart-{Home, Factory, City}
 - find active endpoints using IoT protocols
 - classify software components or IoT devices
 - identify network traffic reflectors (DDoS)
- For **security researchers** to:
 - perform „black box” testing of IoT devices
 - identify known security vulnerabilities
 - identify OEM devices (by classification)
- For **developers** or **vendors** of IoT devices to:
 - fuzz components or interfaces
 - test traffic amplification (DDoS)



Cotopaxi – features in 4th release

Reconnaissance phase:

- **Service ping**
 - checking availability of network services
- **Security scanner** – verifying security settings (e.g. supported ciphersuites, certificates)
- **Software fingerprinting** – recognizing the software used by remote network server
- **Resource listing „dirbusting“**
 - discovering resources identified by given URLs
- **Device identification** – passive analysis of traffic and device classification using Machine Learning



Source: <https://www.amazon.com/Wenger-16999-Swiss-Knife-Giant/dp/B001DZTJRQ>

Pre-exploitation phase:

- **Amplification sniffing** – detecting network traffic amplification
- **Protocol fuzzing** – fuzzing implementation of protocol
 - From 2nd release for protocol clients and servers
- **Vulnerability testing** – identifying known vulnerabilities
 - From 2nd release for protocol clients and servers

Cotopaxi – supported IoT protocols

- **Supported in 1st release:**

- Constrained Application Protocol (CoAP)
- Datagram Transport Layer Security (DTLS)
- Multicast DNS (mDNS) / DNS-System Discovery (DNS-SD)
- Message Queuing Telemetry Transport (MQTT)

- **Added in 2nd and 3rd releases:**

- Hyper Text Coffee Pot Control Protocol (HTCPCP)
- Simple Service Discovery Protocol (SSDP)
- Real Time Streaming Protocol (RTSP)

- **Added in 4th release:**

- Advanced Message Queuing Protocol (AMQP)
- MQTT-Sensor Networks (MQTT-SN)
- Quick UDP Internet Connections (QUIC)

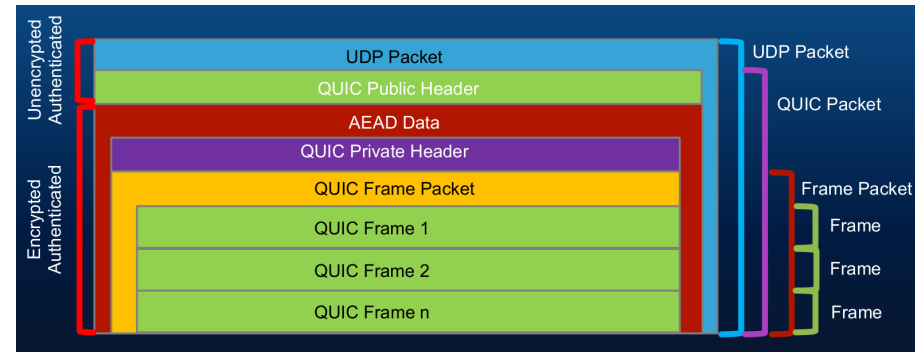


- **Planned for next releases:**

- Distributed Network Protocol (DNP3)
- KNX (building automation protocol)

IoT protocols – Quick UDP Internet Connections (QUIC)

- Created by Google and widely used in Google Apps like Docs, Maps and IoT devices
- Transport layer for HTTP/3
- General-purpose transport layer protocol with:
 - built-in communication security
 - multiple streams in one connection
 - low latency on setup
- UDP-based



Source: Kate Pearce „HTTP/2 & QUIC Teaching Good Protocols To Do Bad Things“

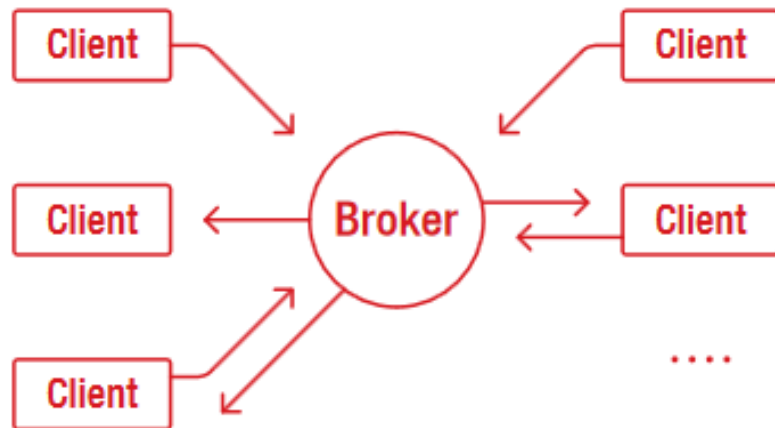
Message Queuing Telemetry Transport – Sensor Networks

- **MQTT**

- most popular IoT messaging protocol
- TCP-based

- **MQTT-SN (Sensor Networks)**

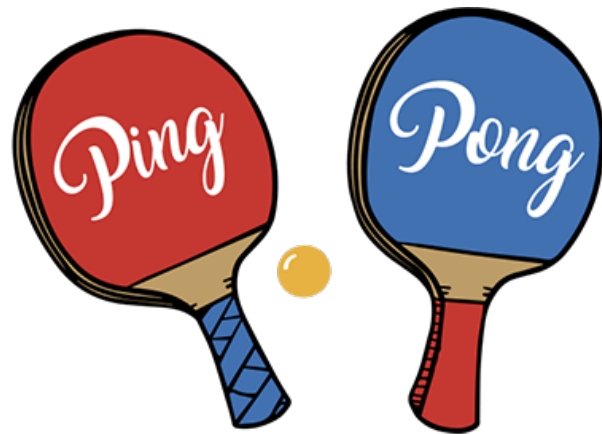
- UDP-based clone of MQTT
- small changes in packet formats
- all ideas are the same
- not as popular as MQTT



Source: Trend Micro „The Fragility of Industrial IoT’s Data Backbone”

Cotopaxi feature – service ping

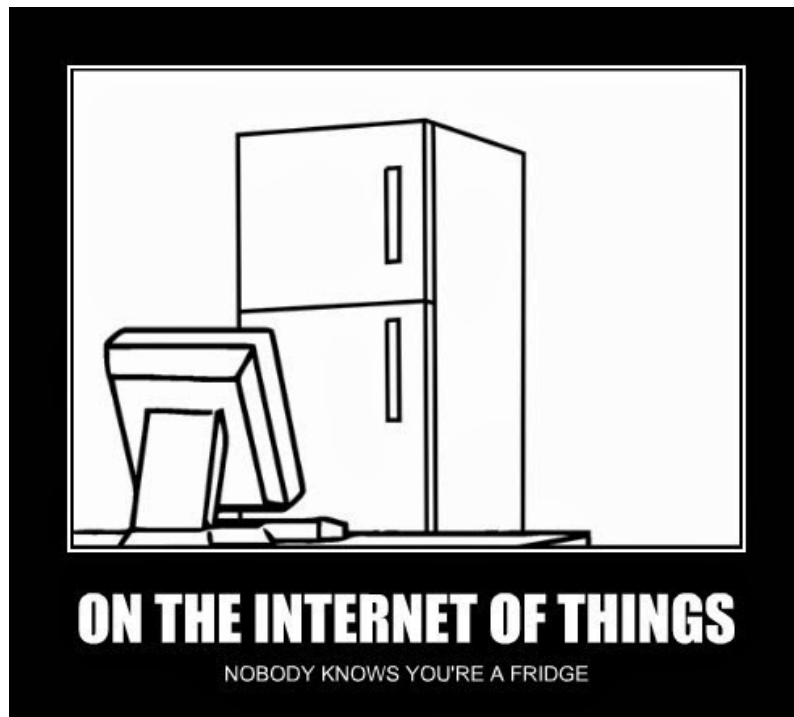
- Identifies active service endpoints (IPv4/IPv6:port)
- Not uses ICMP echo!
- For each protocol there is a set of messages that triggers responses in all tested servers
e.g. DTLS – Client Hello in all DTLS versions
- Better than using standard tools:
nmap and wireshark do not recognize IoT traffic on non-standard ports



Source: <https://www.tenstickers.pl/naklejki/naklejka-ping-pong-10870>

Cotopaxi feature – service fingerprinting

- Detection of software and version used by server
- Equivalent to *nmap -sV*
(Service and Application Version Detection)
- Uses machine learning classifier
- In this version works only for CoAP, DTLS



Source: <https://www.secmeme.com/2014/02/on-internet-of-things-nobody-knows.html>

Cotopaxi feature – device identification



- Identification of IoT devices using captured traffic (PCAP format) using machine learning classifier
- In this version supports 60+ devices: webcams, SmartHome devices, SmartTVs
- Corpus of traffic for IoT devices provided by authors of papers:

Jingjing Ren, Daniel J. Dubois, David Choffnes,
Anna Maria Mandalari, Roman Kolcun, Hamed Haddadi
*"Information Exposure for Consumer IoT Devices:
A Multidimensional, Network-Informed Measurement Approach"*

A. Sivanathan, H. Habibi Gharakheili, F. Loi, A. Radford,
C. Wijenayake, A. Vishwanath and V. Sivaraman,
*"Classifying IoT Devices in Smart Environments
Using Network Traffic Characteristics"*

```
-----  
[.] Started classification for IP: 13.37.0.17  
[!] Found 1000 packets to or from this IP  
[*] Device was classified as:  
    Xiaomi Mi Cam 2  
[*] Results of device identification:  
    Xiaomi Mi Cam 2: 59.09%  
    Amcrest Camera: 36.87%  
    Wansview Camera: 1.92%  
    Yi Camera: 1.62%  
    Bosiwo Camera: 0.20%  
    Luohe Cam: 0.20%  
    Lefun Cam: 0.10%  
[.] Classification time: 3.19 sec  
-----
```

Cotopaxi feature – resource listing (dirbusting)

- Equivalent to:
 - *DirBuster/dirb* for CoAP/mDNS/SSDP
 - *nmap* script *coap-resources*
- Uses a list of:
 - URIs (for CoAP)
 - services (for mDNS and SSDP)
 - media files (for RTSP)
- Cotopaxi includes sample lists of resource
- User can provide own list of URIs or services



Source: <https://medium.com/coinbundle/blockchain-internet-of-things-iot-be58703617c9>

Cotopaxi feature – protocol fuzzer (black-box)

- Uses corpus of malformed protocol messages (payloads) prepared with afl (American Fuzzy Loop)
- Checks „service ping” before and after sending payload
- Allows to use own corpus of payloads and integrate with mutating fuzzer
- In verbose mode displays payload and response packet
- Calculates RTT for payloads with responses and displays Top 10% - potentially interesting because of longest processing on server

	Protocol						
	AMQP	CoAP	DTLS	mDNS/ DNS-SD	MQTT	HTCPCP	SSDP
Size of corpus	147	557	395	77	285	469	271

Cotopaxi feature – vulnerability tester

- Types of vulnerabilities:
 - **information disclosure** – unauthorized access to internal information
 - **crash (DoS)** – leads to crash of server (detected by service ping)
 - **traffic amplification (for DDoS)** – responses larger than request
 - **memory leak** – server wastes memory after processing payload (requires manual confirmation)
 - **remote code execution** (currently only detected as crashes)
- In this release 10  vulnerabilities!

Type of Vulnerability	Protocol							
	CoAP	DTLS	mDNS/ DNS-SD	MQTT	HTTP	HTCPCP	SSDP	RTSP
Information Disclosure					2			2
Crash	5	6	1	1	1	1	1	6
Traffic Amplification	1	2						
Memory Leak	1							
Remote Code Execution		1	2	1				
TOTAL* = 34	7	9	3	2	3	1	1	8

* some vulnerabilities are currently in responsible disclosure process

Cotopaxi feature – vulnerability tester

Example bugs found by our team and identified by Cotopaxi:

- **CVE-2019-9747** in tinysvcmdns (hang)
mDNS server goes into infinite loop after receiving DNS query with recursive referenced names
- **CVE-2019-9004** in Wakaama (memory leak)
CoAP server leaks (wastes) 24 bytes per each processed crafted packet
- **CVE-2019-9750** in IoTivity (traffic amplification)
CoAP server responds with 6 error messages
- **CVE-2019-18840** in WolfSSL (crash)
TLS and DTLS servers and clients can be crashed using malformed x509 certificate



Source: Grasshopper shot near Miles
City Mont. C. 1937 Coles Studio
Glasgow Mont

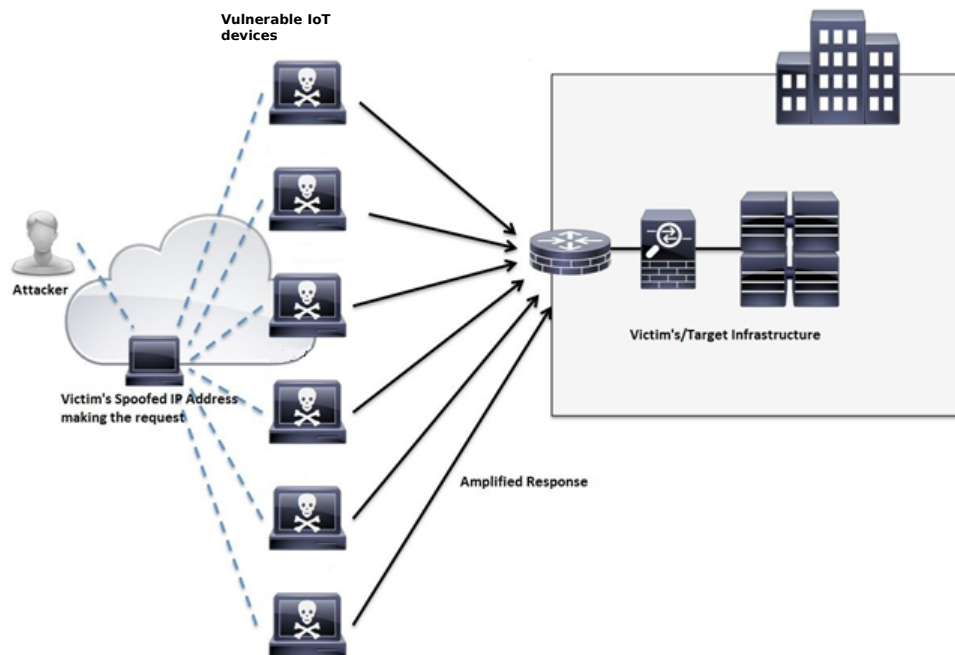
Cotopaxi feature – amplification sniffer

- Sniffs for all packets incoming to and outgoing from specified target
- Calculates amplification factor ($\text{size_out}/\text{size_in} - 1$)
- Tracks req/resp with highest amplification factor and display record on exit
- Should be placed on router or use network tap to see all traffic to/from target



Source: <http://m.quickmeme.com/Rat-Sniffer/>

DDoS attack via IP Spoofing and Traffic Amplification



- Identified amplifiers-reflectors:

- **CoAP**

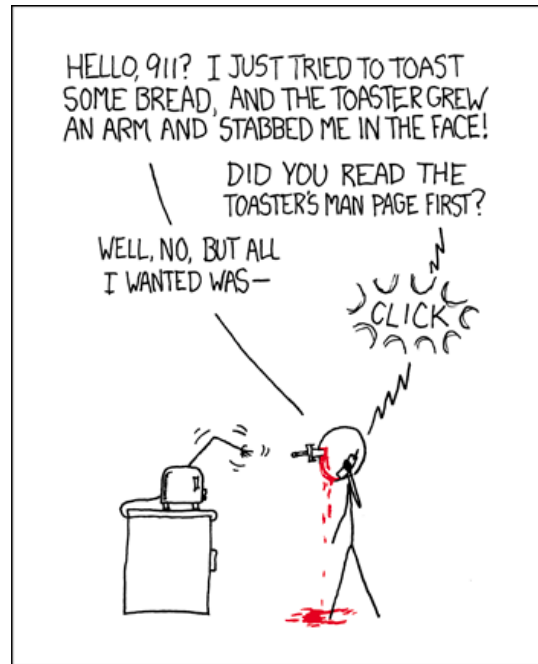
- every request with large response
 - e.g. CoAPthon example /big
 - IoTivity 4.04 6x repeated response
 - IOTIVITY_000 issue in tester

- **DTLS**

- every DTLS server without Hello Verify Request
 - e.g. Botan_000 issue in tester

Cotopaxi – next steps

- Download from Samsung GitHub
<https://github.com/samsung/cotopaxi>
- **Hack the planet!**
...but only with the written consent of the owner :-)
- Read the friendly manual in Readme.md
- Report any issues or requests for new features on GitHub
- Pull request with new features



Source: <https://while2.ghost.io/rtfmc/>

Thank You!

Do you have any questions?

