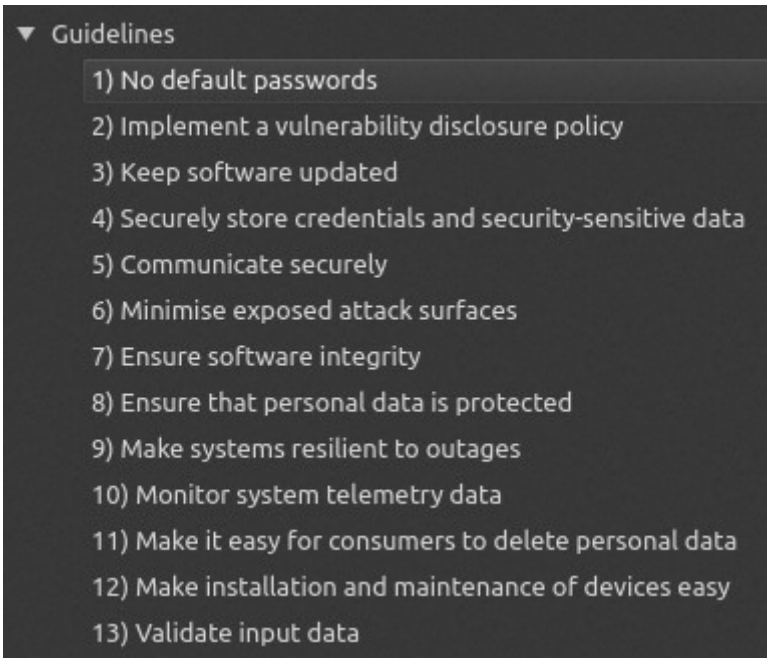
Useful resources:

- *UK Govt*
  - *Code of Practice for Consumer IoT Security, DCMS, Govt of UK, Oct 2018 – 13 practical 'real-world' guidelines / recommendations for IoT security*



1) No default passwords

All IoT device passwords shall be unique and not resettable to any universal factory default value

2) Implement a vulnerability disclosure policy

All companies that provide internet-connected devices and services shall provide a public point of contact as part of a vulnerability disclosure policy in order that security researchers and others are able to report issues. Disclosed vulnerabilities should be acted on in a timely manner.

3) Keep software updated

Software components in internet-connected devices should be securely updateable. Updates shall be timely and should not impact on the functioning of the device. An end-of-life policy shall be published for end-point devices which explicitly states the minimum length of time for which a device will receive software updates and the reasons for the length of the support period. The need for each update should be made clear to consumers and an update should be easy to implement. For constrained devices that cannot physically be updated, the product should be isolatable and replaceable.

4) Securely store credentials and security-sensitive data

Any credentials shall be stored securely within services and on devices. Hard-

coded credentials in device software are not acceptable.

5) Communicate securely

Security-sensitive data, including any remote management and control, should be encrypted in transit, appropriate to the properties of the technology and usage. All keys should be managed securely.

6) Minimise exposed attack surfaces

All devices and services should operate on the ‘principle of least privilege’; unused ports should be closed, hardware should not unnecessarily expose access, services should not be available if they are not used and code should be minimised to the functionality necessary for the service to operate. Software should run with appropriate privileges, taking account of both security and functionality.

7) Ensure software integrity

Software on IoT devices should be verified using secure boot mechanisms. If an unauthorised change is detected, the device should alert the consumer/administrator to an issue and should not connect to wider networks than those necessary to perform the alerting function.

8) Ensure that personal data is protected

Where devices and/or services process personal data, they shall do so in accordance with applicable data protection law, such as the General Data Protection Regulation (GDPR) and the Data Protection Act 2018. Device manufacturers and IoT service providers shall provide consumers with clear and transparent information about how their data is being used, by whom, and for what purposes, for each device and service. This also applies to any third parties that may be involved (including advertisers). Where personal data is processed on the basis of consumers’ consent, this shall be validly and lawfully obtained, with those consumers being given the opportunity to withdraw it at any time.

9) Make systems resilient to outages

Resilience should be built in to IoT devices and services where required by their usage or by other relying systems, taking into account the possibility of outages of data networks and power. As far as reasonably possible, IoT services should remain operating and locally functional in the case of a loss of network and should recover cleanly in the case of restoration of a loss of power. Devices should be able to return to a network in a sensible state and in an orderly fashion, rather than in a massive scale reconnect.

10) Monitor system telemetry data

If telemetry data is collected from IoT devices and services, such as usage and measurement data, it should be monitored for security anomalies.

11) Make it easy for consumers to delete personal data

Devices and services should be configured such that personal data can easily be removed from them when there is a transfer of ownership, when the consumer wishes to delete it and/or when the
consumer wishes to dispose of the device. Consumers should be given clear

instructions on how to delete their personal data.

12) Make installation and maintenance of devices easy

Installation and maintenance of IoT devices should employ minimal steps and should follow security best practice on usability. Consumers should also be provided with guidance on how to securely set up their device.

13) Validate input data

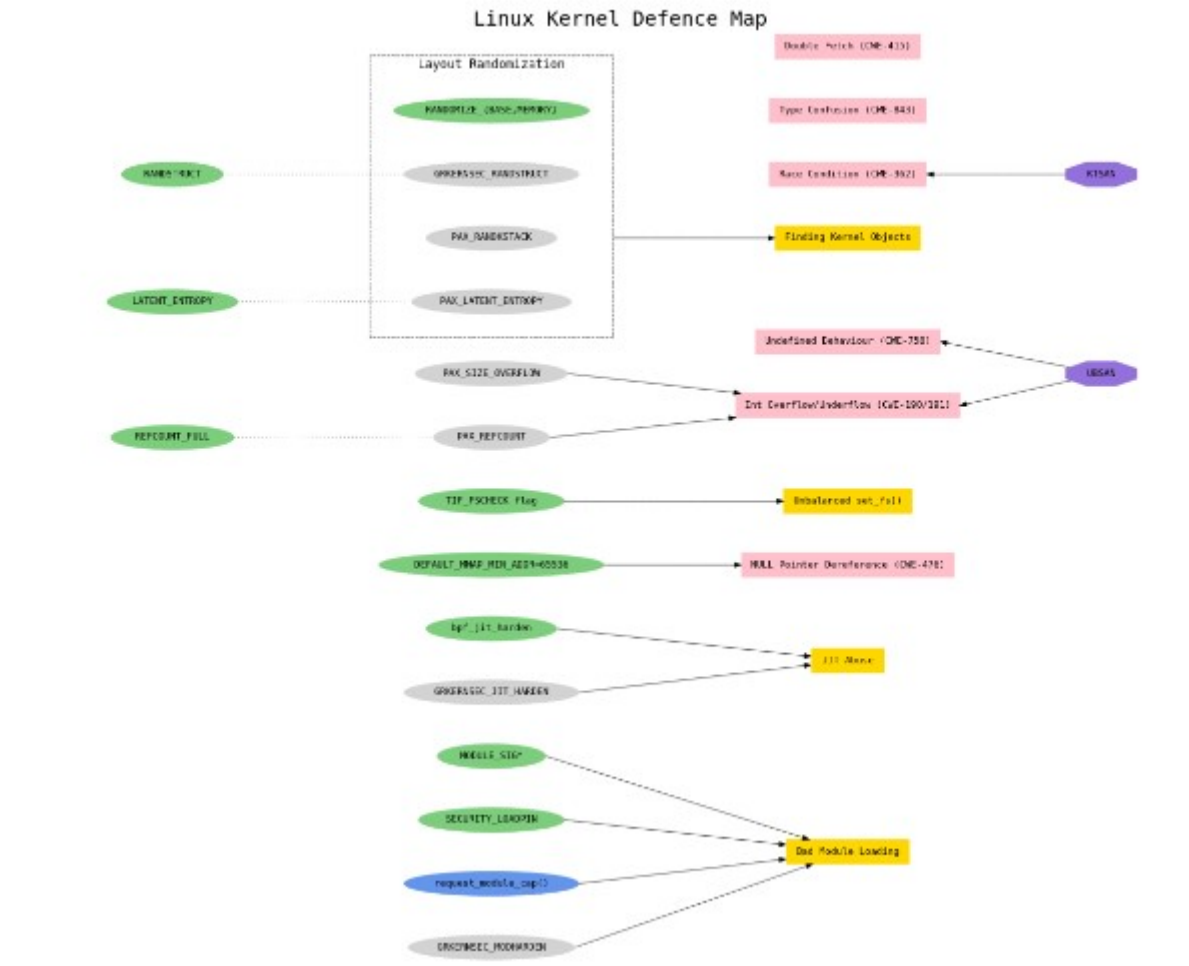
Data input via user interfaces and transferred via application programming interfaces (APIs) or between networks in services and devices shall be validated.

- *Mapping of IoT Security Recommendations, Guidance and Standards to the UK's Code of Practice for Consumer IoT Security, Oct 2018 (huge list of linked industry resources for each of the above 13 recommendations made)*

- *Improving your Embedded Linux Security Posture **with Yocto** [PDF]*

- *Linux Kernel Defence Map, Alexander Popov (below!)*

The Map for v5.3



[...]

- *Mind Map – Security Basics, Mayur Pahwa, Jan 2020*
- *Security - Resources, courtesy Mayur Pahwa*

Besides the coding/dev side of things.. also, lets get particular to some hardware-software combination so that the discussion remains practical.

So, we say
- hardware: Raspberry Pi 3 Model B+ (Broadcom BCM2835 based SoC with quad core ARM Cortex-A53); commonly used in several DIY IoT projects
- software: Raspbian OS (Debian-based), Linux kernel 4.x

Ref:

How to secure your Raspberry Pi
Find out what sensible steps you can take to protect your Raspberry Pi and other IoT devices. Ian Kluft.

Attack surface reduction
- Physical security (what if someone steals the microSD / device itself!)
  - USB ports, ethernet – disable?
- Risk matrix – categorize components 'risk' (critical, high, medium, low)
- change the default password! (to a secure one)
- use addition security at the OS- Linux LSMs like SELinux in enforcing mode
- use encryption (SSL/TLS, SSH, etc) – all via software packages
- keep the system updated
  - products Must have a secure & tested OTA or other update framework in place
- check security updates /status - CVEs / NIST / etc

Web: OWASP Top Ten Cheat Sheet

Securing a Raspberry Pi embedded in your IoT device, Ori Pomerantz, May 17, 2017, IBM Developer
Biased towards network traffic monitoring, firewall (iptables) setup.. code-based (Node.js).

IOT SECURITY: HARD PROBLEM, NO EASY ANSWERS Fahmida Y. Rashid

IoT Security Wiki : One Stop for IoT Security Resources
Huge number of resources (whitepapers, slides, videos, etc) on IoT security

Hacks-

- US-CERT Alert (TA16-288A) - Heightened DDoS Threat Posed by Mirai and Other Botnets
"On September 20, 2016, Brian Krebs' security blog (krebsonsecurity.com) was targeted by a massive DDoS attack, one of the largest on record, exceeding 620 gigabits per second (Gbps).[1] An IoT botnet powered by

Mirai malware created the DDoS attack. The Mirai malware continuously scans the Internet for vulnerable IoT devices, which are then infected and used in botnet attacks. The Mirai bot uses a short list of 62 common default usernames and passwords to scan for vulnerable devices. Because many IoT devices are unsecured or weakly secured, this short dictionary allows the bot to access hundreds of thousands of devices.[2] The purported Mirai author claimed that over 380,000 IoT devices were enslaved by the Mirai malware in the attack on Krebs' website.[3]
In late September, a separate Mirai attack on French webhost OVH broke the record for largest recorded DDoS attack. That DDoS was at least 1.1 terabits per second (Tbps), and may have been as large as 1.5 Tbps.[4] ..."

- *Reg insecure / weak passwords, see this tweet*

https://github.com/lcashdol/IoT/blob/master/passwords/list-2019-01-29.txt

- *Hacking DefCon 23's IoT Village Samsung fridge, Aug 2015 (DefCon 23)*

- *HACKING IoT: A Case Study on Baby Monitor Exposures and Vulnerabilities*
"... these devices are marketed and treated as if they are single purpose devices, rather than the general purpose computers they actually are. ...
...
IoT devices are actually general purpose, networked computers in disguise, running reasonably complex network-capable software. In the field of software engineering, it is generally believed that such complex software is going to ship with exploitable bugs and implementation-based exposures. Add in external components and dependencies, such as cloud-based controllers and programming interfaces, the surrounding network, and other externalities, and it is clear that vulnerabilities and exposures are all but guaranteed."

<< See pg 6, 'Ch 5: COMMON VULNERABILITIES AND EXPOSURES FOR IoT DEVICES' ; old and new vulnerabilities mentioned;

Pg 9: 'Disclosures' - the vulns uncovered in actual products >>

Just too much. Bottom line: critical to perform adequate pentesting – either outsource or DIY

Miscellaneous

"Unsafe at any clock speed: Linux kernel security needs a rethink"
Ars reports from the Linux Security Summit—and finds much work that needs to be done.
J.M. PORUP - Sept 2016.

IoT App Development Protocols

- Advanced Message Queuing Protocol
- OASIS Message Queuing Telemetry Transport
- Very Simple Control Protocol
- Constrained Application Protocol
- Extensible Messaging and Presence Protocol

Misc

MQTT Protocol – How it Works *(Message Queuing Telemetry Transport)*

Why is My Perfectly Good Shellcode Not Working?: Cache Coherency on MIPS and ARM
short ans: cache coherency issues.
