Security Testing – A Building Block of DevSecOps

Harri Susi
Twitter @hsusi
11.09.2018

Etteplan

Industrial Cobots Might Be The Next Big **IoT** Security Mess

Threatpost - 22 Aug 2017

Researchers at IOActive have found nearly 50 **vulnerabilities** in industrial collaborative robots, machines that work side-by-side with people in ...

Popular Robots are Dangerously Easy to Hack, Researchers Say

SuperSite for Windows - 23 Aug 2017

Connected cars have an 'indefensible' security **vulnerability**

Network World - 23 Aug 2017

In a blog post published last week, "The Crisis of Connected Cars: When **Vulnerabilities** Affect the CAN Standard," the company publicized an ...

Security flaw in **IoT** solar equipment could disrupt Europe's electricity ...

The Internet of Business (blog) - 11 Aug 2017

A Dutch security researcher claims to have found **vulnerabilities** in internet-connected solar panel equipment installed throughout Europe, ...

DDoS attack that disrupted internet was largest of its kind in history ...

https://www.theguardian.com/technology/2016/oct/26/ddos-attack-dyn-mirai-botnet ▾

Oct 26, 2016 - **Dyn**, the victim of last week's **denial of service attack**, said it was orchestrated using a weapon called the **Mirai botnet** as the 'primary source of ...

For cybercriminals, **IoT** devices are big business, part two

TechTarget (blog) - 17 Aug 2017

The **Internet of Things** (**IoT**) world may be exciting, but there are serious ... That's because these attacks targeted a **vulnerability** for which a ...

3 ways developers can improve **IoT** security on their devices

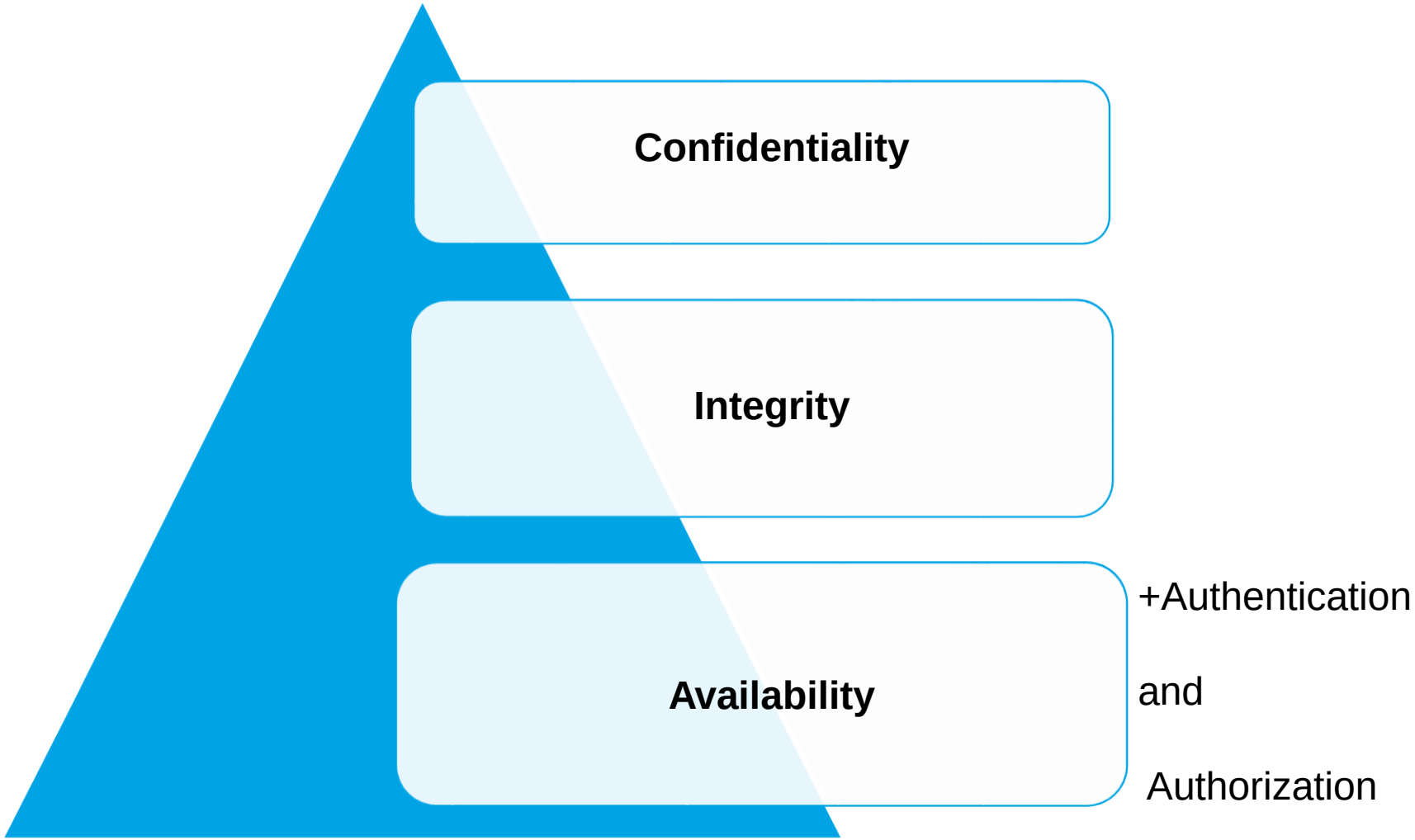
Network World - 18 Aug 2017

Router flaws put AT&T customers at hacking risk

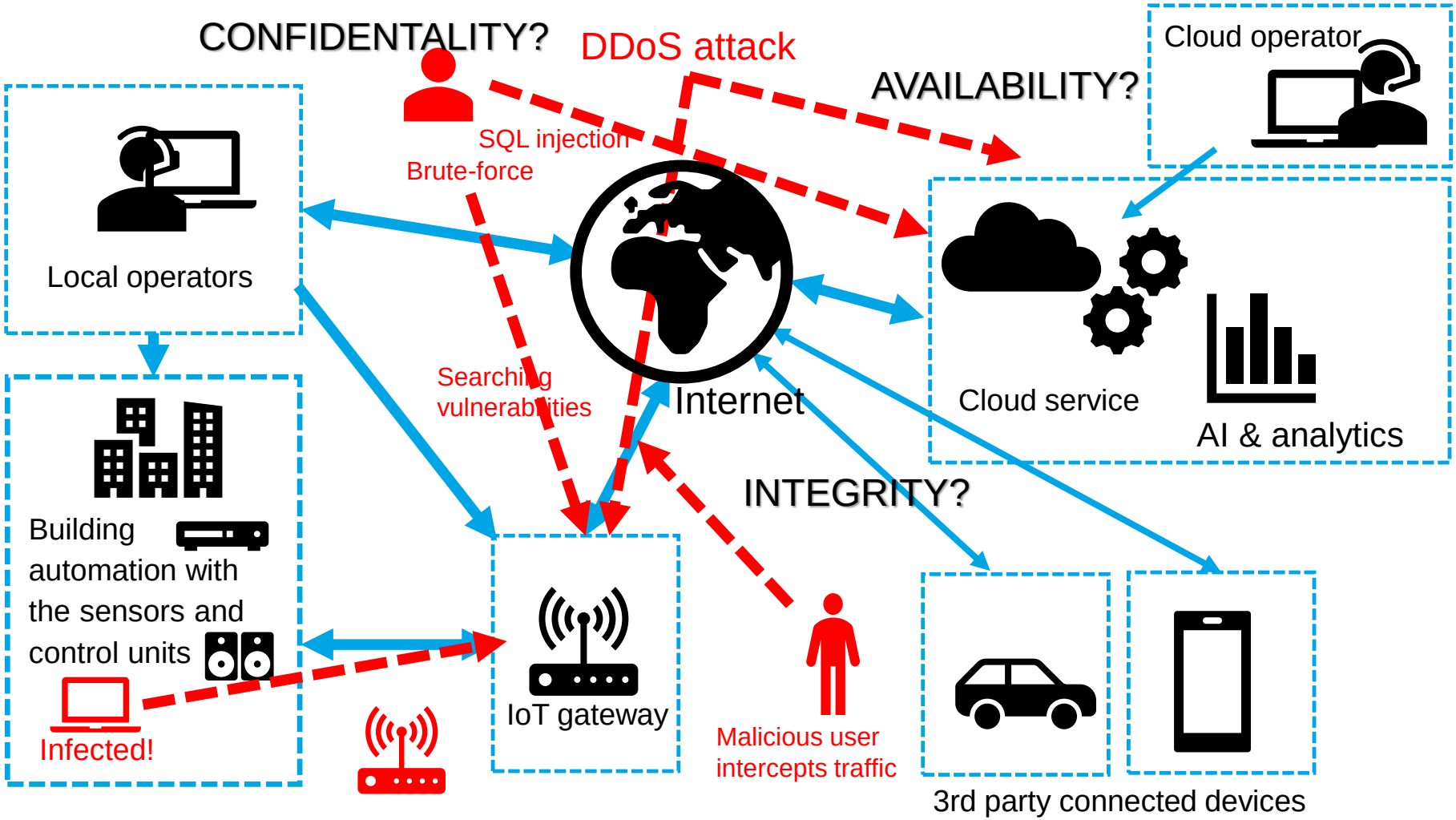
ZDNet - 23 minutes ago

Among the **vulnerabilities** are hardcoded credentials, which can allow ... Rapid7 reported the **vulnerability** as an 8/10, on the higher end of the ...

CIA(AA): Principles of Data Security



CASE: IoT Building Automation & Threats!



Security and Automated Testing

- How much can we automate security testing? And how can we integrate security into the SW development process?
- You know that you should automate as much as possible but don’t expect too much!
- There are a lot of things in security testing that can be automated but there are things which cannot be

The next slides present some security testing types - think about how much they can be automated

Threat Model Driven Testing

There are threats…

… ways to mitigate them

and mitigations should be validated

-> threat model driven testing!

Threat Modelling – What Is That?

- Threat modelling is a process/workshop where a team plans how to design the system in a way that makes it a challenging target for the attackers
- WHY? Understanding how an attacker can compromise the system helps the team analyze the design and make appropriate design decisions to mitigate the potential threats

Threat Modelling Process

- Model the solution/system/application
  - Draw it on a whiteboard with the team! (e.g. use data flow diagram)
- Enumerate Threats
  - STRIDE: Spoofing identity, Tampering with data, Repudiation, Information disclosure, Denial of service, Elevation of privilege
- Mitigate threats by changing your design
  - Create security user stories
  - Create evil-user stories
  - Create test cases
- Validate the mitigations
  - Test it!

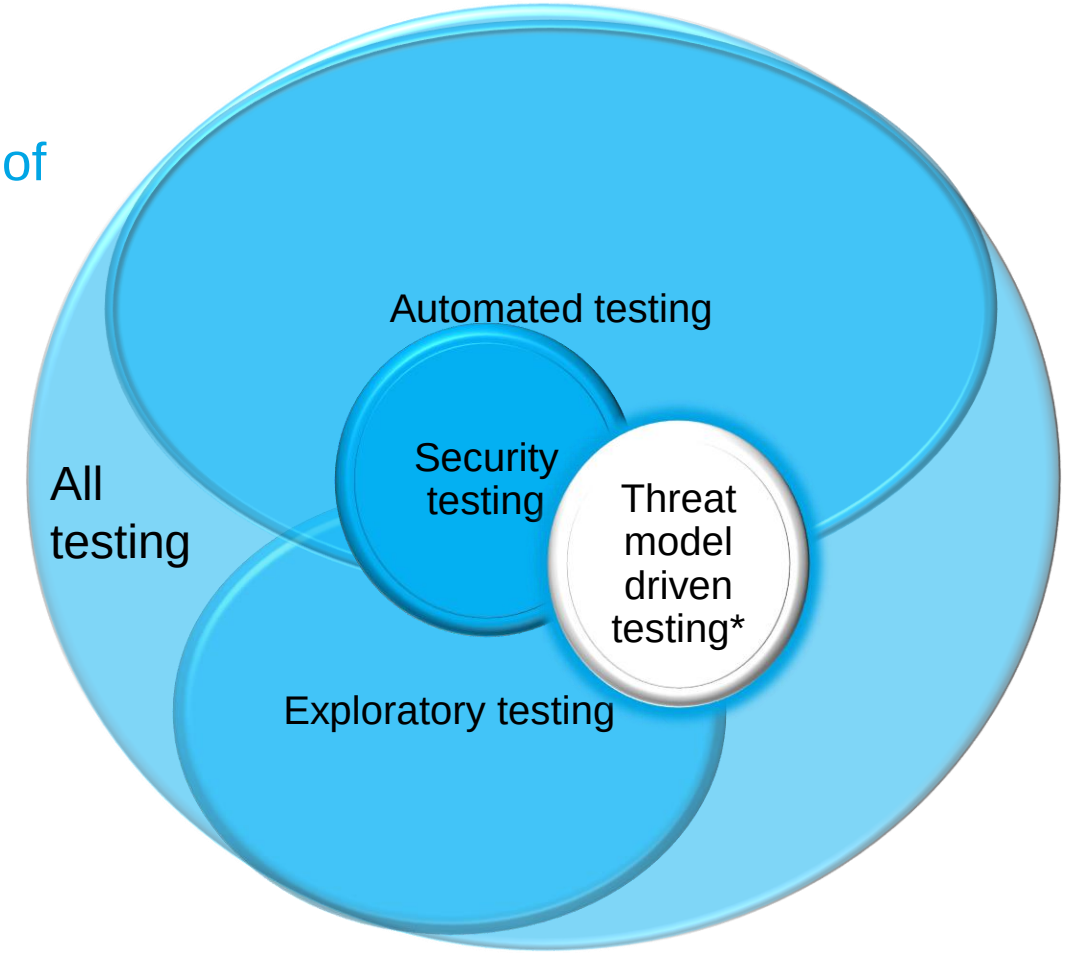
Threat Modelling: STRIDE

the method to detect threats

1) Spoofing identity, e.g. an attacker connects anonymously because authentication is expected to be done in higher level **[Authentication]**
2) Tampering with data, e.g. an attacker replays data without detection because no timestamps/seq numbers are needed **[Integrity]**
3) Repudiation, e.g. an attacker can say ”I didn’t do that” and there is no way to prove them wrong **[Non-repudiation]**
4) Information disclosure, e.g. an attacker can see error messages with security sensitive content **[Confidentality]**
5) Denial of service, e.g. an attacker can make the control unit unstable with too frequent connection requests **[Availability]**
6) Elevation of Privilege, e.g. an attacker manages to exploit a vulnerability of the unpatched registered connected device and gains the higher level access to the control unit **[Authorization]**

Testing Threat Mitigations

Types of testing



* Priority and Focus

Security Testing: Negative Testing - try to break it!

- Behave like a bad user or malicious user
- What if I put SQL commands here instead of proper plain text?
- SQL, HTML, Javascript injections
- Stress testing
- How are error messages shown? Any sensitive information revealed in the error message, e.g. database structures or folder names?

Sounds bit like Exploratory testing?

Security Testing, Positive Testing

Verifying the proper implementation of security features

- Is the traffic encrypted?
- Is the stored data encrypted?

Security Testing: Boundary Tests

automated / manual exploratory testing

- JUST INSIDE – JUST OUTSIDE – think about corner cases
- Off-by-one
- But never forgot to test typical values and error values also (strings into integer fields, etc…)

Sounds like Exploratory testing?

Security Testing: Vulnerability Scanning

Automate this! Integrate vulnerability scanning to CI/CD pipeline!

- Scan the DUT against known vulnerabilities, check all open ports, services
- There are some good vulnerability scanners available, e.g. F-Secure RADAR
- https://www.shodan.io/ Search engine for IoT! Is your system visible? How does it respond?
- Network scanning tools: Wireshark, Nessus, …

F-Secure Radar

Overview

Open ports

Open TCP ports (3): 22, 80, 443
Open UDP ports (1): 53

Most severe (10)

- OpenSSH through 6.9 Multiple Vulnerabilities
- OpenSSH before 7.3 Denial of Service Vulnerability
- OpenSSH 'ssh/kex.c' Denial of Service Vulnerability
- OpenSSH before 7.4 Multiple Vulnerabilities
- OpenSSH before 7.2 Security Bypass Vulnerability

Show all

Vulnerability tags (9)

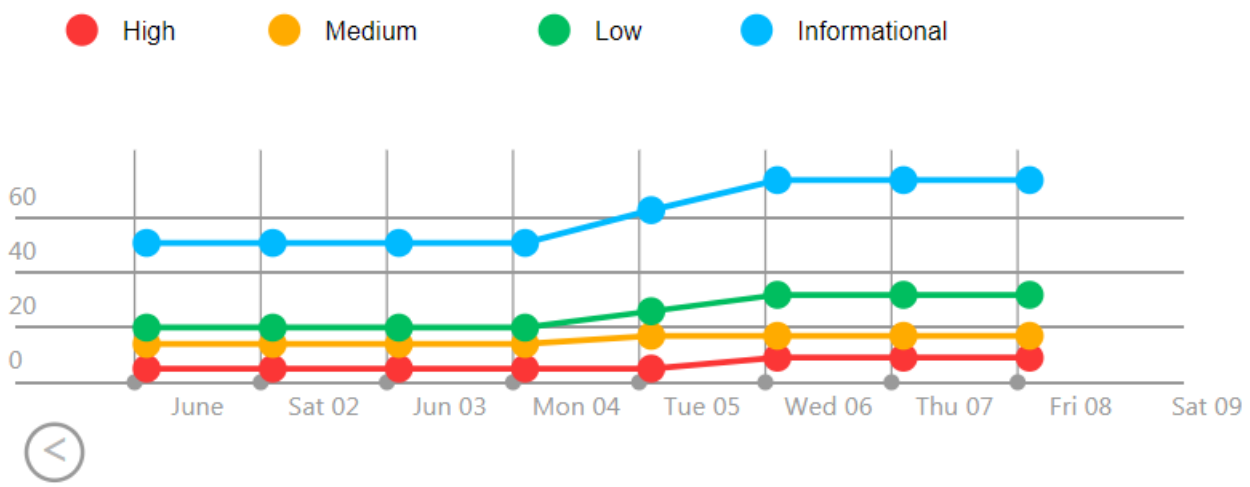
BEAST | Informational | OpenSSH | Service detection | SSH | SSL | Show all

New (0)

No new findings

Fixed (0)

No fixed findings



Etteplan

Security Testing: Fuzzing

Fuzz testers taking less time to spot **vulnerabilities** in **IoT** protocols
SC Magazine UK - 10 Aug 2017
The study also found a common protocol used in **IoT** devices, which are notorious being plagued with **vulnerabilities**, was significantly more ...

Automate this! Integrate it in your CI/CD pipeline!

- Fuzzing is about sending malicious input to the system under test. The input is sent by specific tool (the fuzzer). The sent data is generated with help of fuzzing algorithms. The test engineer feeds the tool with the proper seed data
- Use the best possible tools but quite often you need to do some customization
- Radamsa, Burp Suite, etc.

Security Testing

Can also include **penetration testing** – everything white hat hackers do

E.g. combination of vulnerability scanning, negative testing and some social engineering

but that’s another story!

Security Testing

Should be part of software development process

- static code analysis, unit tests, code reviews…

Automate security testing?

- Use threat modelling as a starting point for the test cases
- Automate performance & stress tests. Follow performance trends to avoid internal DoS design flaws
- Vulnerability scanning can be easily automated: scanners have typically e.g. Jenkins-plugins / easy to use APIs
- Fuzz tests should be integrated as part of test automation (e.g. nightly builds)

Summary & takeways

1. Integrate security into your product development process right from the beginning
2. Take threat modelling to all R&D projects. Without threat modelling you cannot understand testing priorities
3. Establish continuous security testing and automate as much you can:

CI/CD pipeline: threat modelling → test cases → test automation

Finally…

DevOps is Dead – Long Live DevSecOps!

Thank You!

Any Questions?

Harri Susi
Twitter @hsusi
11.09.2018

Etteplan