Making "Things" Secure

Cybersecurity of the IoT

Konstantinos Kolias, Angelos Stavrou

Computer Science Department

George Mason University

Irena Bojanova, Jeff Voas

Information Technology Laboratory

National Institute of Standards & Technology

Internet of Things Defined

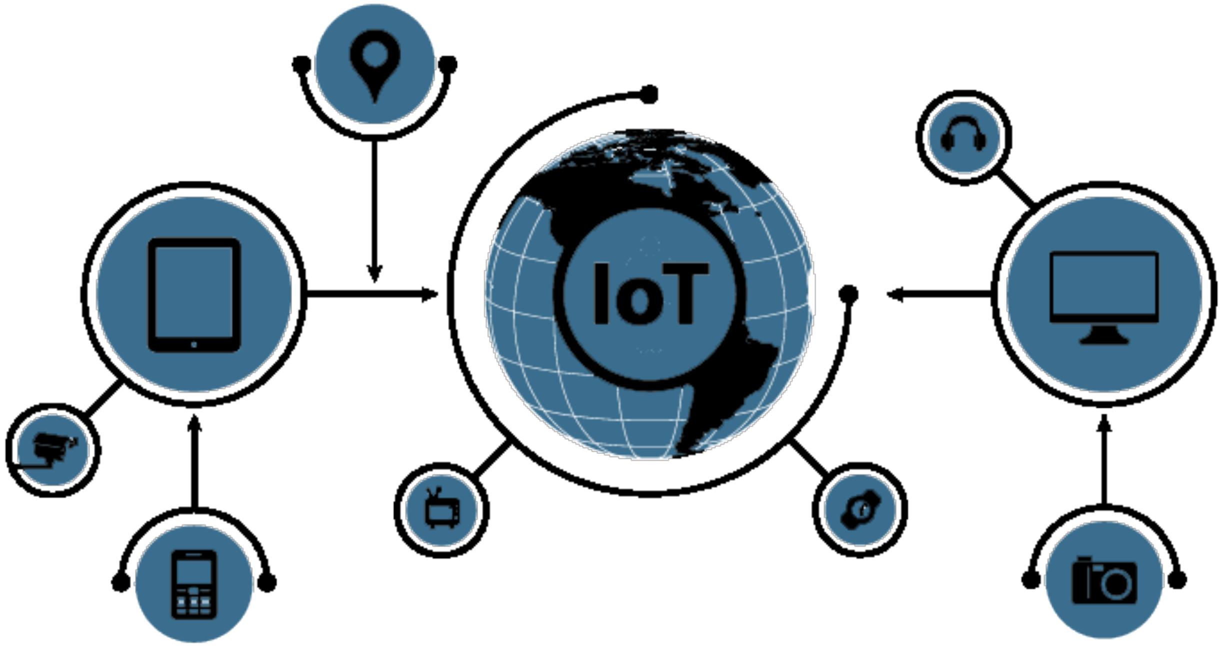
- Kevin Ashton introduced the term Internet of Things (IoT) in 1999

- Network of devices able to configure themselves automatically

- Human is not the center of the system

- **Motivation**: Better understanding of the environment and response to certain events. Machines are doing better in sensing & reporting on conditions

- **Challenge**: Applications of traditional Internet are different than the applications of IoT



What the Future Holds

Drivables

Flyables

Scanables

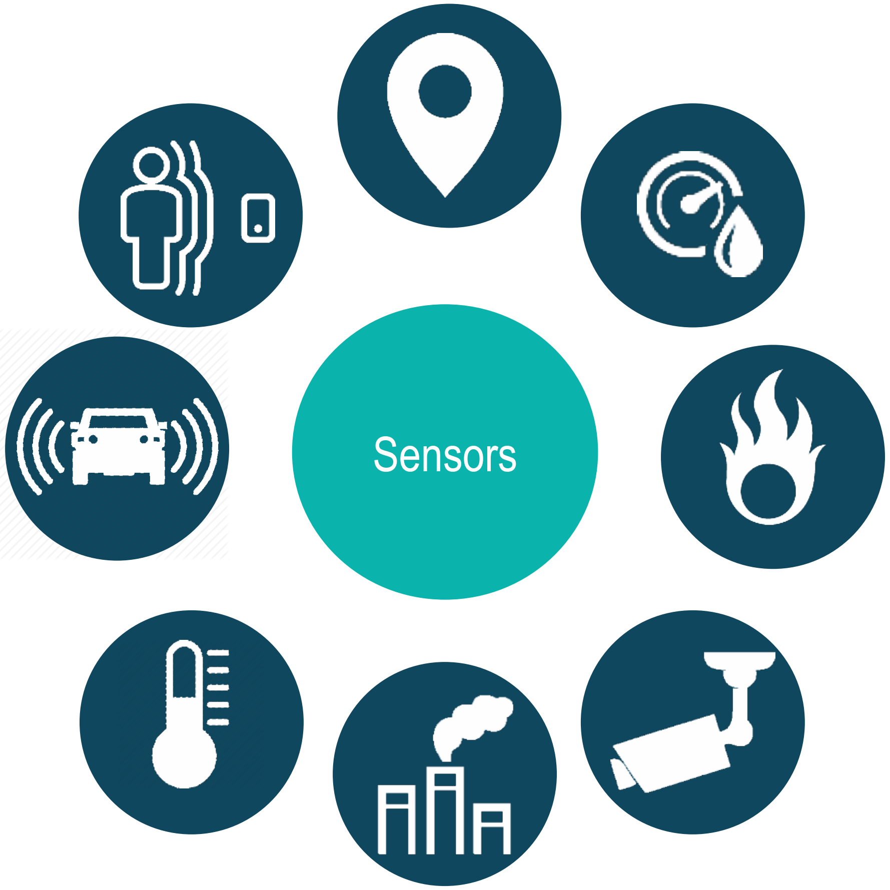
Wearables

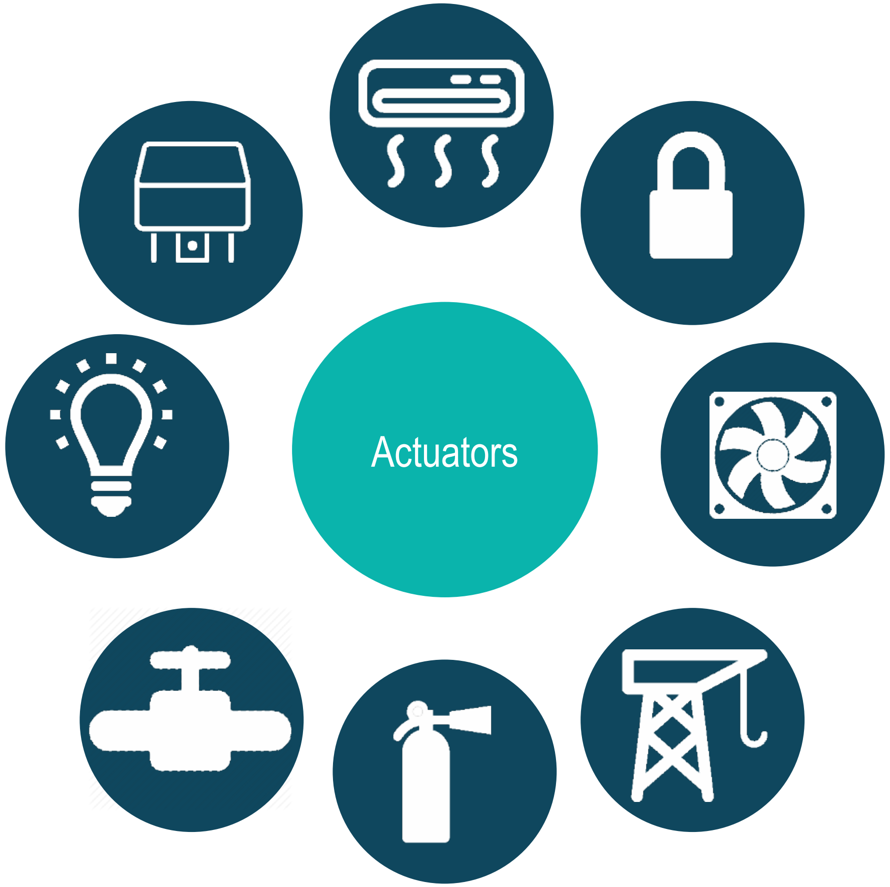
Fast track
11C
MR GARY GALE
BA 0990
to Berlin
12 Oct - 07:20
Scan

Sectors of IoT Applications

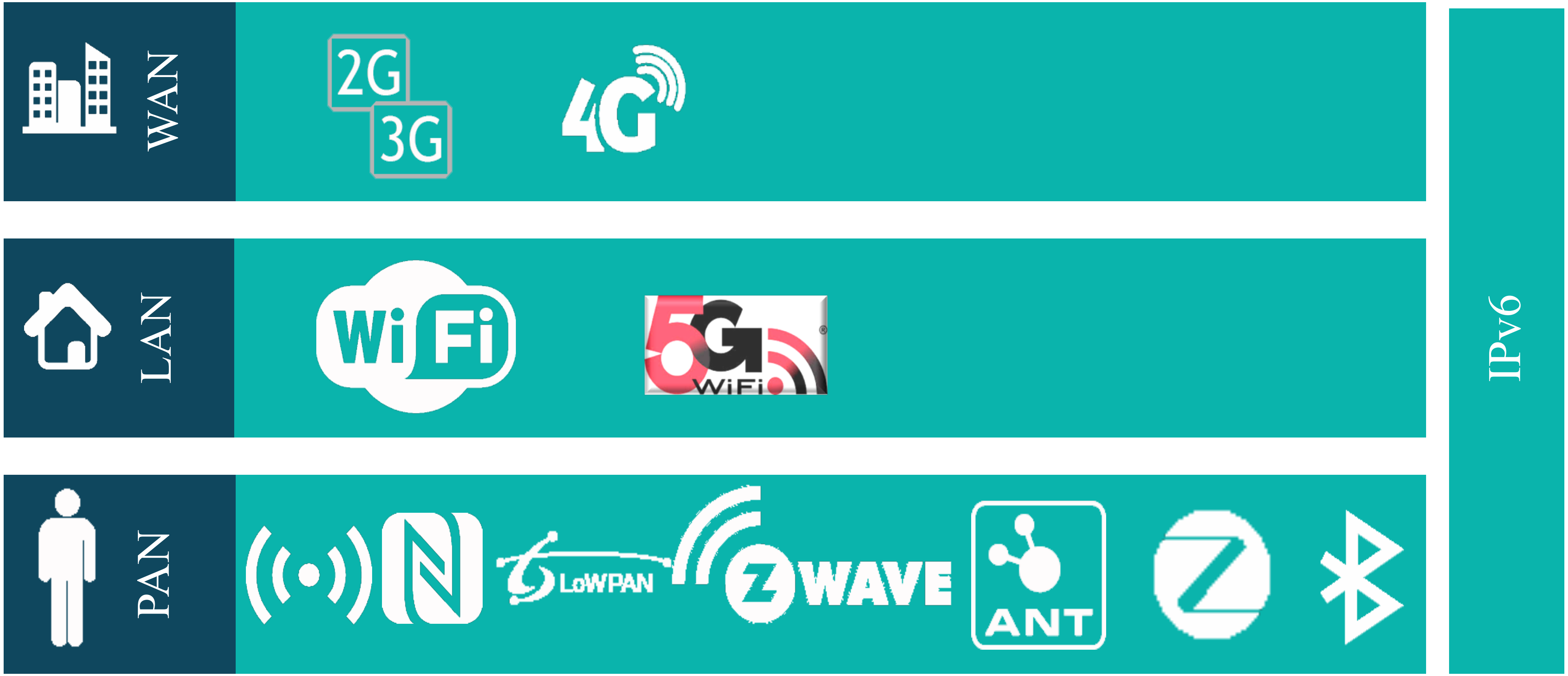
Smart Home	Transportation	Retail	Industry	Healthcare
Home automation	Road safety	Automatic payments	Quality assurance	Condition monitoring
Energy efficiency	Traffic regulation	Efficient cataloguing	Failure prediction	Remote treatment
Home security	Law enforcement	Shipment tracking	Productivity improvement	Personalized advices

Sensors & Actuators





Connectivity



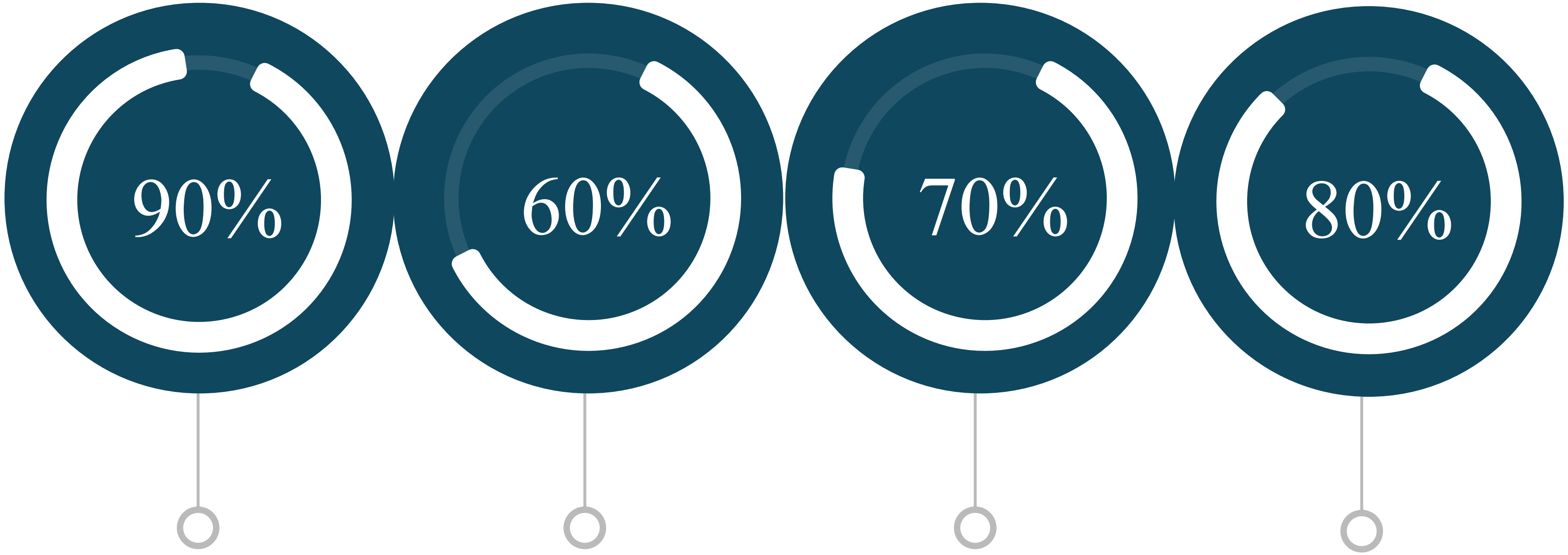
WAN

LAN

PAN

IPv6

Common Security Incidents - OWASP



90%

Private Data Collection

60%

Insecure Interfaces

70%

Unencrypted Communications

80%

Weak Requirements

Top 10 Vulnerabilities (OWASP)

Insecure Web Interfaces
Default accounts, XSS, SQL injection

Inefficient Authentication/Authorization
Weak passwords, no two-factor authentication

Insecure Network Services
Ports open, use of UPnP, DoS attacks

Lack of Transport Encryption
No use of TLS, misconfigured TLS, custom encryption

Private Data
Unnecessary private information collected

Insecure Cloud Interfaces
Default accounts, no lockout

Inefficient Mobile Interfaces
Weak passwords, no two-factor authentication

Insufficient Security Configurability
Ports open, use of UPnP, DoS attacks

Insecure Software/Firmware
Old device firmware, unprotected device updates

Poor Physical Security
Exposed USB ports, administrative accounts

Sensitive Information Leakage

- Fifth (5th) most popular vulnerability in IoT applications (OWASP)
- 90% of most popular IoT applications transmit at least one private piece of information
- Cases where sensitive information is collected but is redundant for the functionality of application
- Cases where the collection of private information is not properly communicated with the user
- User unaware of any leakage

Sensitive Information (Location) Leakage

- Technologies and protocols can be misused
  - New features
  - “Innocent” functionality to a commercial product

- Introduce an opportunity to be tracked
  - Aggressive advertising
  - Government surveillance
  - Terrorism

- Inexpensive to achieve

Use Case: iBeacon + Lights

- Conventional motion sensing switches
  - No personalization effect: on or off
- Phillips Hue Lights can be manipulated (turn on/off, change color) remotely
  - They do not respond differently for different users
- Personalized behavior based on the presence of user in small area (room)
- Combine multiple products to achieve the desired effect

Use Case: Automated Watering System

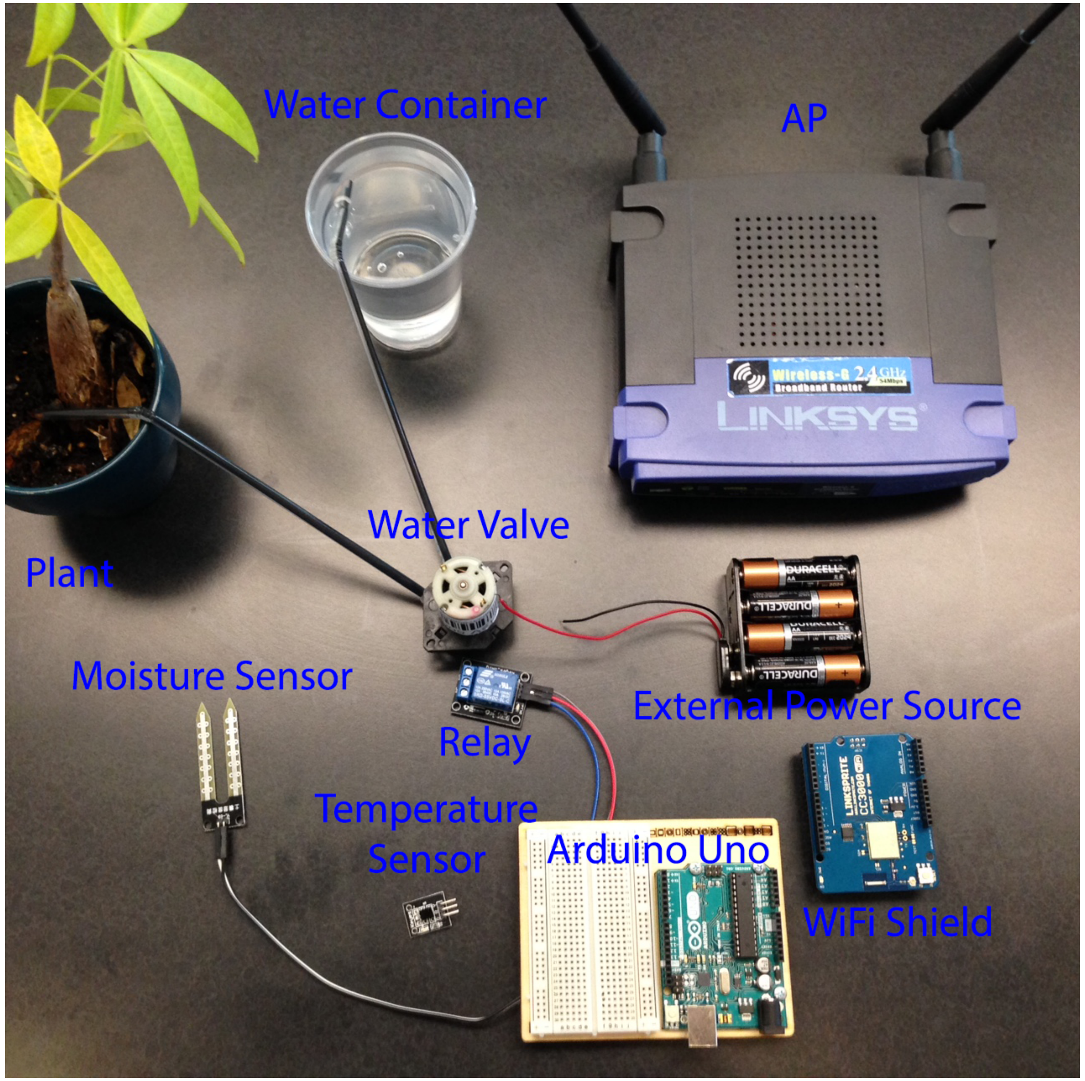
- Conventional watering systems
  - Rely on clock settings
  - No remote control
  - No dynamic behavior

- Remotely monitor the moisture levels on the ground

- Remotely enable/disable watering pump

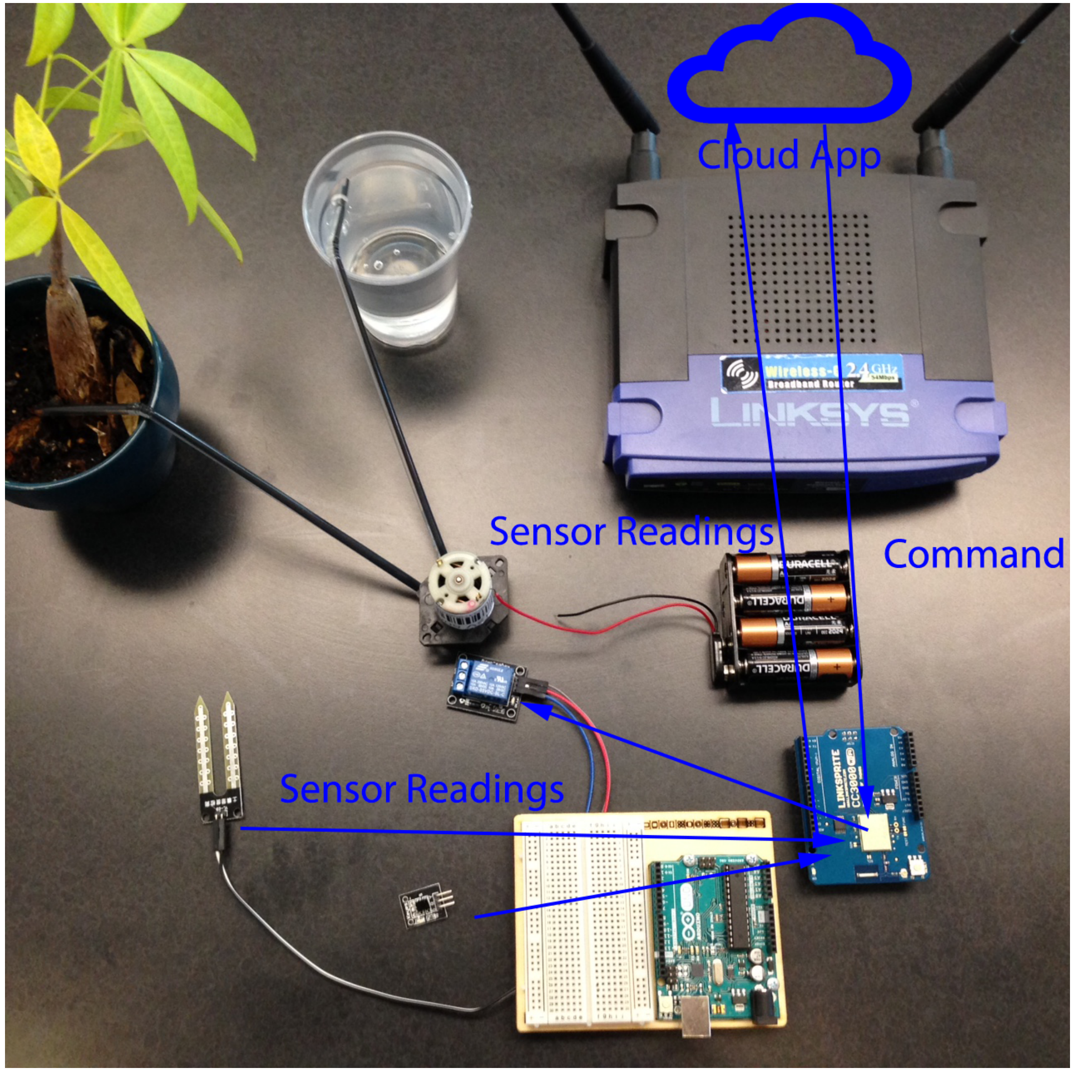
- Temperature/Pressure Readings

Use Case: Automated Watering System



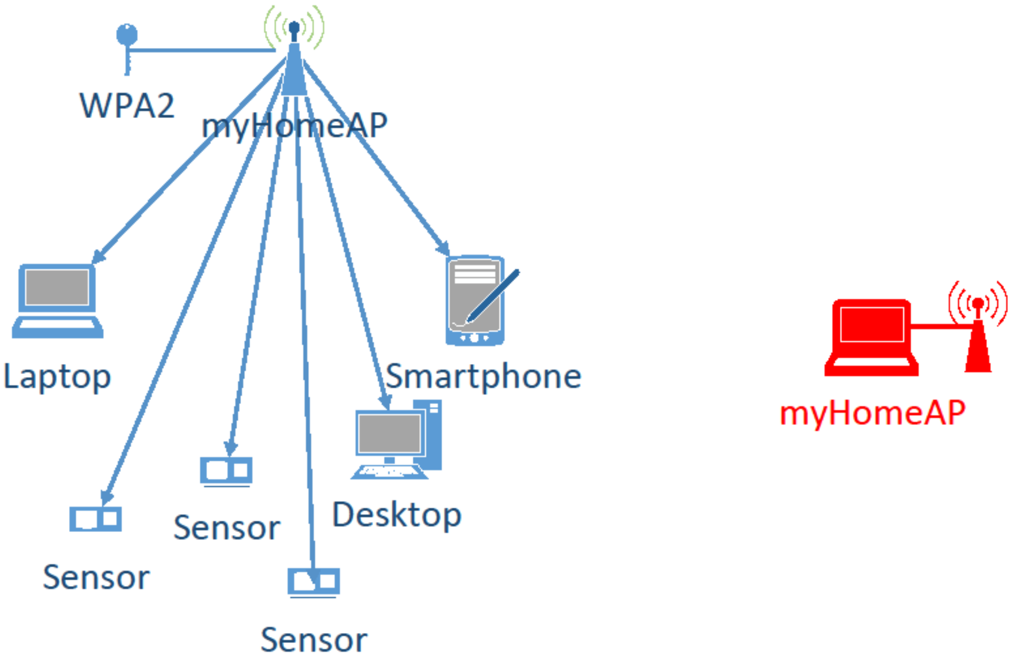
- Inexpensive Arduino Uno board
- Sensors
  - Moisture
  - Temperature
- Actuator
  - Water valve

Use Case: Automated Watering System



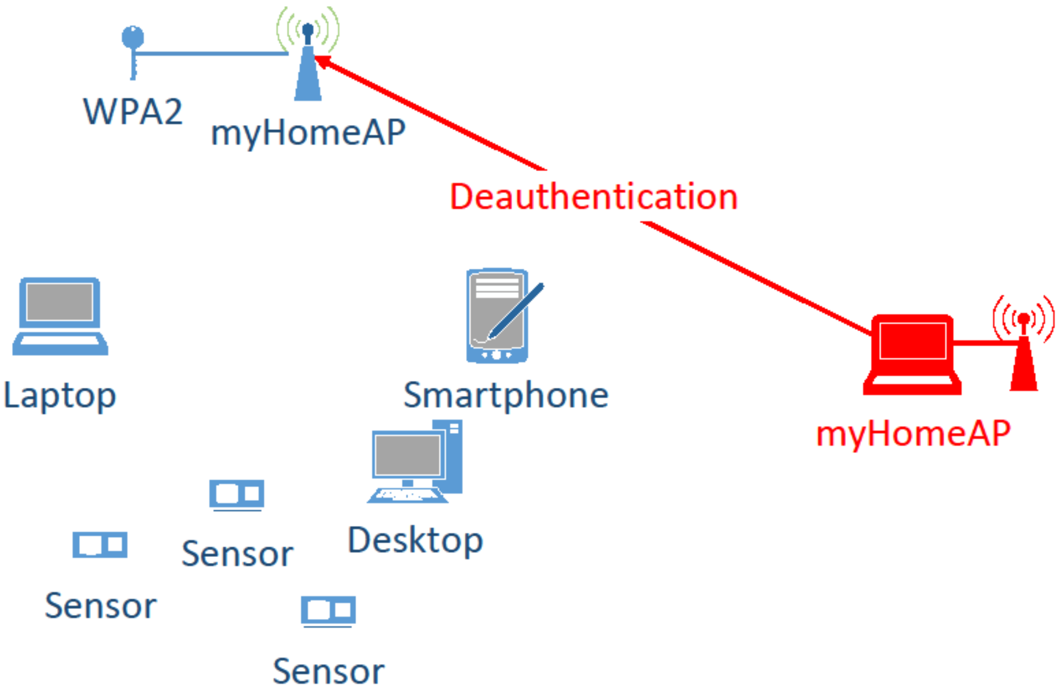
- WiFi “Shield” connects to home network
- Transmits all information to a custom web application
- A user monitors the status of his plant
- If he judges he can issue a command to enable the water valve

What Can Go Wrong?



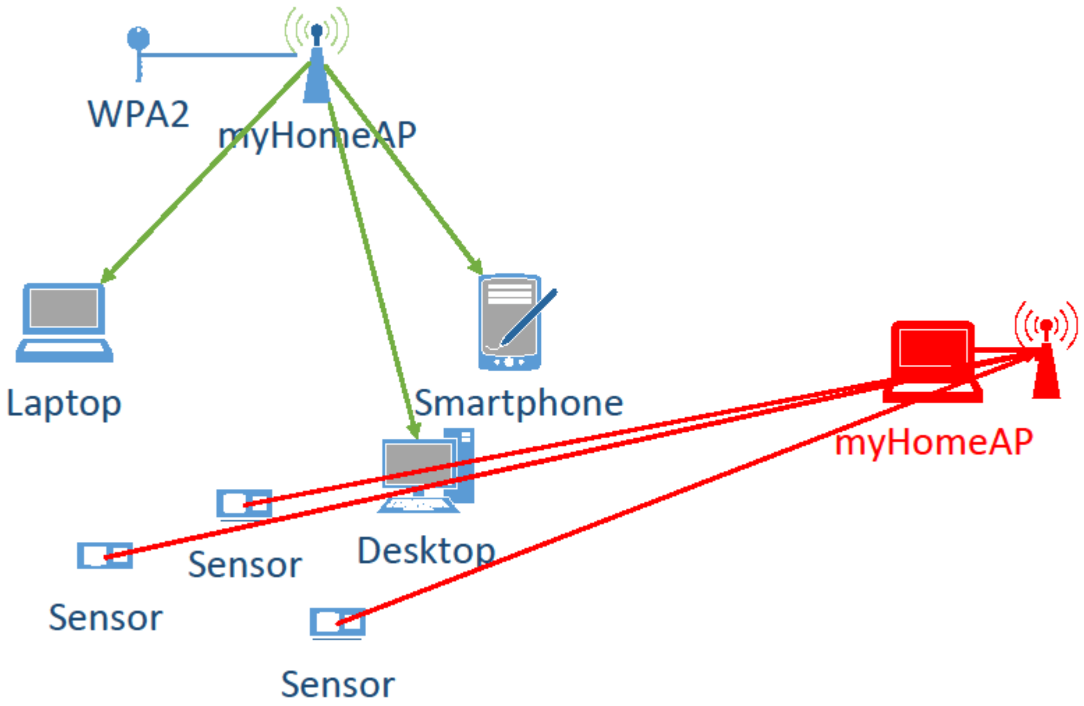
- Attacker introduces a soft-AP with the same characteristics
- No protection

What Can Go Wrong?



- The attacker issues a deauthentication packet
  - Does not have to be associated with the valid network
  - Does not need to know its key
- All clients loose connectivity momentarily

What Can Go Wrong?



- All devices will attempt to connect to the AP with the stronger signal
- Stronger devices will realize that something has changed
  - Protection
- Small sensor do not have "known-AP lists"
  - They will connect to attacker
- Attacker will be able to see all unencrypted traffic

Clear need for Encryption on the Communications!

Why Can Go Wrong?

Sensor data are treated as “non-sensitive”

Example: Transmission of temperature from sensor to cloud service

  - First glance: no leak of private user information

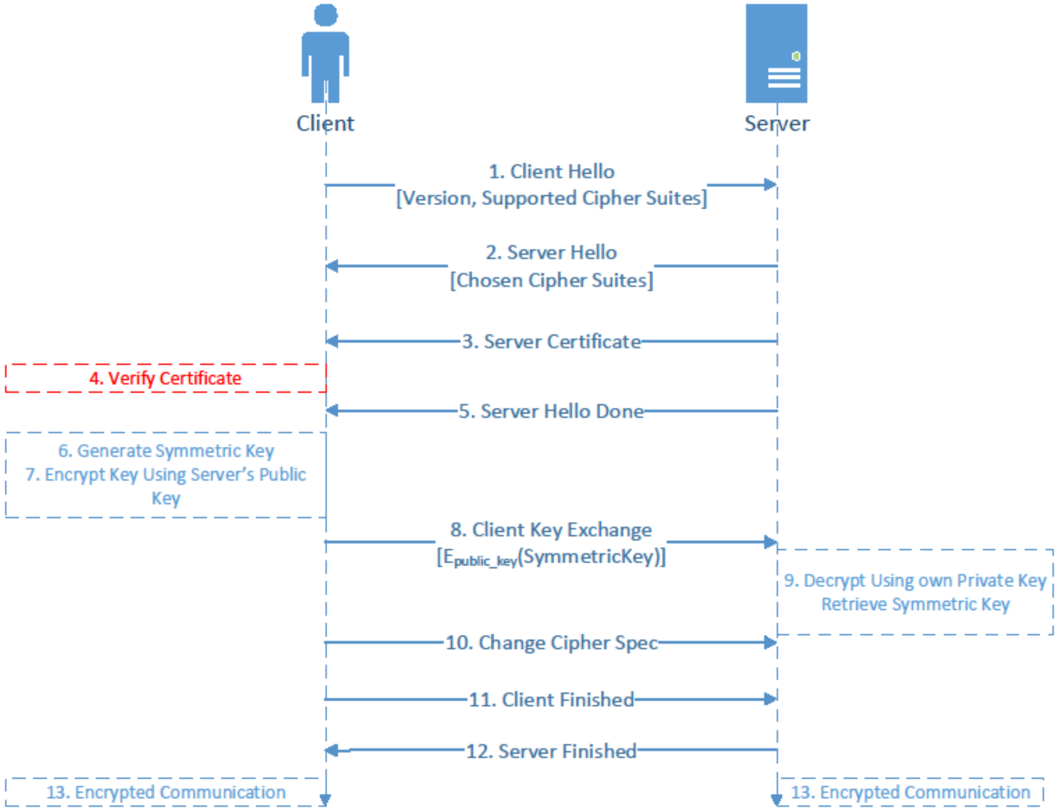
- Data Inference based on rapid changes in temperature
  - Expose human presence
  - Expose location (temperature changes occurring outside)
  - Expose crop requirements (or type)

Why Can Go Wrong?

- **Badly Designed System**
  - Platform that cannot handle encryption (SSL/TLS)
  - Cannot communicate securely with standard servers

- **Badly Implemented Crypto**
  - **Example**: Implement “*Custom*” TLS for “*faster*” operation
  - **Challenge**: Make TLS lighter but maintain compatibility
  - **Method**: Remove the “*heaviest*” operations
    - First contender: verification of server certificate
  - **Result**: Minimalistic hardware can support TLS
  - **Gain**: Use of even cheaper hardware
    - Caveat: possible security holes

How Can Go Wrong?

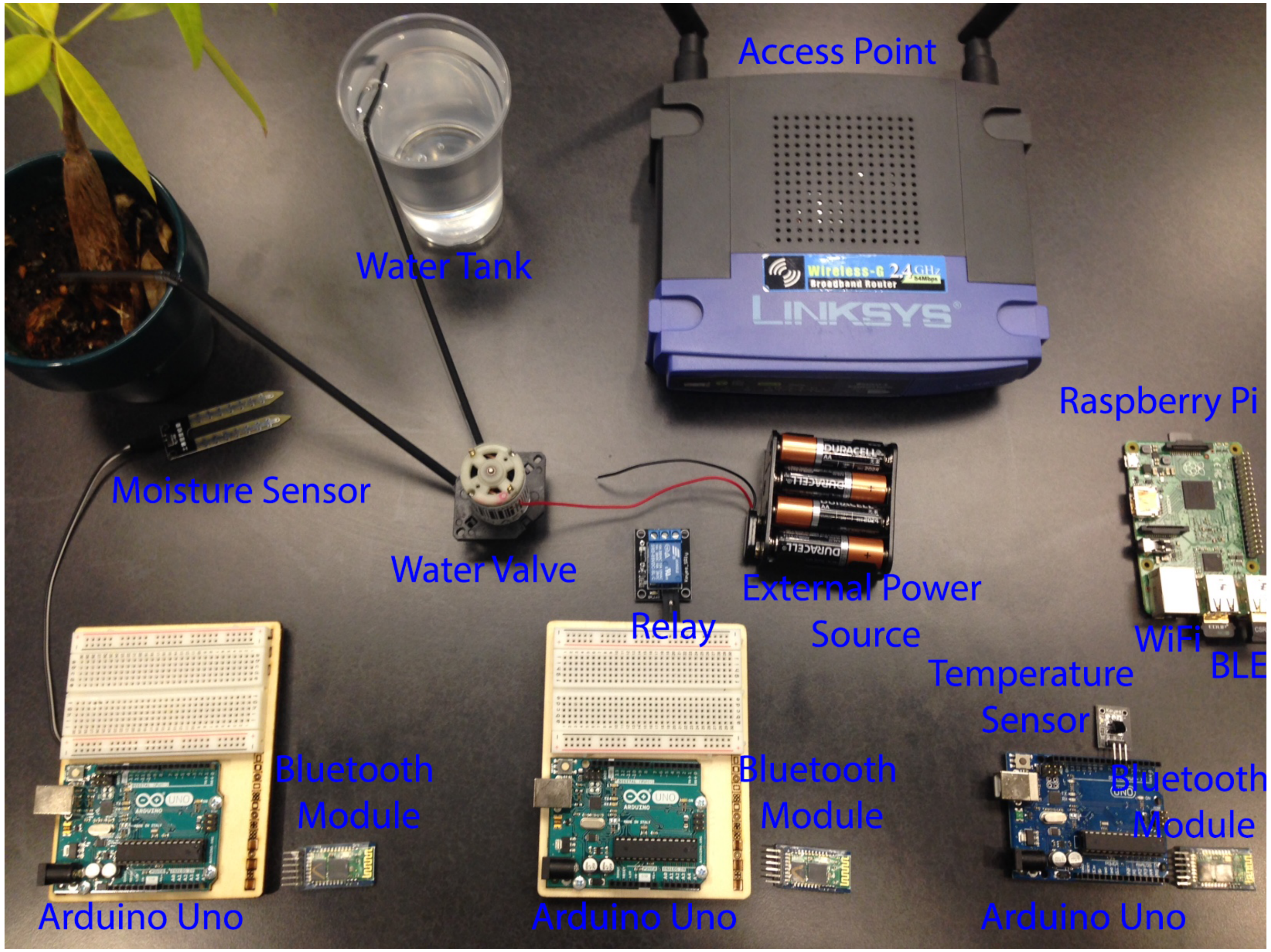


Protocol Hacked at Def Con 2015

- Connects to google calendar to show notes on screen
- Supports SSL/TLS but does not validate server certificates
- Unleash MiM attack
- Steal user's Google credentials

Custom Crypto Implementation not a solution

Possible Solution



- Introduce a “gateway” device
  - Can be inexpensive
  - Can support SSL/TLS
- Break complex devices to simple sensors and actuators
  - Inexpensive equipment
- All traffic is forwarded by the gateway
- Sensor can connect to the “gateway” via Bluetooth
  - Smaller range
- All traffic transmitted to the Internet is encrypted

New Design Advantages

- Eliminate Remote Control & Commands
  - Decisions are made locally
- All data transmitted to the Internet is protected with TLS
  - Raspberry Pi 2 can support SSL/TLS
- All data transmitted locally is encrypted
  - AES 256 → 0.86 ms (small overhead)
- Cost is similar to the original deployment
- Scales when many sensors/actuators are involved
  - Can support many different protocols in the local nodes

Insecure Services Running on the Network

- 3rd most critical vulnerability in IoT (OWASP)
- Having unnecessary open ports on devices
- Services that are vulnerable to buffer overflow attacks
- “Permissive” protocols
    - Universal Plug and Play

Use Case: Automatic Power Appliances

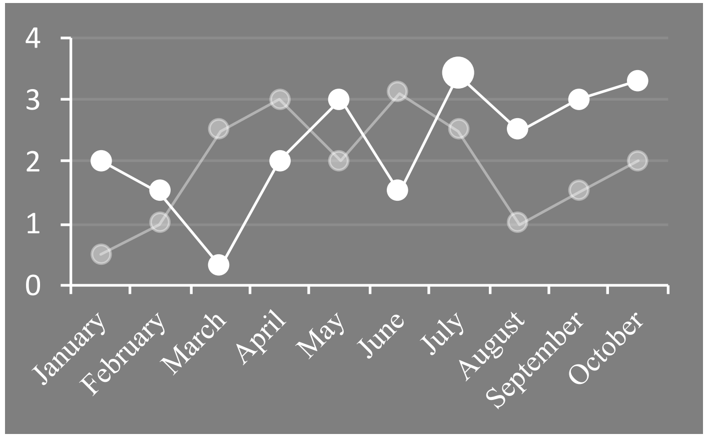
- Motivation: Create another layer of safety against home hazards

- Combine the capabilities of commercial products

- Achieve automatic turn off of “dangerous” appliances when a user sleeps

Our Vision

- ✓ **Identify IoT specific vulnerabilities.**
- ✓ **Study the behavior of systems under attack.**
- ✓ **Recommend a set of good practices**



- ✓ **How secure is an inspected IoT system**
- ✓ **Under what conditions renders insecure**
- ✓ **What are the outcomes of a security breach**

Conclusions

- IoT Security and Reliability still a challenging open problem

- Scale, Vendors, Technologies increase exponentially

- Lack of Standards or Best Practices available
  - Usability & Deployment the primary drivers
  - Interoperability & Reliability and afterthought
  - Security & Privacy not a primary design tenet

- Industry tries to fill the void but not very successfully

Questions?

Angelos Stavrou
astavrou@gmu.edu