Research Points

Article: Blockchain IoT (BIoT): A New Direction for Solving Internet of Things Security and Trust Issues.

Introduction

* IoT is a network made of connected things…
  + CPU, Memory (RAM, ROM, EEPROM, Flash) I/O buses.
  + Network interface (Wi-Fi, Bluetooth, Zigbee)
  + Sensors/Actuators
    - <https://dzone.com/articles/iot-systems-sensors-and-actuators>
    - any physical device that converts one form of energy into another. A microphone is a sensor that takes vibrational energy (sound waves) and converts it to electrical energy in a useful way for other components in the system to correlate back to the original sound.
    - actuator operates in the reverse direction of a sensor. It takes electrical input and turns it into physical action
      * Example: Electric motor, hydraulic system and a pneumatic system…
        + IoT turn on light.
* IoT requirements:
  + Secure Communication: strong, mutual authentication between end-entities, privacy, and integrity for exchanged information
  + Secure storage: for secret values used by communication. Enforced by tamper resistant devices such as secure elements.
    - <https://www.securetechalliance.org/resources/webinars/Secure_Elements_101_FINAL3_032813.pdf>
    - Secure elements are tamper resistant Smart Card chip that facilitates the secure storage and transaction of payment and other sensitive credentials…
      * Smart cards are plastic cards with built in microprocessors…control access to resource it is a part of.
      * Found in credit cards…
  + Node Integrity: secure updates and secure boot. Physical technologies dealing with multi processors or logical technologies such as sandbox may effectively contribute to increase resistance to intrusion.

IoT Frameworks & Security

* Security enforced by layers
  + Operating System: Software integrity & secure updates
  + MAC security features
    - IEEE 802.1 li for Wifi
    - Bluetooth paring
    - Uniqute Master Key & shared link key for Zigbee
    - AES Keyf or Lora clients…
    - HMAC client Key for SigFox
  + TLS/DTLS stacks as identity modules and dealing with symmetric/asymmetric credentials.
    - Transport Layer Security/Datagram Transport Layer security
  + Application features, Performing Authentication features, enforcing message privacy and integrity
* Question for Malik Luti:
  + Would this be translatable to the TCP/IP Model as we see there are Physical?, Data link or network, Transport and application level equivalents…

Trust For IoT

* There were attacks demonstrated against smart bulbs able to download embedded software “over the air” \*meaning wireless\*
* Malicious software update via:
  + Zigbee implementation bug. (Network Interface)
  + recovery of AES key used for update procedures, shared by a community of bulbs. Performed through side channel attack.
  + **Implications: Remotely hijack the functionality of your IoT device for other purposes…**
* TLS/DTLS stacks are used in IoT for identity layer. To avoid object hijack, replace TLS/DTLS identity stacks with secure elements to run TLS/DTLS stacks.
  + Secure elements improve tamper resistance.
* Figure displaying a setup. Uses Four quarter architecture discussed in other paper.
  + **General Purpose Unit**: Main process that runs tiny operating system or only necessary software. Update operations secured through cryptographic means.
  + **Radio SoC** (System on a chip) for connection and MAC security. Asymmetri cryptography. and also has procedures for software updates.
    - **Class Tie in. MAC addresses and the Data link layer.**
  + **Identity module:** secure element with a TLS stack. (javacard or something usually supporting Java Virtual Machine [JVM])
    - Tamper resistant identity module.
    - **For Malik Luti: Is there any insight on the implementation? or we staying general?**
  + Sensors and Actuators. Authorization mechaisms optionally supported.
* Point of the four quarter architecture system?
  + Secure wireless software updates for the GPU and radio SoC
    - Does NOT secure local updates.
  + Dependencies
    - Availability of secure download operations.
      * Symmetric secret key storage and use.

Blockchain IoT:

* Main features:
  + Consensus Algorithm to build blocks; involved entities check transaction coherence and authenticity.
  + Remunerated (requires pay) competition for block certification and computation of a new branch. Proof of stake and Proof of worth algorithms…
    - Proof of Work and Proof of stake
  + Lack of 3rd trusted party:
    - **Take away: Once 3rd party is compromised, every application that uses the system must be redone.**
  + Ledger (collection or database) components stored in widely distributed database
    - **Take Away: in addition to creating more copies to check for coherence and authenticity… if one database goes down system does not crash as a whole…**
* Main services:
  + storage of sensor data:
    - Current: stored in cloud. Usually without authentication and without \*non repudation (denial/rejection) services
  + Benefits include:
    - publication & duplication without tamper
    - authentication
    - timestamps of sensor data…

Article: NeuroMesh: IoT Security Enabled by a Blockchain Powered Botnet Vaccine

Abstract:

* concerns: Security
  + small memory capacity
  + low power processors

Introduction: cybersecurity crisis and statistics on IoT usage.

IOT security issues.

* October 2016, hackers launched large scale DDoS attack using a botnet and leveraging Linux based IoT devices that had been infected with Mirai malware.
  + 3 waves of attacks
    - Github, Amazon, Slack, Visa, HBO
  + Cause? IoT devices are easily hacked.
  + 11 year old boy demonstrated in 2017 how to hack into conference Bluetooth devices with a raspberry pi.
    - penetration tester also gained access to an internet connected teddy bear controlled a message…
  + Many systems are exposed through **open source messaging protocol known as MQTT**
    - protocol not the issue.
    - proper device security provisioning is inconsistent in IoT
  + Conclusion IoT is vulnerable.

Where current IoT systems Fail.

* inconsistency. (take away: Importance of protocol and standards.
  + IoT device firmware
  + Usage of open-source Embedded Linux OS
* Lack of a decisive decision on standards…

Embedded OS

* Linux is dominant for mobile operating and embedded systems.
* Hacker benefits: relative ease of customization.
  + Flaw in TCP allowed hackers to inject malware by hijacking internet traffic.
    - BIOS password bypassing.
    - Data network security weakness
    - Faking device entities
    - DNS spoofing
    - Linux security tool leveraging.

Routers and switches.

* reference to a heist in Bangladesh.
* Implementation of second hand, $10 routers that did not have appropriate firewall or any file system security was an issue….
* Routers are always “on”
  + open source firmware with default usernames and passwords…
  + Many use Universal Plug-inPlay protocol…
    - automatically open ports for data transfer.
* Hackers take control: install malware and launch a DDoS attack.

Hardware Network monitoring.

* Efforts to prevent cyberattacks: hardware devices that monitor network traffic…
  + analyze packets. and flag activity…
* Hacker response: zero-day attack that can defeat hardware firewall in place
* Other means to bypass firewall rules.
* Hardware firewalls do not prevent installation of malicious code on a vulnerable device after hacker gains access…
  + Network monitoring hardware cannot quickly kill malicious code installed on vulnerable nodes…

Botnets

* Mirai botnet attack had 7 steps

1. uncover default credentials of weakly configure dIoT devices with 62 likely username/password pairs hardcoded.
2. Bot forwards device characteristics to report server using different port.
3. Through command and control server, botmaster locates new targets by continuous communication with report server and through Tor
4. Once targets are found, botmaster deploys infected command in the loader which holds all required details: hardware infrastructure and IP addresses
5. Loader logs into target device and directs it to download and implement malware.
   1. Launches self-protecting script that prevents other malware from entry…
   2. Mirai can now communicate with the Command and Control server.
   3. Device ready to receive attack commands.
6. Target server identified for attack. botmaster commands launch of attack using CnC server.
7. using one of ten available attack variants, bots begin attacking the target server…

* Hard to guard against…developed using Mirai as functional model.

Neuromesh solution:

* Neuronode endopoint protection, Rendezvous servers, Neurocloud command and control server, and NeuroPrime security operations center (SoC)
* Uses Neural and mesh networks to secure IoT devices…
  + proactively detects and remove IoT device malware.
  + blacklists/whitelists IPBAC
  + Enables secure communications and updates to IoT devices over the Bitcoin communication protocol…
* Solution is based on a botnet, ironically a vaccine

NeuroMesh Botnet Communication Protocol:

* Traditional botnet architecture has a central point of failure…
  + command center is discovered? could be taken over.
* Rely on Peer to Peer infrastructure:
  + Implement kademlia protocol
    - Use distributed hash table and P2P bot communication to disseminate commands.
    - Vulnerability: bot tkane over? Attacker access DHT with information on all other bots
      * Know all other members? begin to Poison.
* \***Reference to previous paper called ZombieCoin to propose botnet operated over bitoin blockchain\* fails because of constraitns of IoT technology. AKA Memory capacity.**
* Neuromesh builds on it. Establish IoT version to use bitcoin block chain to disseminate commands out to IoT bots while maintaining integrity/security.

Endpoint security:

* NeuroNode
  + 32kB bot-like bash script
  + 1MB simplified payment verification(**SPV)** Bitcoin blockchain node.
    - Bash script reads and executes security commands from the node.
    - Bash script contains pre-stocked IoT malware signatures…
      * scan for malware signatures and destroy them.
      * **Take away from CECS 378 this is very much similar to our antivirus programs that have a record of file signatures and act accordingly.**
  + Download/installation methods:
    - direct firmware flash
    - cloud based deployment e.g. AWS Greengrass
  + Upon download:
    - begin antimalware like process
    - create system monitor per individual device in botnet…
      * system monitor will assume that current processes and post malware destruction are sanctioned and will communicate with NeuroMesh that said processes are to be whitelisted…
        + Updates to whitelist are added through blockchain secure communication protocol…
      * Continuously scan and kill any unauthorized processes.
      * Functions without network connectivity.
      * Evaluate existing ports in use and block all non essential ports. to prevent potential threats…
  + conceal its processes through distribution in various parts of file system.
* Upstream transmission of network traffic…
  + Rendezvous servers: NeuroNodes will transmit PCAP files to 3 or more different rendezvous servers.
    - Rendezvous servers are connected to form a mesh network with IoT devices and Neurocloud.
  + Function: prevent NeuroNodes from discovering the CNC server that analyzes network traffic.
  + Solves the concern of. Traditional botnet architecture has a central point of failure…
    - command center is discovered? could be taken over.
  + More conversation on securing it…

Neurocloud: threat detection and communication…

* Neurocloud: is CnC server
  + stores + analyzes network traffic
  + posts security updates to blockchain…
* Data collection and analysis:
  + CnC aggregates traffic data from rendezvous point.
  + Use machine learning to determine malicious IP addresses.
    - Communicate threat info over blockchain communication protocol
      * Notifies all network devices of threat.
* Communicating updates to IoT devices…
  + Use public bitcoin blockchain as communication protocol…
    - Outsourcing reduces costs.
    - Guarantees data integrity…
    - **why this protocol? widespread implementation and test of time.**
      * **theoretical 51% attack… to manipulate consensus.**
  + Hash security commands to bitcoin blockchain…

Article: SURE-H: A Secure IoT Enabled Smart Home System

Introduction:

* Traditional surveillance and security systems of CCTV are costly and also do not inform owners automatically.
* ESP8266 system is better in terms of resolution and low power consumption using Passive infrared (PIR) sensors.
  + PIR sensor detects motion. Info is stored in module.
  + IoT application can remotely monitor activity.
* M2M interaction is a new business concept… more beneficial than systems requiring human intervention.
* **Goal:** secure IoT enabled smart home system
  + **increase safeness from theft**
  + **saves enormous power cost.**

literature Review on smart home automation

Proposed system sure-H

* Cloud server
  + configured based on cloud service known as Blynk.
    - provides end-to-end solution for IoT based application development.
* hardware interface module
* software package or smartphone application…
* Design:
  + ESP8266-12E Module of relay board is programmed with Arduino
  + configured to receive/process specific command transmitted over internet.
  + Configure cloud server.
  + Design smartphone app and connect with wifi microcontroller.
* Features:
  + automated switches for all home appliances
  + capable of movement detection.
  + generate password by combining user password and fingerprints…
    - Based on stored cloud server data…
* Process: store details of appliances and room configuration. Send request to server…
  + motion objects will send alarm with detail report… when new object observes…

IP-Based IoT device detection

* 100k compromised IoT devices launched series DDos attakcks that set records in attack bitrates.
  + 620 Gb/s attack against KrebsOnSEcurity.com.
  + 1 Tb/s attack on French cloud-computing provider OVH.
* Understand threat requires:
  + knowledge of locations, distribution and growth of IoT devices
* Contribution
  + New method to detect IoT devices from observations of Internet traffic.

Methodology:

* IoT devices exchange traffic regularly with servers run by manufacturers.
  + servers identify IoT devices by watching traffic…
* Approach depends on
  + Identifying servers to look for device server names.
    - We know companies have their own Servers. Need to know DNS name.
    - Filtering server candidate names…
      * 3rd Party and human facing servers result in false positives for detection.
    - Handling shared dserver names.
    - Tracking server IP changes
  + Detecting IoT devices with device server IP’s