An Approach for Secure Edge Computing in the Internet of Things

Markus Endler, Anderson O. da Silva and Rafael A.M.S. Cruz
{endler, anderson}@inf.puc-rio.br and ramscrz@gmail.com

Laboratory for Advanced Collaboration
Departament of Informatics
PUC-Rio
Why criminals have a lot of interest in IoT?

Key points of IoT that can contribute to form a digital army:

- Processing capability:
  - Execute malicious software (malware).

- Storage capability:
  - Spread parts of data from crime among the things (P2P).

- Transmission capability:
  - Transmit malicious data (attack messages).

- Internet communication capability:
  - Targeted massive attacks over the Internet (high density botnets).
Criminals are already using IoT to increase their botnets and firepower

Symantec Official Blog

IoT devices being increasingly used for DDoS attacks
Malware is infesting a growing number of IoT devices, but their owners may be completely unaware of it.
By: Symantec Security Response
Created 22 Sep 2016

Mirai: what you need to know about the botnet behind recent major DDoS attacks
Botnet has grown by exploiting weak security on a range of IoT devices.
By: Symantec Security Response
Created 27 Oct 2016

A distributed denial of service attack (DDoS) on DNS provider Dyn last week managed to disrupt an array of the internet’s biggest websites, including Spotify, Twitter, and PayPal.

What was most interesting about this attack was that it was largely carried out using an Internet of Things (IoT) botnet called Mirai (Linux.Gafgyt).
Security Threats to IoT Systems

Threats in the IoT environment might be similar to those in the traditional IT environments, but...

- The overall impact could be very different because the targets are abundant and cover many different industry segments.
  - Currently, IoT technology already supports connection of millions of smart devices and meters.
  - By 2025, it shall support more than 50 billion connected devices.

- The potential impact could span from minor irritant to grave and significant damage to the infrastructure and loss of life.

What are the threats to the classic topology of IoT Systems?

IoT Classic Three-Layer Topology: Data Acquisition – Data Aggregation – Data Analysis
What are the threats to the generic topology of IoT Systems?

Threats that can compromise the security of IoT systems can be grouped into two distinct groups:

- **Group of Threats 1 (GT1)**
  - Threats to the operation of the entities of the IoT system.
    - Smart Objects, Gateway and Cloud

- **Group of Threats 2 (GT2)**
  - Threats to the communication between the entities of the IoT system.
    - Smart Objects – Gateway
    - Gateway – Cloud
Group of Threats 1 (GT1)

Threats of GT1, in general, aims to:

- Gain privileged or unprivileged access
- Tamper control information
- Tamper the firmware
- Produce false data
- Steal information
- Disrupt the system
Common threats of GT2: Monitoring the content of the messages (passive attack)
Group of Threats 2 (GT2)

Common threats of GT2: Intercepting and tampering messages (active attack)
Common threats of GT2: Masquerading (active attack)
Group of Threats 2 (GT2)

Common threats of GT2: Denial of Service (DoS) by Flooding (active attack)
What are the suitable security controls for IoT Systems?

In order to determine the need of a security control, we first have to analyze the security risks.

• This means we have to evaluate, for each risk, its likelihood to occur, technical impact and harm to the business or organization.

• Sometimes, we will accept the risk and choose simple low-cost controls or even no control at all.

• However, other times, we will face scenarios in which it would be irresponsible to choose such simple controls. For these, we have to invest more to acquire equipment aligned with our security needs.

What are the suitable security controls for IoT Systems?

Analysis of the security risks: IoT example scenarios

- In a **non-mission critical** scenario where we need to acquire data about the soil moisture or the environment temperature in order to keep the well-being of the plantation, we can accept the risk to use low-cost smart things with simple security controls.
What are the suitable security controls for IoT Systems?

Analysis of the security risks: IoT example scenarios

• In mission critical scenario, where we have to monitor the same kind of data related to the reactor of a nuclear power plant, we will eventually need special smart things with the necessary processing capabilities to implement classic and well-known high security standards.
Defeating targeted attacks on the cloud and on the gateway:
Defeating targeted attacks on the gateway and on the smart thing:
Defeating targeted attacks on the smart things:

- In order to reinforce the security in the smart things, we propose that these sensors/devices shall provide two distinct operating modes:
  - (i) configuration mode
    - Allows configuration actions such as the modification of operating parameters (e.g. signal strength, cryptographic keys, network address, authentication method) and updating of the firmware, among others.
  - (ii) service mode
    - Common operating mode in which the smart thing do what it is intended to do and allows data to be collected or changed.
- As a security measure, the smart thing shall use an access control method before switching modes, such as validating a PIN (Personal Identification Number).
Applying the Security Architecture to the ContextNet Middleware

ContextNet: Middleware for IoT Systems (developed at the Laboratory for Advanced Collaboration)

- Uses a scalable mobile-cloud communication layer, SDDL (Scalable Data Distribution Layer), plus the mobile component Mobile Hub, which is responsible for discovering and connecting Smart Objects to the Internet.

Introduces the concept of IoMT (Internet of Mobile Things)
Applying the Security Architecture to the ContextNet Middleware

ContextNet: Securing the Smart Object Service Broker (the cloud)
Applying the Security Architecture to the ContextNet Middleware

ContextNet: Securing the Mobile Hub (M-Hub), which is the ContextNet IoT Gateway
Applying the Security Architecture to the ContextNet Middleware

ContextNet: Securing the Mobile Hub (M-Hub) communication with SDDL Core and Smart Things

- For the proposed protocol, the following elements must be acknowledged:

  **Smart Thing / Object (S-Obj)**

  - Symmetric Authentication Keys (Kauth_s-obj)
  - Symmetric Cipher Key (Kcipher_s-obj)

  **Mobile Hub (M-Hub)**

  - Private Key (Kpriv_m-hub)
  - Public Key (Kpub_m-hub)

  **SDDL Core Gateway (SDDL-C GW)**

  - Private Key (Kpriv_sddl-c-gw)
  - Public Key (Kpub_sddl-c-gw)
  - Symmetric Authentication Key (Kauth_sddl-c-gw)

  **S-Objs Database**

  **Access Control Database**
Applying the Security Architecture to the ContextNet Middleware

ContextNet: Securing the Mobile Hub (M-Hub) communication with SDDL Core and Smart Things

- Step 1: Discovery Request Message: broadcast to all in-range S-Objs
Applying the Security Architecture to the ContextNet Middleware

ContextNet: Securing the Mobile Hub (M-Hub) communication with SDDL Core and Smart Things

- Step 2: Discovery Response Message: sent by in-range S-Obj to M-Hub

Discovery_Response(S-Obj_ID)
Applying the Security Architecture to the ContextNet Middleware

ContextNet: Securing the Mobile Hub (M-Hub) communication with SDDL Core and Smart Things

- Step 3: TLS Connection: M-Hub creates a Sec. Assoc. with SDDL-C

![Diagram showing TLS Handshake and Secure Association]

Bi-directional Secure Channel
Applying the Security Architecture to the ContextNet Middleware

ContextNet: Securing the Mobile Hub (M-Hub) communication with SDDL Core and Smart Things

- Step 4: Get Authorization: M-Hub sends message through TLS channel

\[\text{Get\_Authorization}(\text{S-Obj\_ID}, \text{M-Hub\_ID})\]
Applying the Security Architecture to the ContextNet Middleware

ContextNet: Securing the Mobile Hub (M-Hub) communication with SDDL Core and Smart Things

- Step 4.1: SDDL-C verify if the M-Hub is authorized to access the S-Obj

```plaintext
query(S-Obj_ID, M-Hub_ID)
result(True/False)
- checkAuthorization(S-Obj_ID, M-Hub_ID)
  - If (DB_Check_Access_Authorization(S-Obj_ID, M-Hub_ID)) Then
    - Return Go_To_Step_4.2
  - Else
    - Return Authorization_Error
```
Applying the Security Architecture to the ContextNet Middleware

ContextNet: Securing the Mobile Hub (M-Hub) communication with SDDL Core and Smart Things

- **Step 4.2:** SDDL-C gets S-Obj keys (Kauth_s-obj, Kcipher_s-obj) from DB

  - Query(S-Obj_ID)
  - Result(True/False, Kauth_s-obj_id, Kcipher_s-obj_id)

  - Get STKeys(S-Obj_ID)
    - If (DB_Get_Kauth_Kcipher(S-Obj_ID)) Then
      - Return Go_To_Step_4.3
    - Else
      - Return S-Obj_Keys_Query_Error
Applying the Security Architecture to the ContextNet Middleware

ContextNet: Securing the Mobile Hub (M-Hub) communication with SDDL Core and Smart Things

- Step 4.3: SDDL-C generates the OTPChallenge (random positive value)

  SDDL-C generates the OTPChallenge (random positive value)
  
  - GenerateOTPChallenge(nonce)
  - OTPChallenge = GeneratePositiveRandom(nonce)
  - Return OTPChallenge
  - Go To Step 4.4
Applying the Security Architecture to the ContextNet Middleware

ContextNet: Securing the Mobile Hub (M-Hub) communication with SDDL Core and Smart Things

- Step 4.4: SDDL-C generates the OTP value (seed of the key used by HMAC)

  - GenerateOTP(S-Obj_ID, M-Hub_ID, OTPChallenge, Kauth_s-obj)
  - OTP=Concat(S-Obj_ID, M-Hub_ID, Kauth_s-obj, OTPChallenge)
  - For i=1 To OTPChallenge Do
    - OTP=HASH(OTP)
  - Return OTP

  Go To Step 4.5
Applying the Security Architecture to the ContextNet Middleware

ContextNet: Securing the Mobile Hub (M-Hub) communication with SDDL Core and Smart Things

**Step 4.5: SDDL-C generates the session cipher key (Ksession)**

- GenerateKsession(nonce)
- Ksession=GenerateRandomKey(nonce)
- Return Ksession
- Go To Step 4.6
Applying the Security Architecture to the ContextNet Middleware

ContextNet: Securing the Mobile Hub (M-Hub) communication with SDDL Core and Smart Things

- Step 4.6: SDDL-C generates the secret package for S-Obj (Package_K)

  - GenerateST_PackageK(OTPChallenge, Ksession, Kchiper_s-obj)
  - Package=CreatePackage(OTPChallenge, Ksession)
  - Package_K=Encrypt(Package, Kcipher_s-obj)
  - Return Package_K

  Go To Step 4.7
Applying the Security Architecture to the ContextNet Middleware

ContextNet: Securing the Mobile Hub (M-Hub) communication with SDDL Core and Smart Things

- **Step 4.7: SDDL-C signs the secret package for S-Obj (Signed_Package_K)**

\[\text{S-Obj} \rightarrow \text{M-Hub} \rightarrow \text{SDDL-C GW} \rightarrow \text{HMAC} \]

- \(\text{SignST\_Package\_K(Package\_K, Kauth\_sddl-c)}\)
  - \(\text{Package\_K\_HMAC=}\) \(\text{GenerateHMAC(Package\_K, timestamp, Kauth\_sddl-c)}\)
  - Return \(\text{Package\_K\_HMAC}\)
  - Go To Step 5
Applying the Security Architecture to the ContextNet Middleware

ContextNet: Securing the Mobile Hub (M-Hub) communication with SDDL Core and Smart Things

• Step 5: SDDL-C sends OTP, Ksession and Package_K to the M-Hub

Authorization_Response(OTP, Ksession, S-Obj_ID, Package_K_With_HMAC)
Applying the Security Architecture to the ContextNet Middleware

ContextNet: Securing the Mobile Hub (M-Hub) communication with SDDL Core and Smart Things

- **Step 5.1: M-Hub stores Ksession and OTP for the S-Obj_ID**

  - StoreKey(S-Obj_ID, Ksession, OTP)
    - If (DB_Insert_Key(S-Obj_ID, ksession, OTP)) Then
      - Return Go_To_Step_5.2
    - Else
      - Return Ksession_OTP_Insert_Error
Applying the Security Architecture to the ContextNet Middleware

ContextNet: Securing the Mobile Hub (M-Hub) communication with SDDL Core and Smart Things

- Step 5.2: M-Hub signs the Hello Message with HMAC (OTP is seed of key)

  - **SignHelloMessage**(*M-Hub_ID*, **OTP**, **timestamp**, **Package_K_With_HMAC**)
    - **Kauth_m-hub**=Generate_M-Hub_Auth_Key(OTP)
    - **Hello_Message_HMAC**=GenerateHMAC(M-Hub_ID, timestamp, Kauth_m-hub, Package_K_With_HMAC)
    - Return Hello_Message_HMAC
    - Go To Step 6
Applying the Security Architecture to the ContextNet Middleware

ContextNet: Securing the Mobile Hub (M-Hub) communication with SDDL Core and Smart Things

- Step 6: M-Hub sends a signed Hello Message (HMAC) to S-Obj

Hello(M-Hub_ID, timestamp, HelloMessageHMAC, Package_K_With_HMAC)
Applying the Security Architecture to the ContextNet Middleware

ContextNet: Securing the Mobile Hub (M-Hub) communication with SDDL Core and Smart Things

- **Step 6.1:** S-Obj checks the signature of the Package_K_With_HMAC

  ![Diagram of S-Obj checking signature of Package_K_With_HMAC]

  ```
  CheckSignForPackage(Package_K_With_HMAC, Kauth_sddl-c)
  - If (CheckSign(Package_K_With_HMAC, Kauth_sddl-c)) Then
    - Return Go_To_Step_6.2
  - Else
    - Return Package_K_Signature_Error
  ```
Applying the Security Architecture to the ContextNet Middleware

ContextNet: Securing the Mobile Hub (M-Hub) communication with SDDL Core and Smart Things

- **Step 6.2: S-Obj decrypts Package_K to get Ksession and OTPChallenge**

  - DecryptPackage(Package_K, Kcipher_s-obj)
    - Package=Decrypt(Package_K, Kcipher_s-obj)
    - Return Package
    - Go To Step 6.3
Applying the Security Architecture to the ContextNet Middleware

ContextNet: Securing the Mobile Hub (M-Hub) communication with SDDL Core and Smart Things

• Step 6.3: S-Obj generates the OTP value (seed of the key used by HMAC)

- GenerateOTP(S-Obj_ID, M-Hub_ID, OTPChallenge, Kauth_s-obj)
  - OTP=Concat(S-Obj_ID, M-Hub_ID, Kauth_s-obj, OTPChallenge)
  - For i=1 To OTPChallenge Do
    - OTP=HASH(OTP)
- Return OTP
- Go To Step 6.4
Applying the Security Architecture to the ContextNet Middleware

**ContextNet: Securing the Mobile Hub (M-Hub) communication with SDDL Core and Smart Things**

- **Step 6.4:** S-Obj checks the signature of the Hello Message sent by M-Hub

  - CheckSignForHelloMessage(M-Hub_ID, OTP, HelloMessage)
    - Kauth_m-hub=Generate_M-Hub_Auth_Key(OTP)
    - Hello_Message_HMAC=GenerateHMAC(M-Hub_ID, timestamp, Kauth_m-hub, HelloMessage)
    - If (CheckSignature(HMAC, New_HMAC) Then
      - Return Go_To_Step_6.5
    - Else
      - Return Hello_Message_Signature_Error
Applying the Security Architecture to the ContextNet Middleware

ContextNet: Securing the Mobile Hub (M-Hub) communication with SDDL Core and Smart Things

- Step 6.5: S-Obj stores Ksession and OTP for the M-Hub

  Insert(M-Hub_ID, Ksession, OTP)
  Result(True/False)

  - StoreKey(M-Hub_ID, Ksession, OTP)
    - If (DB_Insert_Key(M-Hub_ID, ksession, OTP)) Then
      - Return Go_To_Step_6.6
    - Else
      - Return Ksession_OTP_Insert_Error
Applying the Security Architecture to the ContextNet Middleware

ContextNet: Securing the Mobile Hub (M-Hub) communication with SDDL Core and Smart Things

- **Step 6.6:** S-Obj signs the Hello Accepted Message with HMAC

  - `signHelloAcceptedMessage(S-Obj_ID, M-Hub_ID, OTP, timestamp)`
  - `Kauth_m-hub=Generate_M-Hub_Auth_Key(OTP)`
  - `Hello_Accepted_Message_HMAC=GenerateHMAC(S-Obj_ID, M-Hub_ID, timestamp, Kauth_m-hub)`
  - Return `Hello_Accepted_Message_HMAC`
  - Go To Step 7
Applying the Security Architecture to the ContextNet Middleware

ContextNet: Securing the Mobile Hub (M-Hub) communication with SDDL Core and Smart Things

- Step 7: S-Obj sends the signed Hello Accepted Message (HMAC) to M-Hub
Applying the Security Architecture to the ContextNet Middleware

ContextNet: Securing the Mobile Hub (M-Hub) communication with SDDL Core and Smart Things

- **Step 7.1:** M-Hub checks the signature of the Hello Accepted Message

  - CheckSignForHelloAcceptedMessage(M-Hub_ID, OTP, HelloAcceptedMessage)
    - Kauth_m-hub=Generate_M-Hub_Auth_Key(OTP)
    - Hello_Message_HMAC=GenerateHMAC(M-Hub_ID, timestamp, Kauth_m-hub, HelloAcceptedMessage)
    - If (CheckSignature(HMAC, New_HMAC) Then
      - Return Security_Association_Established_With_S-Obj
    - Else Return Hello_Accepted_Message_Signature_Error
Requirements to apply our security architecture to IoT Systems:

- Smart Things shall provide the necessary processing capability to execute, at least, basic cryptographic algorithms, such as, HMAC and RC4.

- Smart Things shall provide minimum amount of memory to store the session cipher key (Ksession) and the M-Hub authentication key (OTP).

- M-Hub shall support some kind of VPN protocol based on IPsec or TLS.

- M-Hub shall provide the necessary processing capability to execute the smart thing control service (ex: SNMP-based) and the access control service (Ex: local or remote, such as LDAP or Radius).
Conclusion

Advantages of the ContextNet and our security architecture to IoT Systems:

• The Smart Object may be stationary or mobile (IoMT).

• The IoT gateway (M-Hub) may also be stationary or mobile and may communicate to any Smart Object within wireless range.

• The Smart Object handover between gateways is also possible.

• A Smart Object can be handled by more than one M-Hub at the same time.

• The SSDL Core generates different cryptographic and authentication session keys on the fly for each pair of Smart Object and M-Hub.
Thank you!

Markus Endler, Anderson O. da Silva and Rafael A.M.S. Cruz
{endler, anderson}@inf.puc-rio.br and ramscrz@gmail.com

Laboratory for Advanced Collaboration
Department of Informatics
PUC-Rio
Related Work

Discussed in the paper:

  • Focus on smart things and study the IoT network security issues in the smart home, health care and transportation domain, and then present a taxonomy of security attacks.

  • Also addresses security of smart things but focuses on how data about/from smart things can be made secure using cryptography.
Related Work

Discussed in the paper:

  • The paper presents a general threat model that can be used to develop a security protection methodology for IoT services against cyber-attacks and shows that an Anomaly Behavior Analysis (ABA) Intrusion Detection System (ABA-IDS) can detect and classify a wide range of attacks against IoT sensors.
Related Work

Discussed in the paper:

  
  - The paper presents an analysis on the current status and concerns of IoT security, and proposes some countermeasures such as authentication measure, trust establishment, federated architecture and security awareness to reinforce security.

  
  - The vulnerabilities of Gateways and Edge networks are presented, and in this scope the author discusses three specific security threats: Network exposure of the Smart Objects, Man-in-the-Middle (MitM) and Impostor Attacks.