Workgroup: IOT Operations Working Group
Internet-Draft: draft-morais-iotops-inxu-01
Published: 26 May 2022
Intended Status: Standards Track
Expires: 27 November 2022
Authors: S.V. Morais C.M. Farias
IFRN UFRJ

Intra-Network eXposure analyzer Utility Specification

Abstract

This document proposes the Intra-Network eXposure analyzer Utility (INXU) as a vulnerability management solution for IoT networks. The goal of INXU is to take advantage of the functions of the RFC 8520 to allow a Security Experts Team on protecting multiple heterogeneous IoT networks, even when there is a few or none private information of the networks.

INXU identifies and analyzes the capability of an IoT device being exploited by an well known malicious activity. We also propose the Malicious Traffic Description (MTD), a data-model to describe traffic related to malicious activities.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at https://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on 27 November 2022.

Copyright Notice

Table of Contents

1. Introduction

While more Internet of Things (IoT) devices are deployed, the vulnerability management process turns even more difficult. This is mostly caused by the high heterogeneity and density of the IoT systems and devices, and challenges security teams on keeping Firewall and Intrusion Detection/Prevention Systems (IDS/IPS) rules up to date.

In some way, the Manufacturer Usage Description (MUD) [RFC8520] provides an alternative protection by reducing the capability of an IoT device being exploited -- as vector or target -- by malicious activities over the Internet. MUD does this by providing means to automatically build an allow-list of the IoT devices in a network based on the device manufacturers specification of expected traffic. This improves devices security by reducing their threat surface by blocking traffic with unexpected nodes/protocols, but still allows attacks which exploit vulnerabilities into the allowed traffic.

Besides this lack, the implementation of the [RFC8520] provides information that can support the identification of well-known vulnerabilities, as mentioned in its specification. This can be done by combining the allow-lists provided by the MUD manager into a communication graph of the connected IoT devices. With the communication graph, we can compare the traffic allowed by MUD with signatures of well-known malicious activities to identify -- and potentially block -- exposure of vulnerabilities into the network.

Integrating this analysis in traditional IDS or IPS can improve their efficacy and cover the MUD lack, but they only apply for scenarios where there is a security management team, such as corporate, smart grid and industrial IoT networks. On the other hand, in scenarios where there is a high heterogeneity of devices and low (or none) specialized support, such as in Home IoT Networks and Smart Cities, the process for keeping the attack signatures updated is not that simple.

Therefore, envisioning to overcome this gap, this document proposes INXU (Intra Network eXposure analyzer Utility) as a security tool

that takes advantage of the MUD-based network communication graph to prevent the exploitation of well-known vulnerabilities. To do this, INXU blocks threats on the Local Area Network (LAN) after identifying them by comparing the signature of well-known malicious activities with the traffic flow allowed by the MUD. In short words, while MUD builds allow-lists, INXU builds a blocklist on top of MUD's allow-lists.

The core component of INXU is Malicious Traffic Description (MTD), a document produced by a security specialist that describes ongoing malicious activities and well-known vulnerabilities and helps INXU find chains of connected IoT devices that can expose them to a threat. On top of MUD's threat surface reduction, INXU adds another security layer that enables protection against incidents not addressed or even caused by the manufacturers.

The MTD data model, as in MUD, abstracts network addresses to allow describing the traffic without the need to know the network's addressing schema or the connected devices. This resource allows creating portable descriptions of malicious traffic and protects the privacy in the LAN by not exposing private information to third parties in the security decision-making process. At the same time, it simplifies the sharing of knowledge about attacks between distinct networks.

Another relevant feature in INXU is its architecture that enables one Security Operation Center (SOC) to protect multiple distinct networks by sharing MTDs in a process similar to computer antivirus vaccines. This feature makes INXU a tool to protect LANs and the entire Internet ecosystem by making the operation of botnets and other attacks that affect the Internet's stability more difficult.

1.1. Simple Example

An Internet Service Provider (ISP) connects tens to hundreds of houses to the Internet. Each one of these homes contains a wide range of IoT devices connected in their internal networks, in diverse topologies, and with different usages by each end-user. By the variety of scenarios, these home networks potentially contain a few devices infected by a DDoS capable botnet.

Due to the attacks carried by this botnet, frequently the ISP has a considerable part of its traffic being consumed by DDoS attacks, and often the clients call helpdesk for problems with devices caused by the botnet. The ISP knows that the malware's infection occurs by a TCP/23 connection with a neighbour host, and the command and control occurs by a TCP/80 connection with a server located at mybotnet.example.com.

With this information, the ISP releases an MTD File describing this traffic, which can be used by its clients. In the home networks, the Customer Premises Equipment (CPE) collects the MTD File and compares it to the network communication graph provided by MUD, identifies exposures of vulnerabilities internally into the network, INXU evaluates the risk of the exposures and suggests blocks to prevent exploitations.

1.2. Key Aspects

This work in progress aims to propose a tool that reinforces IoT networks' security by taking advantage of the functions provided by the [RFC8520]. The specific contributions of INXU are listed below:

- Simplify the process of sharing attack signatures that targets or exploits IoT systems;
- Allow a small team of security specialists to protect multiple distinct IoT networks without expose the networks' privacy;
- Protect the Internet's ecosystem by hindering distributed attacks that targets its infrastructure.

1.3. INXU Intended Use

The intended use for INXU is in the support of the vulnerability management of diverse heterogeneous IoT networks in scenarios where there is a small team of security specialists (e.g. Smart Cities). It is also intended to be used in scenarios where the end networks need their privacy kept, as Home IoT networks.

The deployment of INXU in networks populated by both IoT and general purpose devices is NOT RECOMMENDED. Due to the greater computing power and wider openness to other attacks, general purpose devices might expose the IoT network to unnecessary risk. Instead of having both types on the same sub-network, we recomend to isolate IoT devices in a separate sub-network as they announce their MUD URLs, and developers should take advantage of MUD's "controller" and "my-controller" hosts as application gateway between general purpose and IoT devices. In the case of needing a direct communication between the two categories, this could be specified with MUD's "local-networks" specification.

1.4. Terminology

- INXU: Intra-Network eXposure analyzer Utility.
- INXU Module: a system that crosses data from malicious activities and MUD's allow-lists to identify and analyze the exposure of vulnerabilities in the connected IoT devices.

*MTD: Malicious Traffic Description data model.

*MTD File: a file that contains descriptions of traffic associated with malicious activities that targets or exploits IoT devices.

*MTD Manager: a system that requests and receives the MTD File. It is responsible for verifying MUD File's authenticity and integrity. The MTD Manager also requests the MTD File after it's cache validity expires.

*MTD URL: a URL, configured in the MTD Manager, that locates the MTD File provided by the SOC in charge to protect the client network.

*MTD Server: a web server, managed by the SOC, that hosts the MTD File.

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119][RFC8174] when, and only when, they appear in all capitals, as shown here.

2. The MTD Data Model

The main aim of this data model is to enable describing malicious traffic so that distinct networks can interpret and implement security measures, no matter the connected IoT devices or network topology. Another feature addressed by this data model is allowing the association between the detected exposure and the malicious activity that exploits it, as well as the grouping of vulnerabilities related to the same malicious activity.

The MTD data model makes use of Access Control Lists (ACLs) [RFC8519] under YANG language [RFC7950] to describe the malicious traffic, addressing the classification feature. Furthermore, such as in MUD, there are available two network address abstractions to describe the traffic so that different networks can adapt the description to its context: one abstraction for addresses in the local networks, and the other for using domain names to hosts on the Internet. The data model also includes control fields that support the manageability of the MTD File, so the contained data can be categorized in control data and description data.

This data model covers the complete description of malicious activities from simple attacks with just a few ACE inputs to complex malware that exploits many different vulnerabilities and network resources. Furthermore, to prevent false-positive threat detections, the MTD data model allows inserting context information into the descriptions. The context information in the MTD data model plays

the role of specifying the correlation between the described malicious traffic, determining the combinations of exposures that become a risk, and suggesting the action to be taken with each detected threat. This feature supports reducing false-positive detection, as a single traffic exposure may not represent a threat itself.

Basically the context information supports the categorization of a set of vulnerabilities as an effective threat or not. To incorporate the contextual information, we considered the statements listed below, which were assimilated from the IoTSec ontology in [Mozzaquatro2015]:

* A Threat represents an effective risk if the exposure of one or more vulnerabilities can be exploited by an attacker;

* A Vulnerability does not represent a risk by itself, as it can be hidden behind security mechanisms, such as blocking its exposure for potential attackers.

So, in short words, an asset is under threat only when an attacker can exploit one or more vulnerabilities to take advantage of it. Thus, merging the concepts from the ontology and the aims on MTD data model, this document considers the following statements:

* Each Access Control Entry (ACE) has an associated severity defined by the unsigned integer field named risk. When the exposure to the ACE is detected, its risk is considered part of its ACL's vulnerability classification;

* Each ACL has alert-threshold and risk-threshold fields, both represented by unsigned integer values. When the sum of the exposed ACEs risks reaches the risk threshold, the exposure to the ACL is considered as a vulnerability;

* Each malicious activity described contains a list of critical ACL sets. A malware or attack is classified as a threat when at least one set of critical ACLs contains all its ACLs classified as a vulnerability exposure. When one set's condition is satisfied, its associated action-to-take has to be triggered.

The three possible actions to be taken are listed below:

* block-all: blocks all ACLs that expose vulnerabilities related to the description. Expected to be used when any traffic associated with the malware or attack threatens the IoT device;

* block-attack: blocks all ACLs that expose vulnerabilities under the attack-traffic group. Expected to be used when only risky

ACLs associated with attacks (isolated or in the context of a malware) threatens the IoT device;

* block-not-attack: blocks all ACLs that expose vulnerabilities under the malware's not-attack-traffic group. Expected to be used when just blocking the operation traffic of a malware prevents exploitation.

2.1. The draft-inxu-mtd YANG Module

A simplified graphical representation of the data models is used in this document. The meaning of the symbols in these diagrams is explained in [RFC8340].

```
module: draft-inxu-mtd
  +--rw mtd!
     +--rw mtd-url                  inet:uri
     +--rw last-update              yang:date-and-time
     +--rw mtd-signature?           inet:uri
     +--rw cache-validity?          uint8
     +--rw malicious-descriptions
        +--rw malicious-list* [name]
           +--rw name                        string
           +--rw specific-devices*           inet:uri
           +--rw critical-acl-sets* [name]
           |  +--rw name                 string
           |  +--rw critical-acl-set*   -> /acl:acls/acl/name
           |  +--rw action-to-take      draft-inxu-mtd:action-to-take
           +--rw to-device-attacks
           |  +--rw traffic-lists
           |     +--rw traffic-list* [name]
           |        +--rw name                -> /acl:acls/acl/name
           |        +--rw specific-devices*   inet:uri
           +--rw from-device-attacks
           |  +--rw traffic-lists
           |     +--rw traffic-list* [name]
           |        +--rw name                -> /acl:acls/acl/name
           |        +--rw specific-devices*   inet:uri
           +--rw to-device-not-attacks
           |  +--rw traffic-lists
           |     +--rw traffic-list* [name]
           |        +--rw name                -> /acl:acls/acl/name
           |        +--rw specific-devices*   inet:uri
           +--rw from-device-not-attacks
              +--rw traffic-lists
                 +--rw traffic-list* [name]
                    +--rw name                -> /acl:acls/acl/name
                    +--rw specific-devices*   inet:uri

  augment /acl:acls/acl:acl/acl:aces/acl:ace/acl:matches:
    +--rw mtd
       +--rw local-networks?   empty
  augment /acl:acls/acl:acl/acl:aces/acl:ace:
    +--rw risk?   uint8
  augment /acl:acls/acl:acl:
    +--rw risk-threshold?    uint8
    +--rw alert-threshold?   uint8
  augment /acl:acls/acl:acl/acl:aces/acl:ace/acl:matches/acl:l4/acl:tcp/
  acl:tcp:
    +--rw direction-initiated?   mud:direction
  augment /acl:acls/acl:acl/acl:aces/acl:ace/acl:matches/acl:l3/
  acl:ipv4/acl:ipv4:
    +--rw src-dnsname?   inet:host
```

```
  +--rw dst-dnsname?   inet:host
augment /acl:acls/acl:acl/acl:aces/acl:ace/acl:matches/acl:l3/
acl:ipv6/acl:ipv6:
  +--rw src-dnsname?   inet:host
  +--rw dst-dnsname?   inet:host
```

2.2. MTD Data Model Definition of Control Fields in the Root "mtd" Container

Here we describe the leafs placed into the "mtd" root container that plays the role of controlling the operation of the MTD File.

2.2.1. mtd-url

Required field that stores the URL where the security authority hosts the MTD File.

2.2.2. mtd-signature

Optional field used to store a URL where the MTD File signature file can be found. It is applicable for offline authenticity verification of the file.

2.2.3. last-update

Required field that contains the timestamp information of the MTD File generation.

2.2.4. cache-validity

Optional field that contains the number of hours to the expiration of the MTD File, starting from "last-update". This field supports integer values between 1 and 160, and if not defined, it is assumed to be 48 hours by the MTD Manager.

2.3. MTD Data Model Definition of Traffic Description Fields in the Root "mtd" Container

The traffic description fields are organized under the "malicious-descriptions" container. The description of a malicious activity allows the aggregation of different attacks, and also other not attack traffic that only turn into malicious when related to the malware operation. This aggregation is important for the security measures decision-making process, as sometimes only a traffic combination makes the threat effective or blocking just one type of traffic can almost disable it, such as the Mirai's Command and Control traffic.

The description of each leaf is detailed in the Sub-Sections below.

2.3.1. traffic-list

List type field to specify all the traffic in the same direction (incoming/outgoing) that is associated with one attack-traffic or not-attack-traffic.

2.3.1.1. name

Required string field with the name of the ACL that describes one attack-traffic or not-attack-traffic;

2.3.1.2. specific-devices

Optional list to specify the MUD URLs of the IoT devices affected by the described traffic. When this field is filled, INXU only considers the devices here listed as targets of these ACLs.

2.3.2. malicious-descriptions

List that holds the traffic description of all the malicious activities covered by the MTD File.

2.3.2.1. name

Required string field to uniquely name the described malicious activity.

2.3.2.2. specific-devices

Optional list to specify the MUD URLs of the IoT devices that can be affected by the malicious activity. When this field is filled, INXU only considers the devices here listed as affected by this malicious activity.

2.3.2.3. critical-acl-sets

List to specify all the sets of critical ACL and their respective actions to take when all listed ACLs get classified as risky.

2.3.2.3.1. critical-acl-set

List to specify a set of ACLs that, when all listed ACLs get classified as risky, represents a threat caused by the malicious activity.

2.3.2.3.2. action-to-take

Mandatory leaf to specify the action to be taken when the respective set of critical ACLs turns into a threat. The action can be "block-all", "block-attack", or "block-not-attack".

2.3.2.4. to-device-attacks

Container that holds all the malicious activity's attack traffic targeting an IoT device on the LAN.

2.3.2.5. from-device-attacks

Container that holds all the malicious activity's attack traffic outgoing from an IoT device on the LAN.

2.3.2.6. not-attack-traffic

Container that holds the traffic not related to attacks, but that turns into malicious when in this context.

2.3.2.6.1. to-device-not-attack-traffic

List with all the ACLs that describe malicious not attack traffic targeting an IoT device on the LAN.

2.3.2.6.2. from-device-not-attack-traffic

List with all the ACLs that describe malicious not attack traffic outgoing from an IoT device on the LAN.

2.4. Augmentation to the ACL Model

This section describes the proposed augments to the ACL model. These augments are responsible for creating the abstraction for the traffic descriptions, enabling the portability of the knowledge to the different networks, and supporting the risk assessment of each vulnerability exposure.

2.4.1. mtd:local-networks

Optional leaf that, when present, means that the current ACE applies to any device on the local IP networks.

2.4.2. direction-initiated

Optional field incorporated from MUD to specify the TCP initiator.

2.4.3. src-dnsname and dst-dnsname

Optional field to enable the usage of DNS domain names to specify the remote host instead of using IPv4 or IPv6 addresses.

2.4.4. risk

Optional unsigned integer field to specify the risk associated with the exposure of the specified ACE. Its default value is 1.

2.4.5. risk-threshold

Optional unsigned integer field to specify the minimal ACL risk value to classify it as a vulnerability exposure. The ACL's risk

value is calculated by the sum of all its child ACE exposures' risks. Its default value is 1.

2.4.6. alert-threshold

Optional unsigned integer field to specify the minimal ACL risk value to trigger an alert to the exposure. The ACL's risk value is calculated by the sum of all its child ACE exposures' risks. Its default value is 1.

2.5. The MTD YANG Model

```
<CODE BEGINS>file "draft-inxu-mtd@2021-11-22.yang"
module draft-inxu-mtd{
  yang-version 1.1;
  namespace "urn:ietf:params:xml:ns:yang:draft-inxu-mtd";
  prefix draft-inxu-mtd;

  import ietf-yang-types {
    prefix yang;
  }

  import ietf-access-control-list {
    prefix acl;
  }

  import ietf-inet-types {
    prefix inet;
  }

  import ietf-mud {
    prefix mud;
  }

  import ietf-acldns {
    prefix acldns;
  }

  organization "IETF IOTOPS (IOT Operations) Working Group";
  contact
    "WG Web: http://tools.ietf.org/wg/iotops/
     WG List: iotops@ietf.org
     Author: Sávyo Morais
     savyovm@gmail.com
     Author: Claudio Farias
     cmicelifarias@gmail.com";

  description
    "This module is a data-model to describe malicious network
    traffic.

    This module is intended to be serialized via JSON and stored
    as a file, as described in I-D draft-morais-iotops-inxu-01.

    The key words 'MUST', 'MUST NOT', 'REQUIRED', 'SHALL', 'SHALL
    NOT', 'SHOULD', 'SHOULD NOT', 'RECOMMENDED', 'NOT RECOMMENDED',
    'MAY', and 'OPTIONAL' in this document are to be interpreted as
    described in BCP 14 (RFC 2119) (RFC 8174) when, and only when,
    they appear in all capitals, as shown here.

    Copyright (c) 2022 IETF Trust and the persons identified as
    authors of the code.  All rights reserved.
```

```
  Redistribution and use in source and binary forms, with or
  without modification, is permitted pursuant to, and subject
  to the license terms contained in, the Simplified BSD License
  set forth in Section 4.c of the IETF Trust's Legal Provisions
  Relating to IETF Documents
  (http://trustee.ietf.org/license-info).

  This version of this YANG module is part of
  I-D draft-morais-iotops-inxu-01; see the I-D itself for
  full legal notices.";

revision 2022-05-15{
  description
    "Simplifying the data model to one single description of
    malicious traffic.";
  reference
    "draft-morais-iotops-inxu-01: Intra-Network eXposure analyzer
                                  Utility Specification";
}

typedef action-to-take {

  type enumeration {
    enum "alert" {
      value 0;
      description
        "Alert user about a risky exposure";
    }

  description
    "Type to specify the action to take when a threat is detected";

    enum "block-not-attack" {
      value 1;
      description
        "Block risky exposures of not-attack-traffic and warns uses
        about attack-traffic alert exposures";
    }
    enum "block-attack" {
      value 2;
      description
        "Block attack-traffic risky exposures and alert users about
        the block";
    }
    enum "block-all" {
      value 3;
      description
        "Block all risky exposures and alert users about the block";
```

```
    }
  }
}

container mtd {
  presence "Enabled for this particular MTD URL";
  description "MTD-related information";
  uses mtd-groupig;
}

grouping mtd-groupig {
  description
    "This grouping is to create a set of definitions of
    malicious traffic of malware and attacks.";

  leaf mtd-url{
    type inet:uri;
    mandatory true;
    description
      "This is the MTD URL associated with the entry found
      in a MTD File";
  }

  leaf last-update {
    type yang:date-and-time;
    mandatory true;
    description
      "This is intended to be set when the current MTD File is
      generated. MTD Managers SHOULD NOT check for updates
      between this time plus cache validity.";
  }

  leaf mtd-signature {
    type inet:uri;
    description
      "A URI that resolves to a signature file to verify the
      autenticity of the MTD File.";
   }

  leaf cache-validity {
    type uint8 {
      range "1..168";
    }

    units "hours";
    default "48";
    description
      "The information retrieved from the MTD Server is
      valid for these many hours, after which it should be
```

```
      refreshed. MTD Manager implementations do not need to
       discard MTD Files beyond this period.";
  }

  container malicious-descriptions {
    description
      "This container has the descriptions of the malware and
      attacks that can exploit or target the devices";

    uses malicious-list;
  }
}

grouping traffic-lists {
  description
    "A grouping for acess lists of malicious traffic in the context
    of malware or attacks.";

  container traffic-lists {
    description
      "The access lists of the attack's malicious traffic
      targeting or departing from the local IoT devices.";

    list traffic-list {
      key "name";

      description
        "Each entry on this list refers to an malicious
        traffic defined by an ACL that should present the
         overall network communication of the attack.";

      leaf name {
        type leafref{
          path "/acl:acls/acl:acl/acl:name";
        }

        description
          "The name of the ACL for this entry.";
      }

      leaf-list specific-devices {
        type inet:uri;

        description
          "List of MUD URLs of specific devices
          related with the vulnerability";
      }
    }
  }
}
```

```
grouping malicious-list {
  description
    "A grouping for acess control lists of malicious traffic in the
    context of malware or attacks.";

  list malicious-list {
    key "name";

    description
      "The access lists of the malware's or attack's malicious
      traffic targeting or departing from the local IoT devices,";

    leaf name {
      type string;

      description
        "The unique name of the described malicious activity for
        each entry.";
    }

    leaf-list specific-devices {
      type inet:uri;

      description
        "List of MUD URLs of specific devices
        related with the vulnerability";
    }

    list critical-acl-sets{
      key "name";

      description
        "Each list entry represents a malicious activity's critical
        set of risky ACL exposures, followed by the action to take
        when a critical set be detected.";

      leaf name {
        type string;

        description
          "The critical ACL set name";
      }

      leaf-list critical-acl-set {
        type leafref{
          path "/acl:acls/acl:acl/acl:name";
        }

        description
```

```
          "A list to specify a set of ACLs that, when all listed
          ACLs get classified as risky, represents a threat caused
          by the malicious activity";
      }

      leaf action-to-take {
        type draft-inxu-mtd:action-to-take;
        mandatory true;
        description
          "A leaf to specify the action to be taken when
          the respective set of critical ACLs turns into
          a threat.";
      }
    }

    container to-device-attacks {
      description
        "The set of attack traffic performed by the
        infected IoT device";

      uses traffic-lists;
    }

    container from-device-attacks {
      description
        "The set of attack traffic performed targeting
        the infected IoT device";

      uses traffic-lists;
    }

    container to-device-not-attacks {
      description
        "The set of attack traffic performed by the
        infected IoT device";

      uses traffic-lists;
    }

    container from-device-not-attacks {
      description
        "The set of attack traffic performed targeting
        the infected IoT device";

      uses traffic-lists;
    }
  }
}

augment "/acl:acls/acl:acl/acl:aces/acl:ace/acl:matches" {
```

```
  description
    "adding abstraction to avoid the need of IP addresses.";
  container mtd {
    description
      "MTD-specific match.";
    leaf local-networks {
      type empty;
      description
        "IP addresses will match this node if they are
        considered local addresses. A local address may be
        a list of locally defined prefixes and masks
        that indicate a particular administrative scope.";
    }
  }
}

augment "/acl:acls/acl:acl/acl:aces/acl:ace" {
  description
    "Add the risk level information associated to the ACE";
  leaf risk {
    type uint8;
    default "1";
    description
      "Represents risk level of a device being exploited
      when exposes the device through traffic matching the
      described ACE.";
  }

}

augment "/acl:acls/acl:acl" {
  description
    "Add an acceptable risk threshold and an alert risk threshold
    to the ACL";
  leaf risk-threshold {
    type uint8;
    default "1";
    description
      "The acceptable risk threshold represents the minimum
      risk value for the exposure be considered a risk.
      The actual risk of an ACL is calculated by the sum of
      all the ACEs that matched on the INXU Module analysis";
  }

  leaf alert-threshold {
    type uint8;
    default "1";
    description
      "The acceptable alert threshold represents the minimum
```

```
        risk value for the exposure trigger an alert.
        The actual risk of an ACL is calculated by the sum of
        all the ACEs that matched on the INXU Module analysis";
    }
  }

  augment "/acl:acls/acl:acl/acl:aces/acl:ace/acl:matches"
    + "/acl:l4/acl:tcp/acl:tcp" {
    description
      "Add direction-initiated";
    leaf direction-initiated {
      type mud:direction;
      description
        "This node matches based on which direction a
        connection was initiated.";
       }
  }

  augment "/acl:acls/acl:acl/acl:aces/acl:ace/acl:matches"
    + "/acl:l3/acl:ipv4/acl:ipv4" {
    description
      "Adding domain names to matching.";
    uses acldns:dns-matches;
  }

  augment "/acl:acls/acl:acl/acl:aces/acl:ace/acl:matches"
    + "/acl:l3/acl:ipv6/acl:ipv6" {
    uses acldns:dns-matches;
    description
      "Adding domain names to matching.";
  }

  deviation "/acl:acls/acl:acl/acl:aces/acl:ace/acl:actions" {
    deviate not-supported;
    description
      "Field not used in this specification";
  }
}
<CODE ENDS>
```

2.6. MTD File Example

This MTD file describes the traffic of an hipotetic variant of the Mirai botnet. In its attack model, this malware scans for other vulnerable devices in the same network, and its management services (Command and Control, Scan Report, and Loader) are placed in the network edge.

<CODE BEGINS>file "mirai-lan-variant.json"

```
{
  "draft-inxu-mtd":"mtd",
  "mtd-url":"https://example.com/mirai-lan-variant.json",
  "last-update":"2022-05-15T18:17:00-03:00",
  "malicious-descriptions":{
    "malicious-list":[
      {
        "name":"Mirai-Example",
        "critical-acl-sets":[
          {
            "name":"mirai-prevent-spread",
            "critical-acl-set":
            [
              {"name":"mirai_infect_v4from"},
              {"name":"mirai_infect_v4to"},
              {"name":"mirai_scan_v4from"},
              {"name":"mirai_scan_v4to"}
            ],
            "action-to-take": "BLOCK_ATTACK"
          },
          {
            "name":"mirai-prevet-cnc",
            "critical-acl-set":
            [
              {"name":"mirai_cnc_v4from"},
              {"name":"mirai_cnc_v4to"}
            ],
            "action-to-take": "BLOCK_N_ATTACK"
          },
          {
            "name":"mirai-prevet-minimal",
            "critical-acl-set":
            [
              {"name":"mirai_cnc_v4from"},
              {"name":"mirai_cnc_v4to"},
              {"name":"mirai_infect_v4from"},
              {"name":"mirai_infect_v4to"}
            ],
            "action-to-take": "BLOCK_ALL"
          },
        ],
        "to-device-attacks": {
          "traffic-lists": {
            "traffic-list": [
              {"name":"mirai_infect_v4to"},
              {"name":"mirai_scan_v4to"}
            ]
          }
```

```
      },
      "from-device-attacks": {
        "traffic-lists": {
          "traffic-list": [
            {"name":"mirai_infect_v4from"},
            {"name":"mirai_scan_v4from"}
          ]
        }
      },
      "to-device-not-attacks": {
        "traffic-lists": {
          "traffic-list": [
            {"name":"mirai_cnc_v4to"}
          ]
        }
      },
      "from-device-not-attacks": {
        "traffic-lists": {
          "traffic-list": [
            {"name":"mirai_cnc_v4from"}
          ]
        }
      },
    }
  ]
},
"ietf-access-control-list:acls": {
  "acl": [
    {
      "name":"mirai_infect_v4to",
      "risk-threshold":11,
      "type": "ipv4-acl-type",
      "aces": {
        "ace": [
          {
            "name":"infect_23_to",
            "risk":10,
            "matches":{
              "ipv4":{
                "ietf-acldns:dst-dnsname":"urn:ietf:params:\
                  mud:gateway",
                "protocol":6
              },
              "tcp":{
                "destination-port":{
                  "operator":"eq",
                  "port":23
                }
              }
```

```
        }
      },
      {
        "name":"infect_2323_to",
        "risk":10,
        "matches":{
          "ipv4":{
            "ietf-acldns:dst-dnsname":"urn:ietf:params:\
              mud:gateway",
            "protocol":6
          },
          "tcp":{
            "destination-port":{
              "operator":"eq",
              "port":2323
            }
          }
        }
      },
      {
        "name":"bin_download_to",
        "risk":1,
        "matches":{
          "ipv4":{
            "ietf-acldns:dst-dnsname":"urn:ietf:params:\
              mud:gateway",
            "protocol":6
          },
          "tcp":{
            "source-port":{
              "operator":"eq",
              "port":80
            },
          }
        }
      }
    ]
  }
},
{
  "name":"mirai_scan_v4to",
  "risk-threshold":11,
  "type": "ipv4-acl-type",
  "aces": {
    "ace": [
      {
        "name":"scan_23_to",
        "risk":10,
        "matches":{
```

```
    "ietf-mud:mud":{
      "local-networks":[ null ]
    },
    "ipv4":{
      "protocol":6
    },
    "tcp":{
      "source-port":{
        "operator":"eq",
        "port":23
      },
    }
  }
},
{
  "name":"scan_2323_to",
  "risk":10,
  "matches":{
    "ietf-mud:mud":{
      "local-networks":[ null ]
    },
    "ipv4":{
      "protocol":6
    },
    "tcp":{
      "source-port":{
        "operator":"eq",
        "port":2323
      },
    }
  }
},
{
  "name":"scan_report_to",
  "risk":1,
  "matches":{
    "ipv4":{
      "ietf-acldns:dst-dnsname":"urn:ietf:params:\
        mud:gateway",
      "protocol":6
    },
    "tcp":{
      "source-port":{
        "operator":"eq",
        "port":48101
      },
    }
  }
}
```

```
    ]
  }
},
{
  "name":"mirai_infect_v4from",
  "risk-threshold":11,
  "type": "ipv4-acl-type",
  "aces": {
    "ace": [
      {
        "name":"infect_23_from",
        "risk":10,
        "matches":{
          "ipv4":{
            "ietf-acldns:dst-dnsname":"urn:ietf:params:\
              mud:gateway",
            "protocol":6
          },
          "tcp":{
            "source-port":{
              "operator":"eq",
              "port":23
            },
          }
        }
      },
      {
        "name":"infect_2323_from",
        "risk":10,
        "matches":{
          "ipv4":{
            "ietf-acldns:dst-dnsname":"urn:ietf:params:\
              mud:gateway",
            "protocol":6
          },
          "tcp":{
            "source-port":{
              "operator":"eq",
              "port":2323
            },
          }
        }
      },
      {
        "name":"bin_download_from",
        "risk":1,
        "matches":{
          "ipv4":{
            "ietf-acldns:dst-dnsname":"urn:ietf:params:\
```

```
              mud:gateway",
            "protocol":6
          },
          "tcp":{
            "destination-port":{
              "operator":"eq",
              "port":80
            }
          }
        }
      }
    ]
  }
},
{
  "name":"mirai_scan_v4from",
  "risk-threshold":11,
  "type": "ipv4-acl-type",
  "aces": {
    "ace": [
      {
        "name":"scan_23_from",
        "risk":10,
        "matches":{
          "ietf-mud:mud":{
            "local-networks":[ null ]
          },
          "ipv4":{
            "protocol":6
          },
          "tcp":{
            "destination-port":{
              "operator":"eq",
              "port":23
            }
          }
        }
      },
      {
        "name":"scan_2323_from",
        "risk":10,
        "matches":{
          "ietf-mud:mud":{
            "local-networks":[ null ]
          },
          "ipv4":{
            "protocol":6
          },
          "tcp":{
```

```
            "destination-port":{
              "operator":"eq",
              "port":2323
            }
          }
        }
      },
      {
        "name":"scan_report_from",
        "risk":1,
        "matches":{
          "ipv4":{
            "ietf-acldns:dst-dnsname":"urn:ietf:params:\
              mud:gateway",
            "protocol":6
          },
          "tcp":{
            "destination-port":{
              "operator":"eq",
              "port":48101
            }
          }
        }
      }
    ]
  }
},
{
  "name":"mirai_cnc_v4to",
  "type": "ipv4-acl-type",
  "aces": {
    "ace": [
      {
        "name":"cnc_socket_to",
        "risk":1,
        "matches":{
          "ipv4":{
            "ietf-acldns:dst-dnsname":"urn:ietf:params:\
              mud:gateway",
            "protocol":6
          },
          "tcp":{
            "source-port":{
              "operator":"eq",
              "port":2030
            },
          }
        }
      }
```

```
            ]
          }
        },
        {
          "name":"mirai_cnc_v4from",
          "type": "ipv4-acl-type",
          "aces": {
            "ace": [
              {
                "name":"cnc_socket_from",
                "risk":1,
                "matches":{
                  "ipv4":{
                    "ietf-acldns:dst-dnsname":"urn:ietf:params:\
                      mud:gateway",
                    "protocol":6
                  },
                  "tcp":{
                    "destination-port":{
                      "operator":"eq",
                      "port":2030
                    }
                  }
                }
              }
            ]
          }
        }
      ]
    }
  }
<CODE ENDS>
```

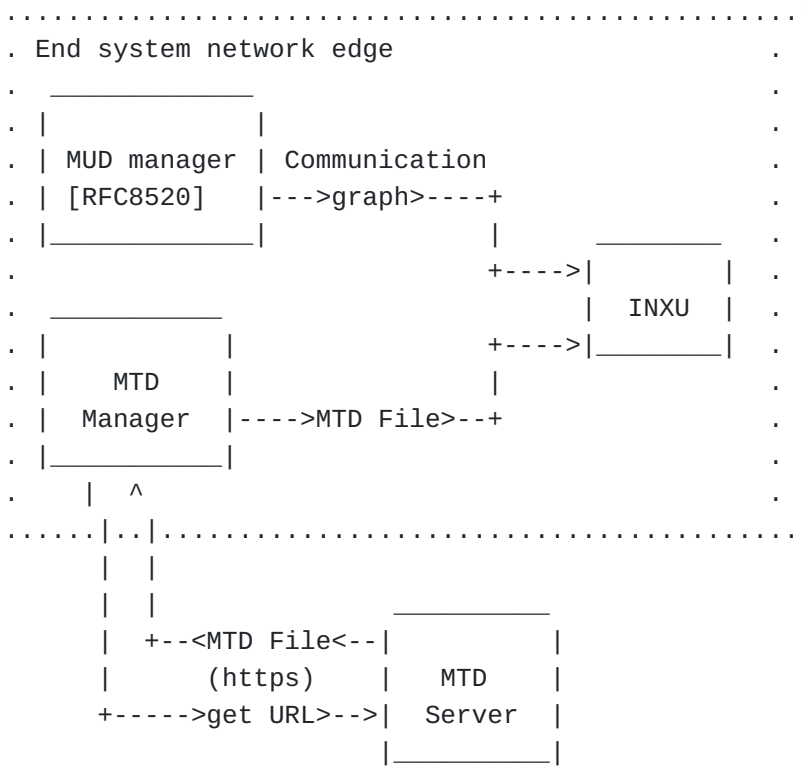
3. The Intra-Network eXposure analyzer Utility

INXU was designed to have as main features: (i) enable quick responses to new vulnerabilities; (ii) allow mitigation of the damages of a new vulnerability, simultaneously in multiple distinct networks; and (iii) enable a decision-making process about security measures on the network edge, avoiding the disclosure of private information to third parties.

To cover these requirements, INXU enables a security expert team - such as a SOC or an ISP Abuse Desk - to describe the traffic of ongoing malicious activities using the MTD data model (more details of the MTD data model are discussed in Section 3). With these descriptions, the security experts team can use the INXU to prevent multiple distinct networks when releasing new MTD Files for every new malicious activity discovered, in a process similar to the antivirus programs vaccines.

3.1. INXU Architecture and Components

The ASCII diagram below shows the architecture of INXU. The proposed architecture contains 4 main components: MTD Server, MTD Manager, INXU Module, and MUD manager [RFC8520].



The MTD Server is responsible for storing and delivering the malicious traffic descriptions made by a security expert. This component was designed to enable trusted third-party specialists to

share knowledge about well-known malicious activities affecting IoT and allow IoT networks to make use of this knowledge to protect themselves. The MTD Server is composed by a HTTPS server that hosts and delivers the MTD File for the clients. The MTD File is a file where the network traffic associated with malicious activities are described to INXU. This component also contains data for version control, authenticity, and validity time. The content of a MTD File is defined by a security expert in a JSON file, following the YANG data model described in the Section 2.

The MTD Manager has the function of managing the MTD File on the system. It is responsible for requesting the file to the MTD Server, verifying authenticity, and requesting a new file when the current file validity expires. The default way to ensure MTD File authenticity is by HTTPS protocol, but the MTD Manager can also use the means described in the Section 3.1 of the [RFC2818]. At the end of the process, the MTD Manager forwards the MTD File to the INXU Module.

The INXU (Intra-Network eXposure analyzer Utility) module is the main component of this proposal. It is responsible for verifying all the network communications trying to identify possible exposures to malicious traffic. To do this, the INXU Module compares the malicious traffic described in a MTD File with the network graph generated by MUD manager. The exposure analysis process is detailed in Section 3.5.

3.2. Workflow

The workflow adopted to INXU may vary, but it will mostly follow the process described below.

1. MTD Manager fetches the MTD File.

2. After confirming MTD File authenticity, the MTD Manager sends it for the INXU Module.

3. The MUD manager sends the network communication graph -- including the devices' MUD URLs -- to the INXU Module.

4. INXU Module identifies potential vulnerability exposures into the network.

5. INXU analyzes the detected vulnerabilities to evaluate if they represent an effective threat. After detecting threats, INXU may act as an Intrusion Prevention System and block the exposures, or serve as input source for other security systems, depending on the implementation.

6. If the MUD manager detects any change in the network topology, or the MTD Manager gets new definitions from MTD Files, the process returns to step 4.

3.3. Acquiring a MTD File

The main method for acquiring a MTD File is by configuring the MTD URL into the MTD Manager. The MTD URL is a Universal Resource Locator (URL) [RFC3986] provided by the Security Experts team designated for protecting the network.

MTD URLs MUST use the "https" scheme [RFC7230].

An alternative manner for acquiring a MTD File is by manually importing and its respective signature file into the MTD Manager. The mechanisms for doing so are not described in this document.

3.4. Processing a MTD URL

Disclaimer: The specification in this section is in our roadmap but still not done. Our initial intention is to use the same specification as [RFC8520] in Section 1.6. To simplify understanding, we copied the original MUD text, pasted it below, and replaced the MUD references with MTD.

MTD Managers that are able to do so SHOULD retrieve MTD URLs and signature files as per [RFC7230], using the GET method [RFC7231]. They MUST validate the certificate using the rules in [RFC2818], Section 3.1.

Requests for MTD URLs SHOULD include an "Accept" header field ([RFC7231], Section 5.3.2) containing "application/mtd+json", an "Accept-Language" header field ([RFC7231], Section 5.3.5), and a "User-Agent" header field ([RFC7231], Section 5.5.3).

MTD Managers SHOULD automatically process 3xx response status codes.

If a MTD Manager is not able to fetch a MTD URL, other means MAY be used to import the MTD File and its associated signature file. So long as the signature of the file can be validated, the file can be used. In such environments, controllers SHOULD warn administrators when cache-validity expiry is approaching so that they may check for new files.

3.5. INXU Vulnerability Analysis Process

The exposure analysis algorithm of the INXU Module uses malicious traffic descriptions from a MTD File to compare with the IoT traffic flows allowed by MUD -- provided by the network communication graph generated by MUD manager -- and tries to detect vulnerabilities on

the network. In this context, INXU identifies one exposure when some graph edge matches with any entry of the MTD File.

Based on the MUD files, each host expected to communicate with the IoT devices, or the IoT devices themselves, are represented by nodes on the network communication graph generated by MUD manager. The host network address represents the nodes, and in the case of IoT devices on the LAN, the MUD URL is associated with the node information. The graph edges represent TCP or UDP sockets, or ICMP communications, where a directed edge represents a communication path.

3.5.1. Exposure Identification

For each IoT node in the MUD-based communication graph, the Exposure Identification process verifies if five information match between edge and ACEs: source and destination host addresses, communication protocol, and source and destination ports -- for transport protocols -- or ICMP message type and code. We only consider an exposure when all five information match.

The ACEs considered here MUST be applicable for any device OR include the IoT device's MUD URL in the specific-devices list.

A match on source or destination host address happens when the addresses are equals OR when the ACE uses the local address abstraction and the node is local.

Protocols match when the specified protocols (TCP, UDP, ICMP, or any) are equal both on ACE and edge OR when the ACE specifies any protocol.

For the ICMP message type and code or for transport's source and destination ports, a match happens when the specified values are equals OR when the ACE specifies any value.

3.5.2. ACL Risk Assessment

Each vulnerability exposure is associated with an ACE and is in the context of an ACL. Therefore, a set of vulnerability exposures of a device becomes a risk when the sum of their ACE risks is bigger than the ACL's risk threshold. There is also a possibility of triggering an alert state when the ACL's risk exceeds the alert threshold.

The risk threshold SHOULD be equals or bigger than the alert threshold.

3.5.3. Threat Analysis

After assessing the risk of each ACL, the next step in the process is the threat analysis. This analysis iterates over the list of the critical ACL sets of a malicious activity.

In this step, the INXU Module verifies if all the ACLs contained in a critical set are classified as risky for a device. If this condition becomes true, the INXU Module SHOULD take the action specified in the set's action-to-take field. If a malicious activity threatens the device with more than one set of critical ACLs, the INXU Module MUST take an action based into a merge of all the threatening sets' action-to-take.

4. Further Considerations and Next Steps

During the development of INXU, we found some important points that could further enhance the proposal in the near future. First of all, although INXU sticks to the Network and Transport layers, many recently reported DDoS attacks exploited the DNS platform to cause damage. This issue requires some treatment in this application layer protocol of the TCP/IP model. As it is a crucial application for the Internet's functioning as we know it today, it is impossible to block traffic over the protocol completely, but we believe that some level of filtering will not negatively impact the devices' usability nor the network's performance.

Another interesting future direction is that although INXU allows identifying, classifying, and mitigating malicious activities on the other hand it does that without any intervention from the user. All the blocking processes do not allow the end-users intervention on the blocks and may lead them to not adopt INXU. An option to overcome this issue is by integrating Software Bill of Materials (SBOM) related information into the MTD data model and in the Threat Analysis process, and allowing end-users feedback on blocking decisions. This may reduce INXU's impact on usability with low security loss and consequently improve its adoption.

Also in this sense, we could use the MTD as a standard data model for attacks signatures involving IoT. It is a useful way to share how attacks can alter the network traffic to be used in controlled experiments and simulations. Also it can be seen also as a systematic way to share information on attacks -- in this sense network administrators, scientists and security analysts could have the same view over a given event in the network.

Finally, also coming from the previous statement, INXU's output could be used as an input filter for IPS/IDS systems in order to prevent attacks and any other malicious event in the network. Since

by using the MTD we could classify the traffic into appropriate or not. Furthermore INXU -- specially the MTD -- could be paired with an AI engine to learn about new network patterns and classify them as an attack or some new device in the network -- the system could write some new MTDs as it learns from the network.

5. Security Considerations

Since INXU uses MUD as a data source, the problems presented at the Security Considerations session of the [RFC8520] are still valid for this proposal, and some new ones arise.

The first new risk is the possibility of INXU causing Denial of Service on their protected IoT devices depending on how the malicious activities are described in the MTD File. To prevent this issue, while describing a malicious activity, the Security Specialist SHOULD be as specific as possible by describing, for example, the specific devices that can be affected by the attack or malware and being assertive while defining ACE risks and ACL risk thresholds.

As with MUD, the MTD Manager may receive a fake MTD File from a rogue MTD Server with a certificate issued by an accredited certification authority (CA). In this case, the same MUD mitigations apply: First, if the signer changes, this may be flagged as an exception by the MTD manager. Second, if the MTD file also changes, the MTD manager SHOULD seek administrator approval (it should do this in any case). In all circumstances, the MUD manager MUST maintain a cache of trusted CAs for this purpose. When such a rogue is discovered, it SHOULD be removed.

Finally, INXU is not effective against attacks that are occurring prior to a new MTD file arriving or ongoing at the moment of an update. The classification of the attack is not accurate since it does not know the rules. A countermeasure is to use an anomaly detection system to identify such attacks. INXU is not responsible for that part.

Further security considerations might arise during this document's evolution.

6. IANA Considerations

This memo includes no request to IANA.

7. References

7.1. Normative References

[RFC2119]

Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/info/rfc2119>.

[RFC3986] Berners-Lee, T., Fielding, R. T., and L. M. Masinter, "Uniform Resource Identifier (URI): Generic Syntax", January 2005, <https://rfc-editor.org/rfc/rfc3986.txt>.

[RFC7230] Fielding, R. T. and J. Reschke, "Hypertext Transfer Protocol (HTTP/1.1): Message Syntax and Routing", June 2014, <https://rfc-editor.org/rfc/rfc7230.txt>.

[RFC7231] Fielding, R. T. and J. Reschke, "Hypertext Transfer Protocol (HTTP/1.1): Semantics and Content", June 2014, <https://rfc-editor.org/rfc/rfc7231.txt>.

[RFC7950] Bjorklund, M., "The YANG 1.1 Data Modeling Language", August 2016, <https://rfc-editor.org/rfc/rfc7950.txt>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", May 2017, <https://rfc-editor.org/rfc/rfc8174.txt>.

[RFC8340] Björklund, M. and L. Berger, "YANG Tree Diagrams", March 2018, <https://rfc-editor.org/rfc/rfc8340.txt>.

[RFC8519] Jethanandani, M., Agarwal, S., Huang, L., and D. Blair, "YANG Data Model for Network Access Control Lists (ACLs)", March 2019, <https://rfc-editor.org/rfc/rfc8519.txt>.

[RFC8520] Lear, E., Droms, R., and D. Romascanu, "Manufacturer Usage Description Specification", March 2019, <https://rfc-editor.org/rfc/rfc8520.txt>.

7.2. Informative References

[Mozzaquatro2015] Mozzaquatro, B. A., Jardim-Goncalves, R., and C. Agostinho, "Towards a reference ontology for security in the Internet of Things", 2015, <https://doi.org/10.1109/IWMN.2015.7322984>.

[RFC2818] Rescorla, E., "HTTP Over TLS", May 2000, <https://rfc-editor.org/rfc/rfc2818.txt>.

Authors' Addresses

Sávyo Vinícius de Morais
IFRN

Natal-
Brazil

Email: savyovm@gmail.com

Claudio Miceli de Farias
UFRJ
Rio de Janeiro-
Brazil

Email: cmicelifarias@gmail.com