Poster: All Your Data Are Available to Us: A Need for Network Segmentation with IoT Devices

Karl Olson
University of Colorado - Boulder
karl.olson@colorado.edu

Nolen Scaife
University of Colorado - Boulder
scaife@colorado.edu

***Abstract*—Universal Plug And Play (UPnP) is a nearly ubiquitous device connectivity protocol used by many home routers and smart devices to streamline user connectivity. With the increase in the prevalence of Internet of Things (IoT) and smart devices within the home, security risks and user exposure will increase due to each connected device having full authenticated access to the local network. In this poster, we show that UPnP can be utilized to automatically segregate network traffic among devices and limit the exposure of high value target devices (PC, User Phone) from potential vulnerabilities in everyday IoT/Smart systems, thereby limiting their potential exposure to compromise and data loss. We develop and demonstrate operation of UPnP Segregation, a backwards compatible protocol framework to segregate device traffic through the use of system identification and controlled traffic forwarding using firewall segmentation. Our analysis shows that with this framework, critical device traffic can be segregated from IoT/Smart devices to limit the exposure to potential vulnerabilities thus providing the first robust network to use UPnP Segregation framework.**

***Index Terms*—IOT, UPnP, Threat Model, Network Segmentation**

I. Introduction

With the pervasiveness of UPnP in the consumer market, many extensions unrelated to security of UPnP have been proposed to address new capabilities such as application interoperability and cloud services. Unfortunately, these designs revert to original protocol specifications without concern for device or communication safety [2]. With the influx of IoT and the repeated failures of UPnP, a number of competing architectures have emerged to fulfil the gaps where UPnP left off. The first of these was UPnP+ which was introduced in 2014 as an evolution of security implementations within UPnP while simultaneously shepherding the start of cloud based connectivity and features. Other software frameworks like AllJoyn, Thread, and the Open Connectivity Foundation (OCF) have all introduced standards based approaches for developing IoT connectivity solutions while many more developers are presenting their own service specific platforms for IoT [1]. When combined with the many IoT technology protocols available for use in communications, the complexity and the security footprint has increased significantly within a standard user's home environment.

While many of the original critical flaws of UPnP have been addressed in these new standards, and basic security implementations are being applied, the threat model used to consider security solutions has changed significantly. Users should no longer be considered trusted within a domain as challenges in domestic disputes, shared living spaces and an influx of connected devices within a home and connectivity via external locations all change the need and requirements for security considerations. As such, with the value in gathering user data by companies such as Google, Amazon and Apple, a significant question and security consideration remains: Why do these devices need complete access to the local network and by extrapolation, all the devices within it? Utilization of UPnP provides devices a method for quickly connecting to the home domain and configuring external access, but never asks the question does the device need the ability to be on the same LAN as every other device in the house? These questions underscore the need to further secure the underlying protocol design to meet the challenges of today's pervasive and blended environments. To date, the authors know of no efforts to utilize UPnP or NAT-PMP to segregate traffic domains based on device need in a local area network. In this work, we explore methods to use the UPnP protocol to automate segregation of network traffic utilizing a custom SCDP/SOAP action as shown in Figure 1. We chose UPnP for our implementation due to its wide availability and established design as a hole-punching protocol to implement our modifications.

II. Implementation

To implement our automatic segregation of IoT devices we utilized an open-source PFSense firewall with miniUPnP daemon to enact network segregation SOAP calls from our IOT device. Procedures from connection to operation follow in Figure 2. 1) IoT device manually added to network where it automatically scans for a UPnP control point to access. 2) Control point responds with location information and SCPD xml list of services available for action. 3)IoT device requests need for segregated network access through custom SOAP request. 4) miniUPnP server reads SOAP request, identifies need to create VLAN on firewall and segregates traffic. 5) Device operates on segregated network. In case of vulnerability or compromise, device would not provide authenticated access to other network systems not part of local VLAN by default. 6) Recommendation for IoT device configuration through a direct WiFi connection or authenticated connection to cloud control point. This maintains device segmentation during communi-

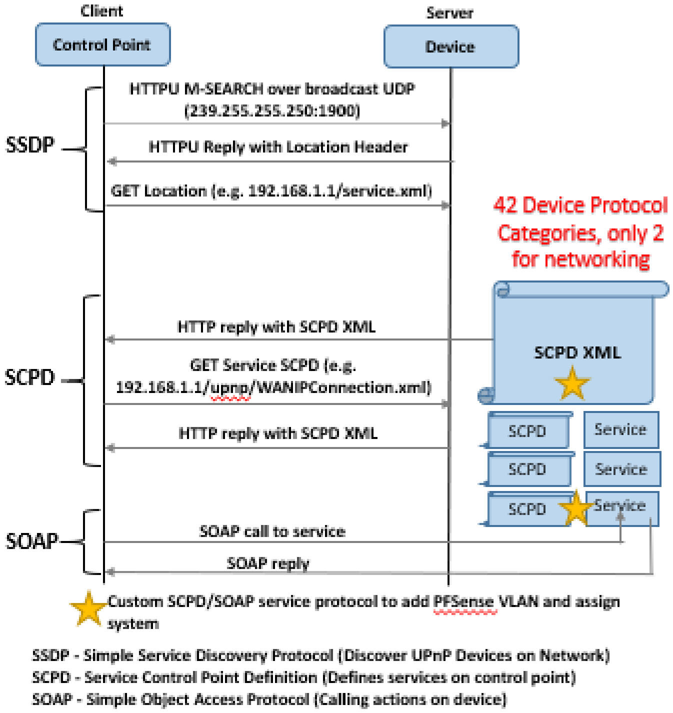

Client
Control Point
Server
Device
SSDP
HTTPU M-SEARCH over broadcast UDP (239.255.255.250:1900)
HTTPU Reply with Location Header
GET Location (e.g. 192.168.1.1/service.xml)
42 Device Protocol Categories, only 2 for networking
SCPD
HTTP reply with SCPD XML
SCPD XML
GET Service SCPD (e.g. 192.168.1.1/upnp/WANIPConnection.xml)
HTTP reply with SCPD XML
SCPD
Service
SCPD
Service
SCPD
Service
SOAP
SOAP call to service
SOAP reply
Custom SCPD/SOAP service protocol to add PFSense VLAN and assign system
SSDP - Simple Service Discovery Protocol (Discover UPnP Devices on Network)
SCPD - Service Control Point Definition (Defines services on control point)
SOAP - Simple Object Access Protocol (Calling actions on device)

Fig. 1. UPnP Protocol Operation with Modification.

cation process. Currently every IoT device would maintain individual segmentation with work being conducted to further control this process.

III. Results

During current standard connectivity approaches used by IoT devices using UPnP, Thread, OCF or similar protocols, devices are connected to the local network without consideration of segmentation. This results in leakage of information, such as connected device information, utilization metrics, or potentially unencrypted network data. Together this data and metadata can be utilized to identify users based on behavior, other potentially vulnerable devices on the local network for pivot, or direct data gathering from a compromised system. (references?). In our test setup, we use a Netgear Arlo Pro security camera and Google Home Mini, two devices that utilized the UPnP protocol to ease network setup for a home user. We monitored the information these two devices shared on the network to establish a baseline of data that could be gathered from such a scenario. Information retrieved include device information, utilization metrics and unencrypted communications, all data that could be used to establish user behavior or establish potential pivot points for further attack. By segregating these devices onto a separate network segment we are easily able to limit this data leakage. While the concept of network segregation is not unique or complicated to perform, many home users do not have the basic understanding or ability to perform such a function. Therefore, we demonstrate a critical shortcoming in IoT systems to build this segmentation effortlessly and improve user home security through automated processes.

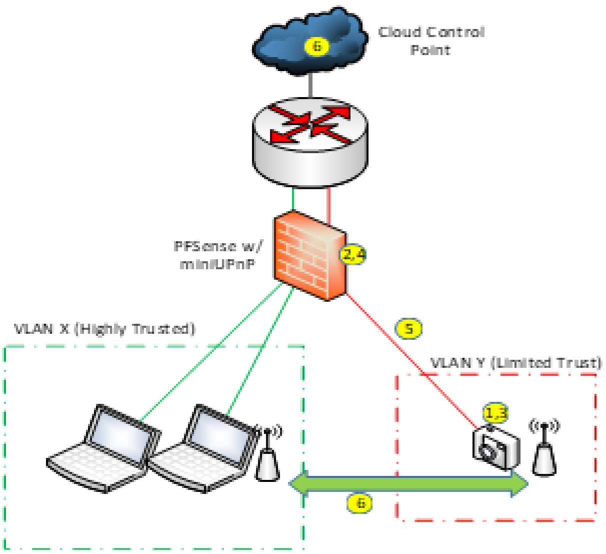


Fig. 2. IoT Device Segmentation Process.

IV. Conclusion and Discussion

As more IoT devices begin to find their way into homes, challenges with consumers patching and securing these devices will result in increasing access to sensitive data within a shared environment. Manufacturer's and protocol developers need to consider a new threat model that considers environmental need for shared network access and to proactively find ways to reduce potential compromise. While we demonstrate a simple capability to ease consumer requirements by automating network segmentation by modifying UPnP, true change needs to occur at a protocol standards level. In future work we will publish our protocol modification along with support for other competing protocols such as NAT-PMP devices.

References

[1] Wemo consumer website. https://www.wemo.com/. Accessed: 2019-12-01.

[2] Gopinath Ganapathy Kalaiselvi Arunachalam. Extending UPnP for application interoperability in a home network. *International Journal of Electrical and Computer Engineering (IJECE)*, pages 2085–2093, 2017.

College of Engineering and Applied Science
University of Colorado **Boulder**

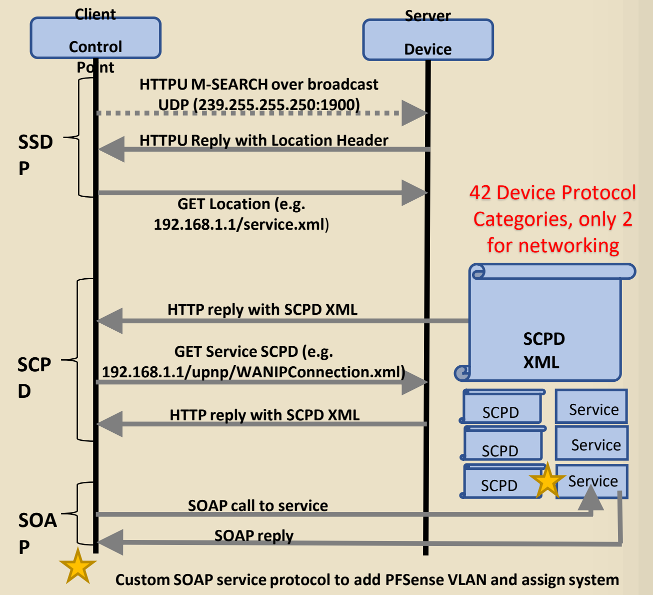
All Your Data Are Available To Us: A Need For Network Segmentation With IoT Devices

Karl Olson, Nolen Scaife

Background

- UPnP and NAT-PMP are nearly ubiquitous device connectivity protocols used by many home routers and smart devices to streamline user connectivity and setup.
- Many flaws and shortcomings in protocol design and incorrect vendor implementations have resulted in high profile network and device vulnerabilities utilizing the protocol.
- With increase in prevalence of Internet of Things and smart devices within the home, security risks and user exposure will increase due to each connected device having full authenticated access to the local network.
- While solutions have been proposed that focus on improving the security of UPnP, no effort to date has questioned the need for devices to have full authenticated access to the network (Threat Model mismatch).
- Many IoT/Smart Home devices rarely are patched and may go their entire life without fixing a vulnerability
- Competing proposals to UPnP such as OCF, AllJoyn, UPnP+ and Thread continue to fail to address concerns in proper device segregation and network access.

UPnP Protocol Operation

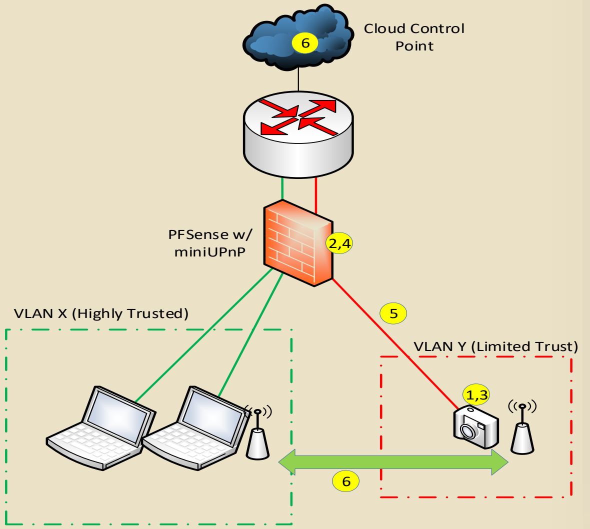

Client
Control Point
Server
Device
HTTPU M-SEARCH over broadcast UDP (239.255.255.250:1900)
HTTPU Reply with Location Header
SSDP
GET Location (e.g. 192.168.1.1/service.xml)
42 Device Protocol Categories, only 2 for networking
HTTP reply with SCPD XML
SCPD XML
GET Service SCPD (e.g. 192.168.1.1/upnp/WANIPConnection.xml)
SCPD
HTTP reply with SCPD XML
SCPD
Service
SCPD
Service
SCPD
Service
SOAP call to service
SOAP
SOAP reply
Custom SOAP service protocol to add PFSense VLAN and assign system

SSDP - Simple Service Discovery Protocol (Discover UPnP Devices on Network)
SCPD - Service Control Point Definition (Defines services on control point)
SOAP - Simple Object Access Protocol (Calling actions on device)

Research Questions

1. Can we improve security of a local network comprised of authenticated privileged systems and IoT/Smart Home devices without sacrificing the ease of consumer setup?
2. What levels of access do home devices require in order to operate under the concept of least privilege?
3. Can we utilize an automatic configuration protocol, such as UPnP, to segregate network access between privileged and unprivileged systems?
4. How can we identify a device appropriately to ensure it is placed on the right network segment for operation without involving user input?
5. How has the threat model changed since inception of UPnP and are we addressing current and emerging concerns correctly?

Experimental Setup



1 New device connects to network and initiates SSDP discovery.

2 miniUPnP server responds with location and SCPD xml list of services. Device obtains list of services on firewall.

3 Device Identifies itself as IoT Device and need for segregated network access through SOAP request

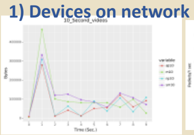
4 miniUPnP server utilizes SOAP instruction to create VLAN on Firewall and segregate traffic

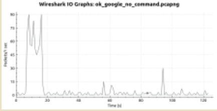
5 Devices operates on segregated network. In case of compromise or vulnerability, device would not provide authenticated access to other home systems.

6 Recommendation for device configuration through *direct* wifi connection or through authenticated connection to cloud control point. Maintains device segregation.

Results / Discussion

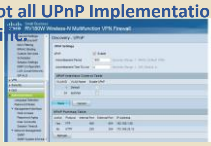
1) Devices on network leak information, can be identified by traffic.



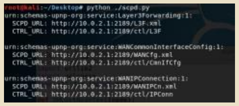


A Netgear Arlo Pro camera and Google Home Mini open the windows to your personal life as you interact with them as seen in the spikes in the picture. The data was collected by Wireshark on the same LAN as these devices. In Google Home Mini, TLSv1 data is collected even when you are not interacting with it.

2) Not all UPnP Implementations are the same! Device/Vendor specific.

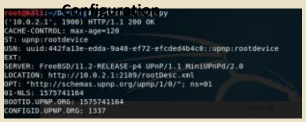


Some allow UPnP setup by VLAN – User intensive Configuration



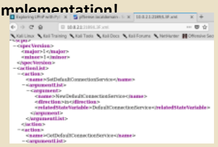
```
root@kali:~/Desktop# python ./scpd.py
urn:schemas-upnp-org:service:Layer3Forwarding:1:
  SCPD_URL: http://10.0.2.1:2189/L3F.xml
  CTRL_URL: http://10.0.2.1:2189/ctl/L3F

urn:schemas-upnp-org:service:WANCommonInterfaceConfig:1:
  SCPD_URL: http://10.0.2.1:2189/WANCfg.xml
  CTRL_URL: http://10.0.2.1:2189/ctl/CmnIfCfg

urn:schemas-upnp-org:service:WANIPConnection:1:
  SCPD_URL: http://10.0.2.1:2189/WANIPCn.xml
  CTRL_URL: http://10.0.2.1:2189/ctl/IPConn
```

No DeviceSecurity:1 Implementation!



```
root@kali:~/D
('10.0.2.1', 1900) HTTP/1.1 200 OK
CACHE-CONTROL: max-age=120
ST: upnp:rootdevice
USN: uuid:442fa13e-edda-9a48-ef72-efcded4b4c8::upnp:rootdevice
EXT:
SERVER: FreeBSD/11.2-RELEASE-p4 UPnP/1.1 MiniUPnPd/2.0
LOCATION: http://10.0.2.1:2189/rootDesc.xml
OPT: "http://schemas.upnp.org/upnp/1/0/"; ns=01
01-NLS: 1575741164
BOOTID.UPNP.ORG: 1575741164
CONFIGID.UPNP.ORG: 1337
```

With our program, we are able to scan network for UPnP devices, pull their service listing and call SOAP actions.



```
- <specVersion>
    <major>1</major>
    <minor>1</minor>
  </specVersion>
- <actionList>
  - <action>
      <name>SetDefaultConnectionService</name>
    - <argumentList>
      - <argument>
          <name>NewDefaultConnectionService</name>
          <direction>in</direction>
          <relatedStateVariable>DefaultConnectionService</relatedStateVariable>
        </argument>
      </argumentList>
    </action>
  - <action>
      <name>GetDefaultConnectionService</name>
    - <argumentList>
```

However, SOAP actions are tied to vendor implementation by UPnP standard. No opportunity to implement VLAN without modifying vendor source code (Proprietary).

3) Automating network segregation will require each vendor to implement independently.

-likely an unattainable proposition (thousands of patches, users, etc)
-demonstrates need to update threat model and protocol standards

4) Despite device security standards avail, no implementation of UPnP+.

-Vendors generally apply bare minimum
-encrypted communication not applied
-Security features avail for nearly 10 years with no usage

Recommendations / Future Work

- Push protocol framework developers to consider updated threat model that involves device segregation measures.
- Create miniUPnP server module library of SOAP instruction set to automate VLAN creation and segregation. Target open source: openWRT, pfSense.
- Assess competing standards for completeness of avail solution sets, rank based on security features and application (Paper)
- Assess threat models to environment, and protocol compliance to address, recommend new threat models for adoption into security standards (Paper)

Acknowledgements:
Project Funded Under Grant#: N/A

For Presentation At:NDSS 2020, San Diego, CA
Date: 23 February 2020