Let’s talk about SOAP, baby.
Let’s talk about UPnP.

Ricky “HeadlessZeke” Lawshae – DEFCON 23
Who am I?

• Security Researcher for HP TippingPoint’s DVLabs team
• At Rapid7 before that, and BreakingPoint before that
• Speaker at Defcon, Recon, Insomni’hack, and Ruxcon
• Voider of warranties
• Reader of comic books
• Drinker of beers

• TRIVIA: I once got a job at a police department while I had 4 active warrants out for my arrest.
What are we talking about?

• The Internet of Things™ (ugh...)
  • It’s here, whether you like it or not
  • “Just put a network interface on it. We’ll worry about why later.”

• Smart devices aren’t very smart
  • Need simple way to talk to each other
  • Ease-of-use: Get the tech out of the way of UX

• Often accomplished with SOAP/UPnP services
  • Super talkative
  • Happily tell you all their capabilities in a well-structured format
  • Also, don’t bother themselves with pesky issues like security
What are we talking about?

• UPnP
  • Universal Plug and Play

• SSDP
  • Simple Service Discovery Protocol

• SCPD
  • Service Control Protocol Definition

• SOAP
  • Simple Object Access Protocol
Let’s talk about all the good things...
UPnP

• 1900/UDP
  • HTTP over UDP allowing devices to discover each other
  • Multicast 239.255.255.250

• UPnP Stack\(^1\)
  • Discovery
    • Advertising and Searching
  • Description
    • An XML file describing the device
  • Control
    • Call an action or query for a value
  • Eventing
    • Used for announcing state changes
  • Presentation
    • UI...web page or management portal I guess?

\(^1\)http://www.upnp.org/specs/arch/UPnP-arch-DeviceArchitecture-v1.0-20080424.pdf
UPnP – Discovery

Advertising

NOTIFY * HTTP/1.1
Host: 230.255.255.250:1900
Cache-Control: max-age=1
Location: http://x.x.x.x:12345/desc.xml
Server: OS 1.0 UPnP/1.0 Realtek/V1.3
NT: upnp:rootdevice
NTS: ssdp:byebye

Searching

M-SEARCH * HTTP/1.1
HOST: 230.255.255.250:1900
MAN: "ssdp:discover"
MX: 5
ST: ssdp:all

Responding

HTTP/1.1 200 OK
CACHE-CONTROL: max-age = 1800
EXT:
LOCATION: http://x.x.x.x:12345/desc.xml
SERVER: Linux/9.0 UPnP/1.0 PROTOTYPE/1.0
ST: uuid:24ef1ceef6ba-589a-39ee-14f69df0eb5
USN: uuid:24ef1ceef6ba-589a-39ee-14f69df0eb5
CONTENT-LENGTH: 0

All you need to know about discovery. Also, this is the really noisy part.
UPnP – Discovery

All you need to know about discovery. Also, this is the really noisy part.
UPnP – Description

• XML file usually hosted on a high number TCP port

• Version info
  • upnp.org spec
  • Usually just 1.0

• Device definitions
  • Device type
  • Make/model/UUID
  • Service list
    • Service type
    • SCPD URL
    • Control URL
    • Event URL
UPnP – Description

<specVersion>
  <major>1</major>
  <minor>0</minor>
</specVersion>
<URLBase>http://10.0.0.1:5000/</URLBase>
<device>
  <pnpx:X_hardwareId>VEN_01f2&amp;…&amp;REV_01</pnpx:X_hardwareId>
  <pnpx:X_deviceCategory>NetworkInfrastructure.Router</pnpx:X_deviceCategory>
  <df:X_deviceCategory>Network.Router.Wireless</df:X_deviceCategory>
  <pnpx:X_compatibleId>urn:schemas-upnp-org:device:InternetGatewayDevice:1</pnpx:X_compatibleId>
  <deviceType>urn:schemas-upnp-org:device:InternetGatewayDevice:1</deviceType>
  <friendlyName>WNDR3400v2 (Gateway)</friendlyName>
  <manufacturer>NETGEAR, Inc.</manufacturer>
  <manufacturerURL>http://www.NETGEAR.com</manufacturerURL>
  <modelDescription>NETGEAR WNDR3400v2 N600 Wireless Router</modelDescription>
  <modelNumber>WNDR3400v2</modelNumber>
  <modelName>WNDR3400v2</modelName>
  <modelURL>http://www.netgear.com</modelURL>
  <UDN>uuid:bc567461-ee40-a9c2-39d3-5338c402cc8d</UDN>
  <iconList>…</iconList>
  <serviceList>
    <service>
      <serviceType>urn:schemas-upnp-org:service:Layer3Forwarding:1</serviceType>
      <serviceId>urn:upnp-org:serviceId:L3Forwarding1</serviceId>
      <SCPDURL>/Public_UPNP_Layer3F.xml</SCPDURL>
      <controlURL>/Public_UPNP_C1</controlURL>
      <eventSubURL>/Public_UPNP_Event_1</eventSubURL>
    </service>
  </serviceList>
</device>
UPnP – Description

<specVersion>
  <major>1</major>
  <minor>0</minor>
</specVersion>
<URLBase>http://10.0.0.1:5000/</URLBase>
<device>
  <pnpx:X_hardwareId>VEN_01f2...&REV_01</pnpx:X_hardwareId>
  <pnpx:X_deviceCategory>NetworkInfrastructure.Router</pnpx:X_deviceCategory>
  <df:X_deviceCategory>Network.Router.Wireless</df:X_deviceCategory>
  <pnpx:X_compatibleId>urn:schemas-upnp-org:device:InternetGatewayDevice:1</pnpx:X_compatibleId>
  <deviceType>urn:schemas-upnp-org:device:InternetGatewayDevice:1</deviceType>
  <friendlyName>WNDR3400v2 (Gateway)</friendlyName>
  <manufacturer>NETGEAR, Inc.</manufacturer>
  <manufacturerURL>http://www.NETGEAR.com</manufacturerURL>
  <modelDescription>NETGEAR WNDR3400v2 N600 Wireless Router</modelDescription>
  <modelNumber>WNDR3400v2</modelNumber>
  <modelName>WNDR3400v2</modelName>
  <modelURL>http://www.netgear.com</modelURL>
  <UDN>uuid:bc567461-ee40-a9c2-39d3-5338c402cc8d</UDN>
  <iconList>...</iconList>
  <serviceList>
    <service>
      <serviceType>urn:schemas-upnp-org:service:Layer3Forwarding:1</serviceType>
      <serviceId>urn:upnp-org:serviceId:L3Forwarding1</serviceId>
      <SCPDURL>/Public_UPNP_Layer3F.xml</SCPDURL>
      <controlURL>/Public_UPNP_C1</controlURL>
      <eventSubURL>/Public_UPNP_Event_1</eventSubURL>
    </service>
  </serviceList>
</device>
UPnP – SCPD

• XML file defining the service actions and arguments

• Version info
  • Same deal as description

• Action list
  • Action name
  • Arguments
    • Argument name
    • Direction (input/output)
    • Variable name

• Variable list
  • Variable name
  • Data type
<actionList>
  <action>
    <name>SetDefaultConnectionService</name>
    <argumentList>
      <argument>
        <name>NewDefaultConnectionService</name>
        <direction>in</direction>
        <relatedStateVariable>DefaultConnectionService</relatedStateVariable>
      </argument>
    </argumentList>
  </action>
  <action>
    <name>GetDefaultConnectionService</name>
    <argumentList>
      <argument>
        <name>NewDefaultConnectionService</name>
        <direction>out</direction>
        <relatedStateVariable>DefaultConnectionService</relatedStateVariable>
      </argument>
    </argumentList>
  </action>
</actionList>

<serviceStateTable>
  <stateVariable sendEvents="yes">
    <name>DefaultConnectionService</name>
    <dataType>string</dataType>
  </stateVariable>
</serviceStateTable>
<actionList>
  <action>
    <name>SetDefaultConnectionService</name>
    <argumentList>
      <argument>
        <name>NewDefaultConnectionService</name>
        <direction>in</direction>
        <relatedStateVariable>DefaultConnectionService</relatedStateVariable>
      </argument>
    </argumentList>
  </action>
  <action>
    <name>GetDefaultConnectionService</name>
    <argumentList>
      <argument>
        <name>NewDefaultConnectionService</name>
        <direction>out</direction>
        <relatedStateVariable>DefaultConnectionService</relatedStateVariable>
      </argument>
    </argumentList>
  </action>
</actionList>

<serviceStateTable>
  <stateVariable sendEvents="yes">
    <name>DefaultConnectionService</name>
    <dataType>string</dataType>
  </stateVariable>
</serviceStateTable>
UPnP – Control

• This is where SOAP comes in (finally!)
• Mostly just frontends for an RPC service or CGI script
• SOAP envelopes
  • XML-formatted API calls
  • Service type from description XML
  • Action name and arguments from SCPD XML
• POST envelope to control URL
POST /Public_UPNP_C1 HTTP/1.1
Content-Type: text/xml; charset=utf-8
SOAPAction: "urn:schemas-upnp-org:service:Layer3Forwarding:1#SetDefaultConnectionService"
Content-Length: 568
Host: x.x.x.x:12345

<?xml version="1.0" encoding="utf-8" ?>
<env:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    xmlns:env="http://schemas.xmlsoap.org/soap/envelope/">
    <env:Body>
        <n1:SetDefaultConnectionService xmlns:n1="urn:schemas-upnp-org:service:Layer3Forwarding:1">"!
            <NewDefaultConnectionService xsi:type="xsd:string">blah</NewDefaultConnectionService>
        </n1:SetDefaultConnectionService>
    </env:Body>
</env:Envelope>
TL;DR

M-SEARCH -> HTTP 200 OK
  • Desc URL

GET Desc URL

HTTP 200 OK
  • Action Name
  • Arguments

GET SCPD URL

HTTP 200 OK
  • Service Type
  • SCPD URL
  • Control URL

POST Control URL
  • Service Type
  • Action Name
  • Arguments

HTTP 200 OK
But what can you do with it?
But what can you do with it?

• Control AV equipment
• Home automation
• Network administration
• Physical security systems (ok, easy there buddy)
• Industrial monitoring and control (uh...what?)
• And this is just the official specs
Neat, so...

- All our devices can talk to each other!
- Brave new worlds of remote control and automation!
- Have your toaster turn on the lights, set the TV to the news channel, and send you a text message when breakfast is ready!
- The future is now!
- Nothing could possibly go wrong!
And the bad things...
What about security?

• Embedded devices
  • Limited memory and processing power
  • Board dev and software dev are often completely different companies
  • Copy-and-paste development
  • Keep costs low
  • Not exactly concerned/knowledgeable

• Deployment
  • Millions of internet-facing UPnP-enabled devices
  • Too many vendors to count
  • Frontend is standardized, backend varies even within same vendor
  • Difficult to patch/update firmware
  • Just because you can, doesn’t mean you should
What about security?

- XML parsing is hard
  - Needs lots of system resources
  - Free-form, user-supplied data
  - In 2013, 2.5% of CVE’s were XML-related\(^2\)
  - Of those, almost 36% had CVSS severity of 7 or above
  - As the use-case for XML grows, so do the classes of vulns
    - Recursion bugs, XXE, command injection, etc...

\(^2\) [http://cve.mitre.org/cgi-bin/cvekey.cgi?keyword=xml](http://cve.mitre.org/cgi-bin/cvekey.cgi?keyword=xml)
Attack surface

• UPnP service
  • HTTP header parsing
  • SSDP parsing
  • OS command injection
  • Information disclosure

• SOAP service
  • HTTP header parsing
  • XML parsing
  • Injection vulns
    • OS command
    • SQL injection
    • SOAP injection
  • Information disclosure
  • Ridiculous levels of unauthenticated device control
Attack surface – UPnP

- CVE-2012-5958
  - Disclosed a couple years ago by HD Moore (one of many)
  - https://community.rapid7.com/docs/DOC-2150
  - Calls strncpy to copy a string from the ST header into TempBuf[COMMAND_LEN]
  - Size argument for strncpy is based on number of characters between colons
Attack surface – UPnP

- CVE-2012-5958
  - Disclosed a couple years ago by HD Moore (one of many)
  - https://community.rapid7.com/docs/DOC-2150
  - Calls strncpy to copy a string from the ST header into TempBuf[COMMAND_LEN]
  - Size argument for strncpy is based on number of characters between colons

```
M-SEARCH * HTTP/1.1
Host:239.255.255.250:1900
ST:uuid:schemas:device:[string longer than
COMMAND_LEN]:blah
Man:"ssdp:discover"
MX:3
```
Attack surface – UPnP

• D-Link DIR-815 UPnP Command Injection
  • Disclosed Feb 2013 by Zach Cutlip
  • Contents of ST header get passed as arguments to M-SEARCH.sh
  • No validation or sanitization
Attack surface – UPnP

- D-Link DIR-815 UPnP Command Injection
  - Disclosed Feb 2013 by Zach Cutlip
  - Contents of ST header get passed as arguments to M-SEARCH.sh
  - No validation or sanitization

M-SEARCH * HTTP/1.1
Host:239.255.255.250:1900
ST:uuid:`[shell command]`
Man:"ssdp:discover"
MX:3
Attack surface – SOAP

- AirTies RT Series SOAPAction Name Buffer Overflow
  - Disclosed earlier this year by Onur Alanbel
  - https://www.exploit-db.com/exploits/36839/
  - ExecuteSoapAction function allocates statically-sized buffer
  - Calls memcpy to copy value of SOAPAction header into it with no bounds checking
Attack surface – SOAP

- AirTies RT Series SOAPAction Name Buffer Overflow
  - Disclosed earlier this year by Onur Alanbel
  - https://www.exploit-db.com/exploits/36839/
  - ExecuteSoapAction function allocates statically-sized buffer
  - Calls memcpy to copy value of SOAPAction header into it with no bounds checking

```
POST / HTTP/1.1
Content-Type: text/xml; charset=utf-8
SOAPAction: "n:schemas-upnp-org:service:WANIPConnection:1# [more than 2048 bytes]"
Content-Length: [length of req]
Host: x.x.x.x:5555
```
Attack surface – SOAP

• Broadcom SetConnectionType Format String Vulnerability
  • Disclosed a couple years ago by Leon Juranic and Vedran Kajic
  • SetConnectionType action feeds value of NewConnectionType argument to `snprintf`
  • No sanitization of user-controlled value
Attack surface – SOAP

• Broadcom SetConnectionType Format String Vulnerability
  • Disclosed a couple years ago by Leon Juranic and Vedran Kajic
  • SetConnectionType action feeds value of NewConnectionType argument to snprintf
  • No sanitization of user-controlled value

```xml
<SOAP-ENV:Body>
  <m:SetConnectionType
    xmlns:m="urn:schemas-upnp-org:service:WANIPConnection:1"
    as=""
  >
    <NewConnectionType>[format string]</NewConnectionType>
  </m:SetConnectionType>
</SOAP-ENV:Body>
```
Attack surface – SOAP

• CVE-2014-3242
  • Disclosed last year by pnig0s
  • http://www.pnigos.com/?p=260
  • SOAPpy allows declaration of user-defined XML External Entities in SOAP request
  • No sanitization of user-controlled value
Attack surface – SOAP

• CVE-2014-3242
  • Disclosed last year by pnig0s
  • http://www.pnigos.com/?p=260
  • SOAPpy allows declaration of user-defined XML External Entities in SOAP request
  • No sanitization of user-controlled value

```xml
<!DOCTYPE v1 [<!ENTITY xxe SYSTEM "file:///etc/passwd">]>
<SOAP-ENV:Envelope ... >
  <SOAP-ENV:Body>
    <echo SOAP-ENC:root="1">
      <v1 xsi:type="xsd:string">&xxe;</v1>
    </echo>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```
Attack surface – SOAP

• CVE-2014-2928
  • Disclosed last year by Brandon Perry (PBerry Crunch!)
  • http://seclists.org/fulldisclosure/2014/May/32
  • F5 iControl API set_hostname action passes value of hostname argument to shell
  • Once again, no sanitization of user-controlled value
Attack surface – SOAP

- CVE-2014-2928
  - Disclosed last year by Brandon Perry (PBerry Crunch!)
  - http://seclists.org/fulldisclosure/2014/May/32
  - F5 iControl API set_hostname action passes value of hostname argument to shell
  - Once again, no sanitization of user-controlled value

```xml
<SOAP-ENV:Body>
  <n1:set_hostname xmlns:n1="urn:iControl:System/Inet">
    <hostname>`[shell command]`.whatever.com</hostname>
  </n1:set_hostname>
</SOAP-ENV:Body>
```
Attack surface – SOAP

• Netgear R6200 SetFirmware fun
  • Spread across a series of blog posts starting in April 2015 (Zach Cutlip again)
  • http://shadow-file.blogspot.com/2015/04/abandoned-part-01.html
  • Dead/non-functional code that shipped with the device...
• Multiple vulnerabilities
• No authentication
• And he works around the fact that the code doesn’t work to upload modified firmware images anyway
DEMO TIME

The things you own end up owning you.
Conclusion

With enough soap, one can blow up just about anything.
Playing along at home

• Know your network
  • M-SEARCH every network you connect to
  • Watch for new NOTIFY messages

• If you don’t need UPnP, disable it
  • If not on the device, then at the router

• Keep on top of firmware updates
  • Not always automatic
Playing along at home

• Fuzz the crap out of it
  • Burp – http://portswigger.net/burp/
  • WSFuzzer – https://www.owasp.org/index.php/Category:OWASP_WSFuzzer_Project
  • Miranda – http://code.google.com/p/miranda-upnp/
  • My stuff...if I ever release it, which I probably won’t...
Hit me up

- @HeadlessZeke on twitter
- Usually lurking on freenode as HeadlessZeke
- headlesszeke@hp.com
Thank you!