

U Plug, We Play

David Middlehurst



BSides Manchester
28 June 2014



Who am I?

David Middlehurst

Twitter: @dtmsecurity

- **Security Consultant / Engineer**
- **Specialisms:**
 - Application Security
 - Develops Security Testing Tools
- **Previous Internal / Client Research**
 - Near Field Communications (NFC)
 - Virtualisation



Agenda

- **Introduce UPnP**
- **Brief overview of UPnP specification**
- **Introduce a new UPnP assessment tool**
- **Tool overview**
- **Discover UPnP targets**
- **Interact with UPnP targets**
- **Learn and save UPnP targets**
- **Spoof UPnP targets on the network**
- **Demo**
- **UPnP device authorisation**
- **Common Flaws**
- **Implications of these flaws**
- **Places you should expect to find bugs**
- **Conclusion**
- **How to get the tool**



UPnP

Universal Plug and Play (UPnP) allows devices to discover each others presence on a network and identify services that are available



UPnP – Basic Discovery

Two main methods

- Devices can announce that they are there by sending a “NOTIFY” packet
- Devices can send out an “M-SEARCH” request asking for devices to reveal themselves if they meet the specified requirements



UPnP – NOTIFY Packet

Devices periodically send multicast UDP packets to indicate they are online. This includes headers for the type of service they are offering and the URL to the XML device description.

```
NOTIFY * HTTP/1.1  
Cache-Control: max-age = 300  
Host: 239.255.255.250:1900  
Location:  
http://192.168.0.15:8080/01c18570-b4a4-4356-8afa-5eeac61aa583/  
NT: urn:schemas-upnp-  
org:service:WANPPPConnection:1  
NTS: ssdp:alive  
SERVER: UPnP-Pentest-Toolkit  
USN: uuid:85e5f606-c406-4c89-8af3-5fc4ef27ee18
```



UPnP – M-SEARCH

A device can probe for other UPnP devices by sending an M-SEARCH request. This can include a header asking for devices that have a specific service.

```
M-SEARCH * HTTP/1.1  
HOST:239.255.255.250:1900  
ST:ssdp:all  
MAN:"ssdp:discover"  
MX:2
```



UPnP – M-SEARCH

Devices respond by sending a 'HTTP 1/1 200' response via UDP to the source IP and source port that the M-SEARCH packet originated from.

HTTP/1.1 200 OK

Cache-Control:max-age = 300

Date:

ST: urn:schemas-upnp-org:service:ContentDirectory:1

USN:uuid

Location: http://192.168.0.15:8080/01c18570-b4a4-4356-8afa-5eeac61aa583/

OPT:"http://schemas.upnp.org/upnp/1/0/"; ns=01

01-NLS:

Server: UPnP-Pentest-Toolkit

Ext:



New Tool Is Born – “UPnP Pentest Toolkit”

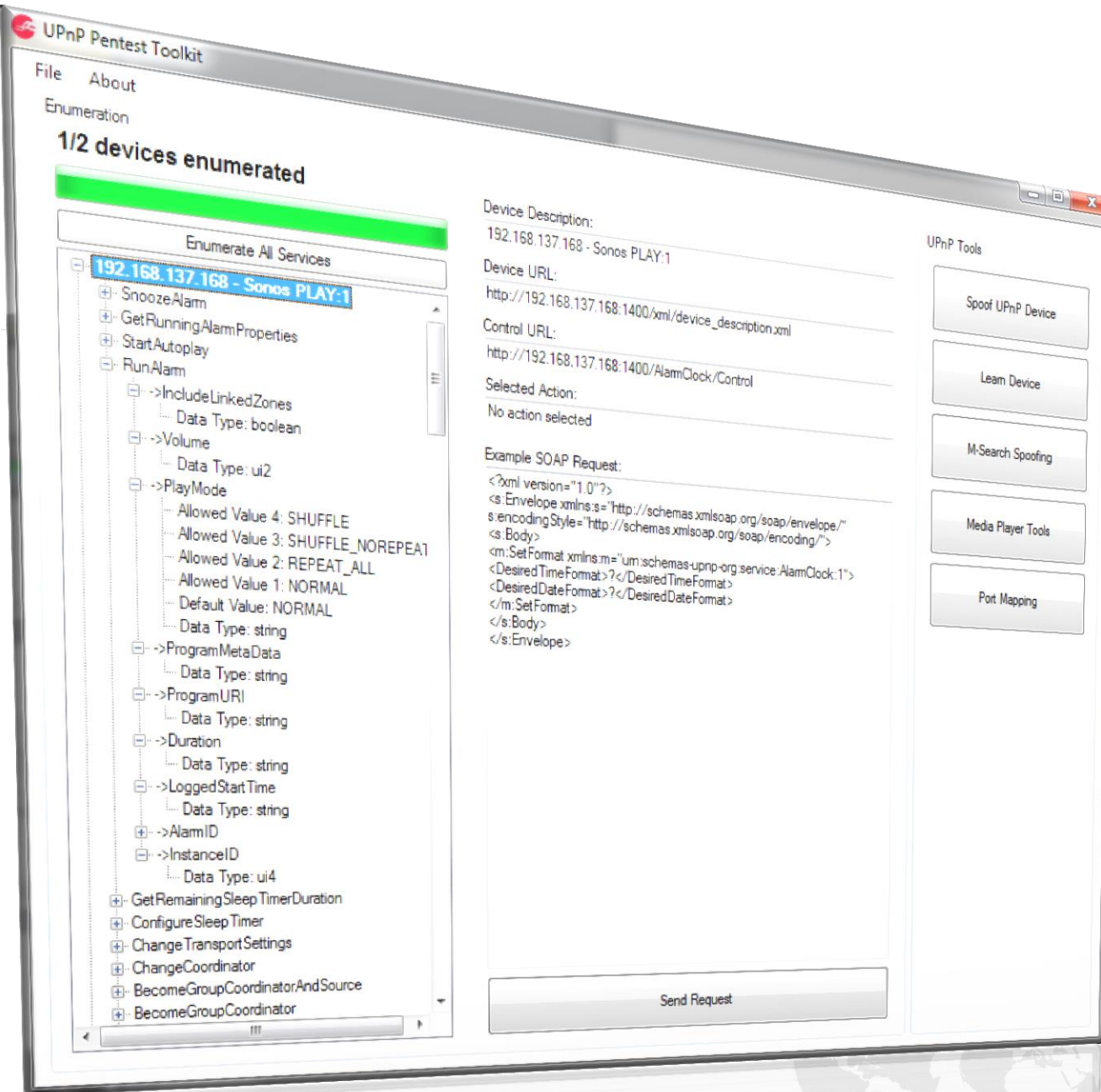
Motivation:

Lack of a security focused tool which provides easy access to the required information and allowed easy interaction and a framework to add new ideas.

Aim:

The aim of this tool is to bring together a range of UPnP assessment features, enabling a quick security assessment with minimal configuration and set-up.





Windows GUI tool

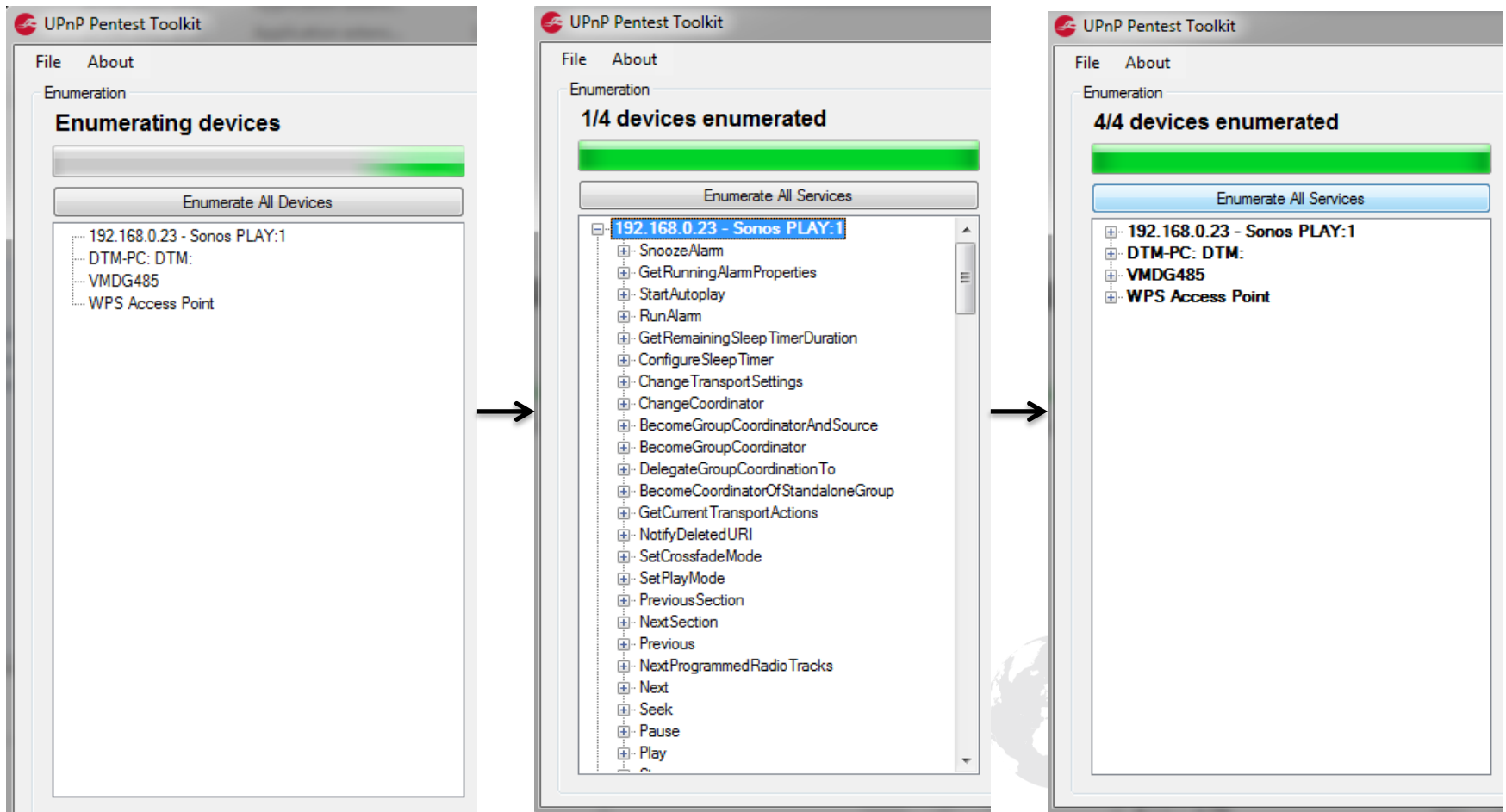
Written in C#

Uses the libraries:

- Managed UPnP
- PcapDotNet



Discovery



Interaction

Enumeration

4/4 devices enumerated

Enumerate All Services

- GetAutoplayLinkedZones
- SetAutoplayLinkedZones
- GetZoneInfo
- GetHouseholdID
- GetZoneAttributes
- SetZoneAttributes
- SeparateStereoPair
- CreateStereoPair
- RemoveBondedZones
- AddBondedZones
- GetInvisible
- SetInvisible
- GetLEDState
- SetLEDState**
 - DesiredLEDState
 - Allowed Value 2: Off
 - Allowed Value 1: On
 - Data Type: string
 - UpdateAvailableServices
- ListAvailableServices
- GetSessionId
- GetDailyIndexRefreshTime
- SetDailyIndexRefreshTime
- ListAlarms
- DestroyAlarm

Device Description:
192.168.0.23 - Sonos PLAY:1

Device URL:
http://192.168.0.23:1400/xml/device_description.xml

Control URL:
http://192.168.0.23:1400/DeviceProperties/Control

Selected Action:
SetLEDState

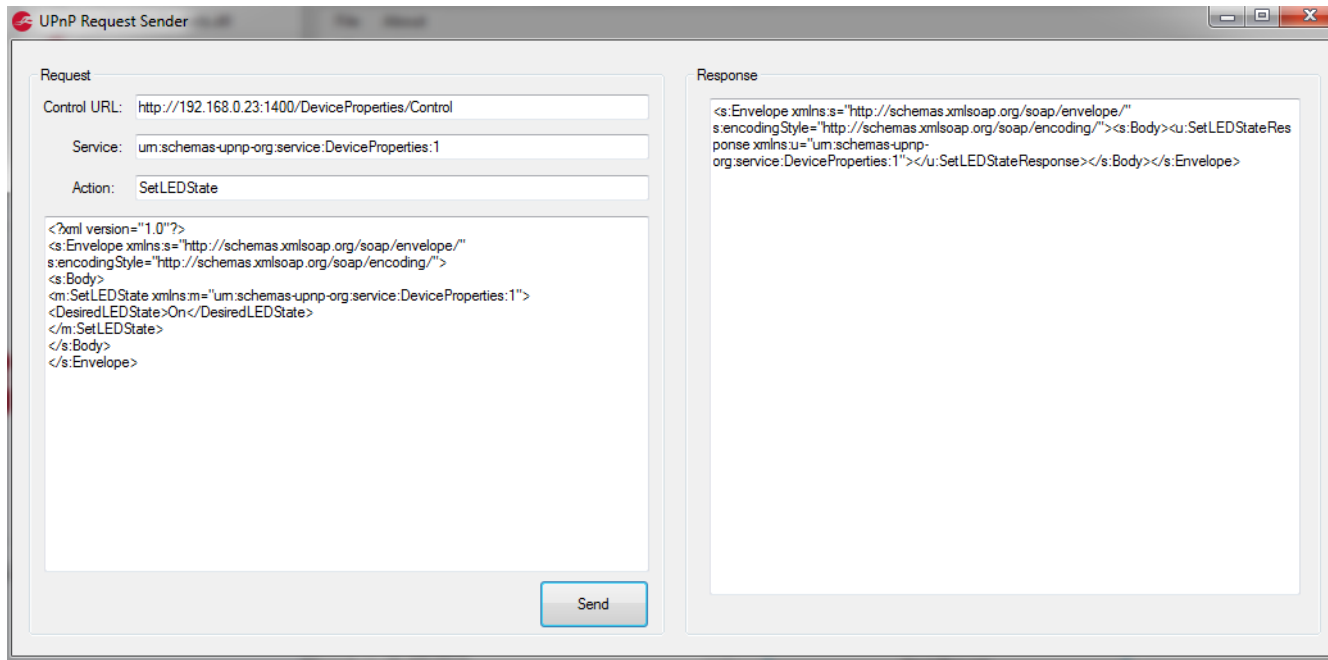
Example SOAP Request:

```
<?xml version="1.0"?>
<s:Envelope
xmlns:s="http://schemas.xmlsoap.org/soap/envelope/"
s:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
<s:Body>
<m:SetLEDState xmlns:m="urn:schemas-upnp-org:service:DeviceProperties:1">
<DesiredLEDState>On</DesiredLEDState>
</m:SetLEDState>
</s:Body>
</s:Envelope>
```

Send Request



Interaction



The screenshot shows the 'UPnP Request Sender' application window. It is divided into two main sections: 'Request' and 'Response'.

Request Section:

- Control URL: `http://192.168.0.23:1400/DeviceProperties/Control`
- Service: `urn:schemas-upnp-org:service:DeviceProperties:1`
- Action: `SetLEDState`

Request XML:

```
<?xml version="1.0"?>
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/"
s:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
<s:Body>
<m:SetLEDState xmlns:m="urn:schemas-upnp-org:service:DeviceProperties:1">
<DesiredLEDState>On</DesiredLEDState>
</m:SetLEDState>
</s:Body>
</s:Envelope>
```

Response Section:

Response XML:

```
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/"
s:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"><s:Body><u:SetLEDStateResponse xmlns:u="urn:schemas-upnp-org:service:DeviceProperties:1"></u:SetLEDStateResponse></s:Body></s:Envelope>
```

A 'Send' button is located at the bottom right of the Request section.



Learning

The image shows a screenshot of a software application window titled "Learn" and a Windows Explorer window overlaid on top.

Learn Application:

- Devices:** A list containing one entry: `http://192.168.0.13:40000/device_description`.
- Log:** A text area showing the following content:
 - Device: `http://192.168.0.13:40000/device_description`
 - Action: `ExportResource`
 - Base SOAP Response:

```
<?xml version="1.0" encoding="utf-8"?>
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/"
s:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
<s:Body>
<u:ExportResource xmlns:u="um:schemas-upnp-
org:service:ContentDirector
</u:ExportResource>
</s:Body>
</s:Envelope>
```
 - Action: `UpdateObject`
 - Base SOAP Response:

```
<?xml version="1.0" encoding="utf-8"?>
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/"
s:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
<s:Body>
<u:UpdateObject xmlns:u="um:schemas-upnp-
org:service:ContentDirector
</u:UpdateObject>
</s:Body>
</s:Envelope>
```
 - Action: `GetTransferProgress`
 - Base SOAP Response:

```
<?xml version="1.0" encoding="utf-8"?>
```

Windows Explorer:

- Path: `Libraries > Documents > Work > UPnP Research > UPT`
- File name: `zero DivX Media Server.upt`
- Save as type: (empty)
- Buttons: `Save`, `Cancel`

Spoofting

ComeGetIt - Spooft UPnP Device

Control
Network Interface: (Network adapter 'Microsoft' on local host) rpcap://^Device\NPF_{CEEA36BB-...}
Start Stop Cycle UDN GUID

Advertise UPnP Device:
NOTIFY Beacon:
NOTIFY * HTTP/1.1
Cache-Control: max-age = 300
Host: 239.255.255.250:1900
Location: http://192.168.0.6:9090/3203b3b9-d878-49a3-b6e7-c761fe154921/
NT: upnp:rootdevice
NTS: ssdp:alive
SERVER: UPnP-Pentest-Toolkit
USN: uuid:3203b3b9-d878-49a3-b6e7-c761fe154921

Beacon Interval (ms):
1000

Source IP: 192.168.0.6 **Source Port:** 1900
Dest IP: 239.255.255.250 **Dest Port:** 1900

M-Search Responder
[Started Capture] Network adapter 'Microsoft' on local host
Log:
M-Search from 192.168.0.6:50934[25/06/2014 22:03:33]
M-Search from 192.168.0.13:54680[25/06/2014 22:03:27]
M-Search from 192.168.0.6:53379[25/06/2014 22:03:27]
M-Search from 192.168.0.6:53379[25/06/2014 22:03:26]
M-Search from 192.168.0.6:53379[25/06/2014 22:03:25]
M-Search from 192.168.0.6:53379[25/06/2014 22:03:24]
M-Search from 192.168.0.6:53379[25/06/2014 22:03:22]
M-Search from 192.168.0.6:53379[25/06/2014 22:03:21]
M-Search from 192.168.0.6:53379[25/06/2014 22:03:20]
M-Search from 192.168.0.6:53379[25/06/2014 22:03:19]

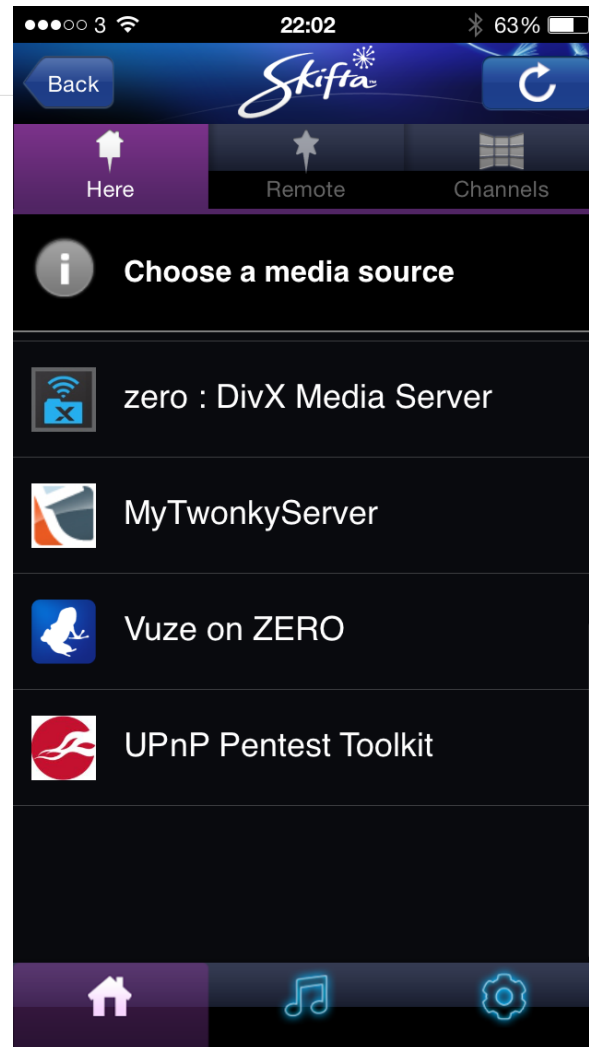
Request:
From: 192.168.0.6:50934
M-SEARCH * HTTP/1.1
HOST: 239.255.255.250:1900
MAN: "ssdp:discover"
MX: 1
ST: urn:dial-multiscreen-org:service:dial:1
USER-AGENT: Google Chrome/35.0.1916.153 Windows

Response:
 Send Multiple Responses (For each known device type)
HTTP/1.1 200 OK
Cache-Control: max-age = 300
Date: Wed, 25 Jun 2014 21:03:33 GMT
ST: urn:dial-multiscreen-org:service:dial:1
USN: uuid:3203b3b9-d878-49a3-b6e7-c761fe154921
Location: http://192.168.0.6:9090/3203b3b9-d878-49a3-b6e7-c761fe154921/
OPT: "http://schemas.upnp.org/upnp/1/0/"; ns=01

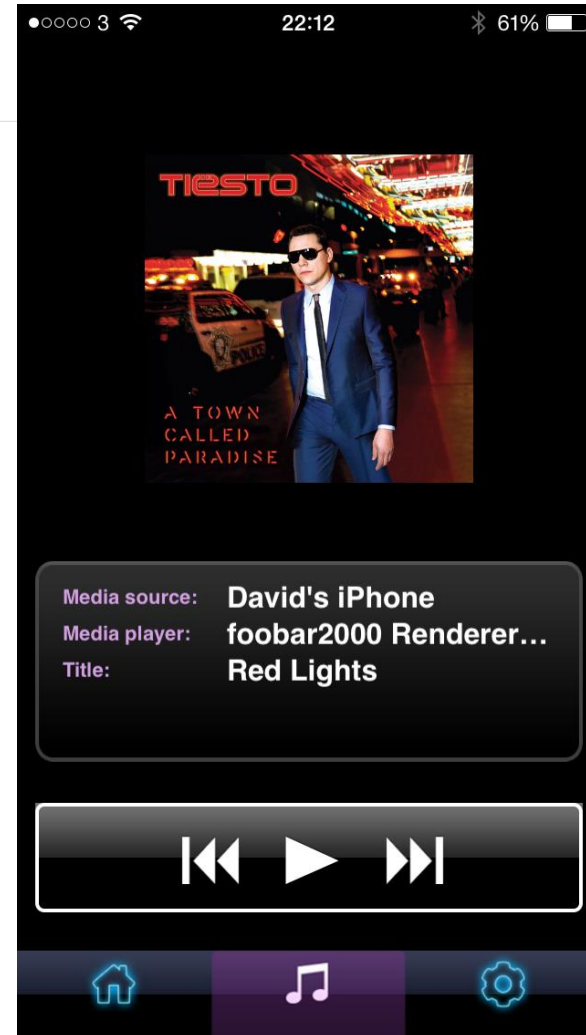
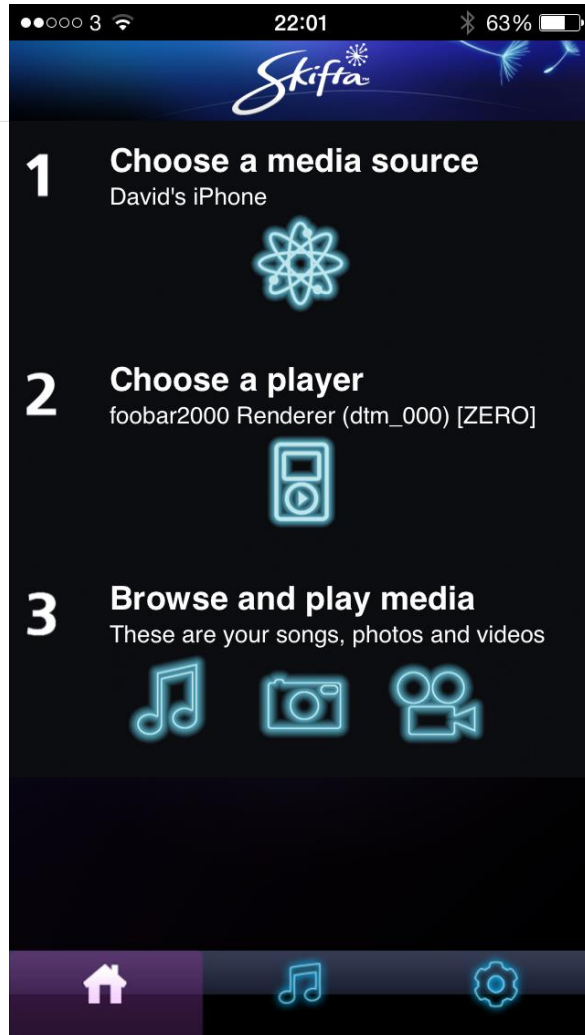
Web Server
162 hits
Device Description URL: Use Custom URL Web Server Port: 9090
http://192.168.0.6:9090/3203b3b9-d878-49a3-b6e7-c761fe154921/
SCPD URL:
http://192.168.0.6:9090/db9c455d-4ce8-42f6-b8df-e56e301aef1/
Log:
GET from: 192.168.0.13:55893 [25/06/2014 22:03:33]
/3203b3b9-d878-49a3-b6e7-c761fe154921/
GET from: 192.168.0.13:55896 Mozilla/4.0 (Compatible; Aggregation, Twonky, 7.3, Windows 6.2 (Build 9200); x86_64) DLNADOC/1.50 [25/06/2014 22:03:32]
/db9c455d-4ce8-42f6-b8df-e56e301aef1/
GET from: 192.168.0.13:55895 Mozilla/4.0 (Compatible; Aggregation, Twonky, 7.3, Windows 6.2 (Build 9200); x86_64) DLNADOC/1.50 [25/06/2014 22:03:32]
/3203b3b9-d878-49a3-b6e7-c761fe154921/
GET from: 192.168.0.13:55894 [25/06/2014 22:03:32]
/3203b3b9-d878-49a3-b6e7-c761fe154921/
GET from: 192.168.0.13:55892 [25/06/2014 22:03:31]
/3203b3b9-d878-49a3-b6e7-c761fe154921/
Current Device Description:
<?xml version="1.0"?><root xmlns="urn:schemas-upnp-org:device-1-0"><specVersion><major>1</major><minor>0</minor></specVersion><device><UDN>uuid:3203b3b9-d878-49a3-b6e7-c761fe154921</UDN><friendlyName>UPnP Pentest Toolkit</friendlyName><deviceType>urn:schemas-upnp-org:device:MediaServer:1</deviceType><manufacturer>UPnP Pentest Toolkit</manufacturer><manufacturerURL>http://192.168.0.6:9090/3203b3b9-d878-49a3-b6e7-c761fe154921/</manufacturerURL><modelName>UPnP Pentest Toolkit</modelName><modelName>1.0</modelName><modelURL>http://192.168.0.6:9090/3203b3b9-d878-49a3-b6e7-c761fe154921/</modelURL><serialNumber>3203b3b9-d878-49a3-b6e7-c761fe154921</serialNumber><dlna:X_DLNADOC xmlns:dlna="urn:schemas-dlna-org:device-1-0">DMS-1.50</dlna:X_DLNADOC><microsoft.magicPacketWakeSupported

Saved Device
Load Saved Device Clear
Device:
No Device Loaded
SOAPAction Log:

Spoofting



Spoofting – Learnt Device



Spoofing – Learnt Device

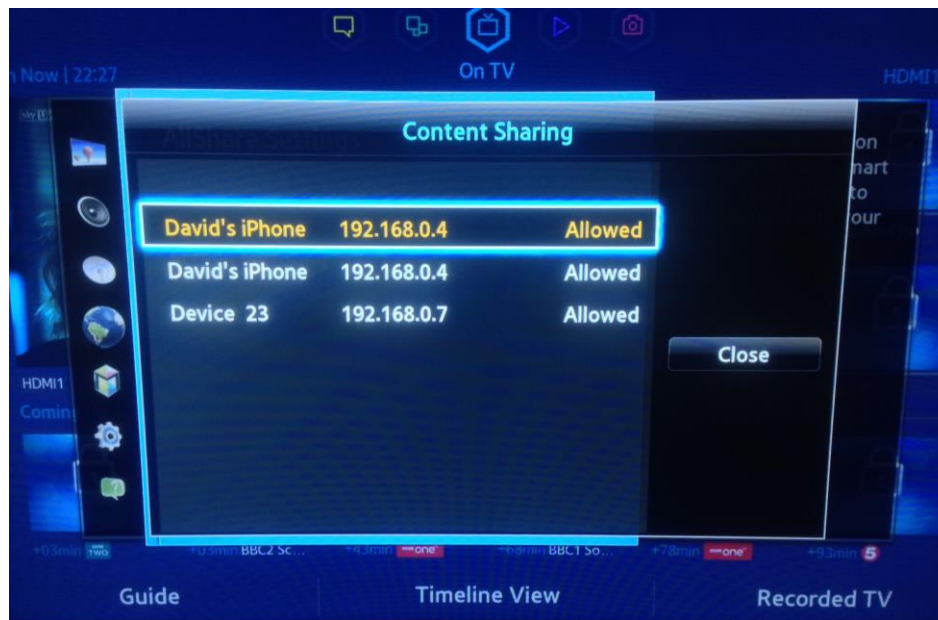
The screenshot shows the 'ComeGetIt - Spoof UPnP Device' application interface, which is divided into several functional panels:

- Control:** Features a 'Network Interface' dropdown menu, 'Start' and 'Stop' buttons, and a 'Cycle UDN GUID' checkbox.
- Advertise UPnP Device:** Contains a 'NOTIFY Beacon' text area with configuration details like 'Cache-Control: max-age = 300' and 'Host: 239.255.255.250:1900'. Below it is a 'Beacon Interval (ms)' slider set to 1000, and fields for 'Source IP', 'Source Port', 'Dest IP', and 'Destination Port'.
- Web Server:** Displays '1599 hits' and includes fields for 'Device Description URL', 'SCPDL URL', and 'Log'. It also has a 'Current Device Description' XML view and a 'Fuzz cases' section with an 'Open' button and a 'Load Fuzz Cases' checkbox.
- M-Search Responder:** Shows a log of incoming M-Search requests and a corresponding 'Response' field containing UPnP service discovery data.
- Saved Device:** Includes 'Load Saved Device' and 'Clear' buttons, a 'Device:' section with a tree view of actions, and a 'SOAP Action Log' section showing the source, action, and request/response XML snippets.

Demo



Authorisation - Good Vendor Practices



Select which of the devices on this network are allowed to access shared media on your device.

Share with Network Devices

- Unknown Device [192.168.0.23]
- DTM-PC: DTM:
- ZERO:
- BLACKBERRY-BEF6
- Unknown Device [192.168.0.13]
- Unknown Device [192.168.0.23]

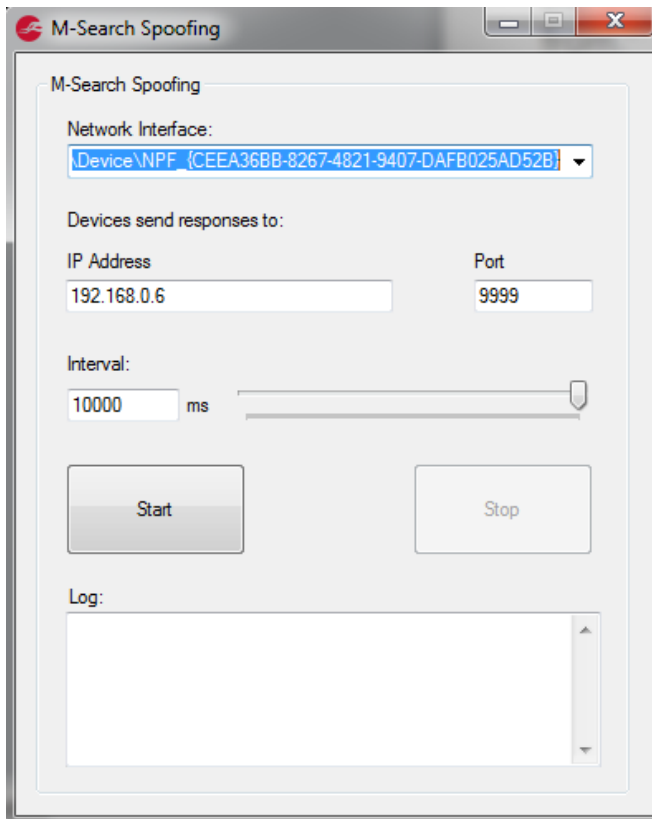


- **Mostly it's a case of:**
 - **“What do you mean by authorisation?”**
- Security by obscurity
 - Looking for fixed User Agents
 - Deviating from the specification slightly



Flaws – M-SEARCH Spoofing

“Devices respond by sending a 'HTTP 1/1 200" response via UDP to the **source IP** and **source port** that the M-SEARCH packet originated from.”



60% of devices* I looked at will respond to any routable destination

* Devices that should respond to M-SEARCH requests i.e. Media Servers responding to Media Players

Of these device mostly all devices did not care about what UDP port they send traffic to. Those that did disallowed low ports i.e. 1-1024



Flaws – External Device URL's

NOTIFY * HTTP/1.1
Cache-Control: max-age = 300
Host: 239.255.255.250:1900
Location: **http://192.168.0.6:9090/685f34e2-9a75-49c8-b642-2b1586ad4433/**
NT: urn:schemas-upnp-org:service:WANPPConnection:1
NTS: ssdp:alive
SERVER: UPnP-Pentest-Toolkit
USN: uuid:685f34e2-9a75-49c8-b642-2b1586ad4433

HTTP/1.1 200 OK
Cache-Control:max-age = 300
Date:Wed, 25 Jun 2014 22:19:27 GMT
ST:upnp:rootdevice
USN:uuid:685f34e2-9a75-49c8-b642-2b1586ad4433
Location:**http://192.168.0.6:9090/685f34e2-9a75-49c8-b642-2b1586ad4433/**
OPT:"http://schemas.upnp.org/upnp/1/0/"; ns=01
01-NLS:2404a5f72618151d22759c04a8cad0b6
Server:UPnP-Pentest-Toolkit
Ext:

What happens if we specify a URL outside of the local subnet i.e. the internet

40% of devices / apps* that I looked at will make requests to any routable destination

* That actually have a need to go and get device URLs i.e. Media Players looking for Media Servers

Of these device mostly all devices did not care about what TCP port they send traffic to.

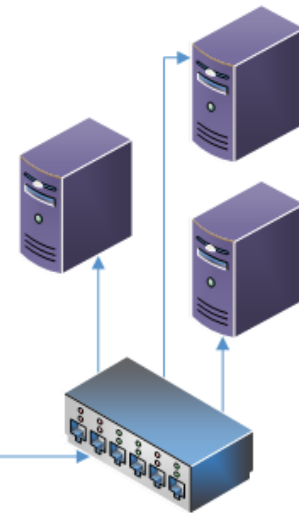
Flaws – External Traffic

Possible implications of UPnP services not taking note of where they are sending traffic

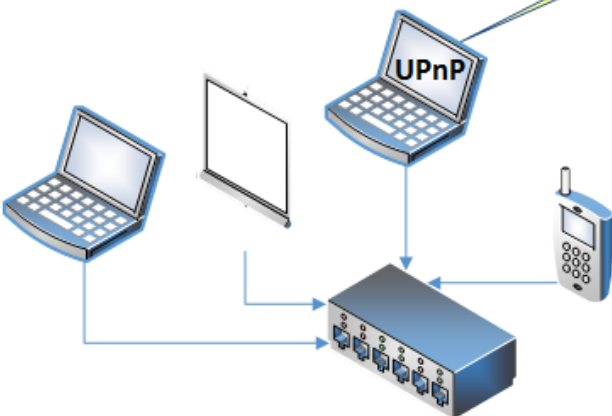
- Sounds like Denial of Service (DoS)?
 - Attacker sends one packet → Multiple packets sent to a target
 - Attacker sends one packet → Multiple devices send packets to a target
 - Send a small packet → Bigger packets sent to a target
- Influence over audit activity:
 - Malware Beacon Signatures
 - Dodgy Web Sites
- Port scanning
- Trust Boundaries...

Trust Boundaries

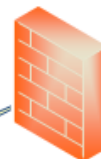
TRUSTED LAN



UNTRUSTED LAN



VPN



Connect your work laptop to your home LAN (Untrusted)

You have some UPnP service i.e. Running a media player

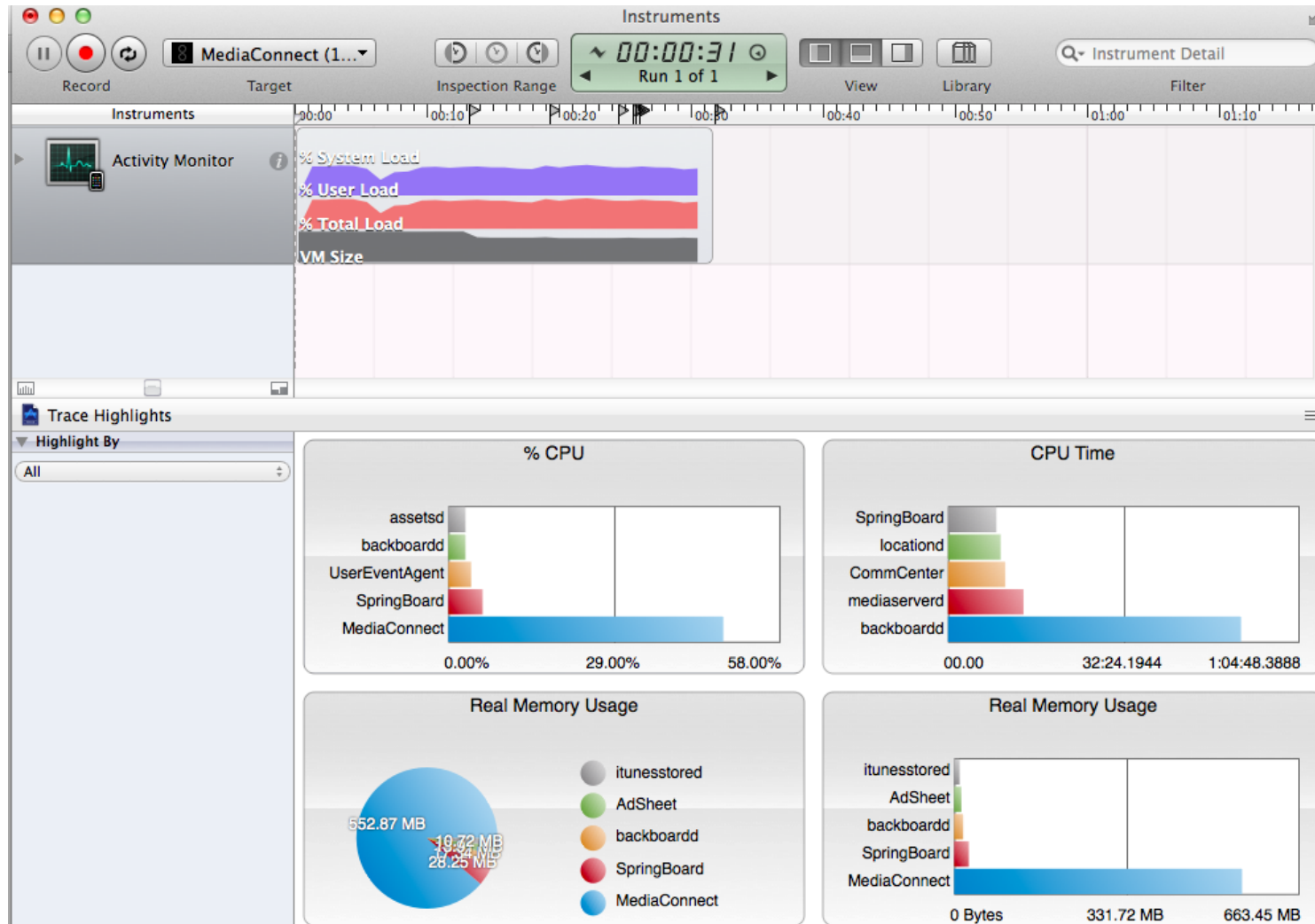
Attacker on your home LAN can cause UDP/TCP traffic on your work LAN (Trusted)

Bugs to look for

- Do devices and apps impose limits on the size of files they attempt to download?
- Do they assume that all they have to process is a small XML file?
- How do they handle traditional XML base attacks? i.e. XML Entity Injection (XXE)
- When they have parsed the information do they safely handle it?

Memory Management

Many UPnP implementations reviewed have problems i.e. Lots of iOS apps



Where are we heading?

“**UPnP+** will expand beyond the group's traditional work on devices using Internet Protocol on WiFi LANs to create bridges to wide-area networks and non-IP devices.”

"Today there's not a good standard for accessing devices across any network, but UPnP [Forum] is working on it," said Alan Messer, a Samsung researcher in San Jose who is vice president of UPnP [Forum].“

Conclusion

- **I think UPnP can be a hidden gem on security assessments**
- **The more a security professional experiences UPnP the more alarm bells start to ring**
- **Vendor best practices vary to huge degree**
- **I have introduced new common classes of bug. I am sure there are plenty of bugs to be identified.**
- **Hopefully UPnP Pentest Toolkit will be a useful tool for people who want to assess UPnP targets or get started with carrying out research in this area**
- **The future looks even more scary! Watch this space, hopefully I will present on some of this in the years to come**

Get UPnP Pentest Toolkit



**Goto:
upnp.ninja**

NCC Group GitHub:

<https://github.com/nccgroup/UPnP-Pentest-Toolkit>

Keep updated

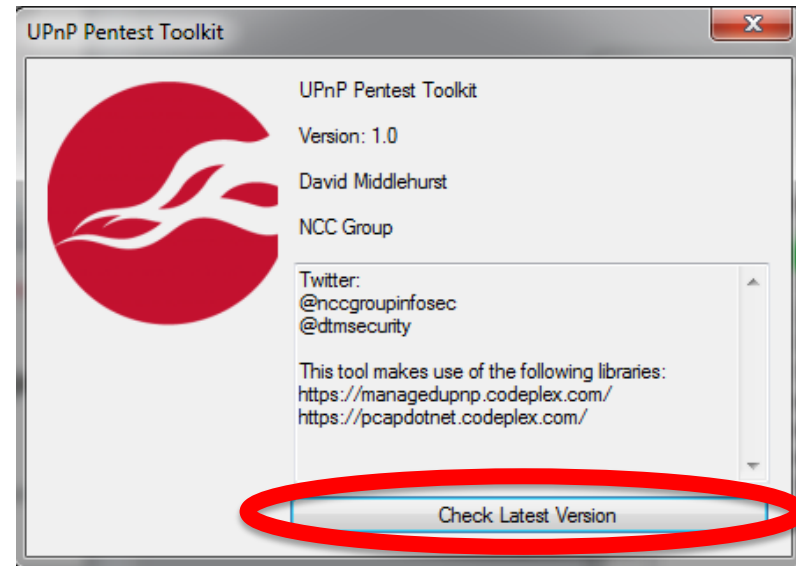
Tool:

Via GitHub page
or the
'About' Dialog

Research:

Follow

@dtmsecurity
@nccgroupinfosec



Questions?



UK Offices

Manchester - Head Office
Cheltenham
Edinburgh
Leatherhead
London
Milton Keynes

European Offices

Amsterdam - Netherlands
Munich – Germany
Zurich - Switzerland



North American Offices

San Francisco
Atlanta
New York
Seattle



Australian Offices

Sydney