Universal plug and play (UPnP) mapping attacks

Daniel Garcia

Abstract

Universal Plug and Play is a popular method for NAT traversal used by common household devices. This document explores the different techniques attackers can use to exploit port mapping services of UPnP/IGD devices on WAN ports. It also details a tool called Umap that can do manual port-mapping(WAN to LAN, WAN to WAN), nat-traversal and SOCKSv4 proxy service that automatically maps to UPnP devices. Devices with WAN ports allowing UPnP actions are the minority, but still a big threat.

Introduction

Universal Plug and Play(UPnP) is a technology developed by the UPnP Forum in 1999, after funding mainly by Microsoft. The goal set by the UPnP forum, at that time, was to allow devices to connect seamlessly and simplify network implementations. The only problem with this goal is that it is inherently insecure. A secure system can't be plug and play, it needs to ask questions and validate information. This is exactly one of the main problems in UPnP, as it lacks any form of authentication.

To worsen the situation, control points are sometimes configured to accept requests from the LAN and WAN side of the device. The control points are URL's where the SOAP requests are directed for the execution of actions in UPnP. The most common actions used are AddPortMapping and DeletePortMapping, used for the port mapping of devices wanting to traverse the NAT.

UPnP Steps

0. **Addressing:** Interaction with the addressing methods used by the devices. It also establishes rules for devices that are unable to get an address through DHCP.

1. **Discovery:** Discovery and announcement of the devices using SSDP. The devices send multicast search requests using HTTPU. Control points respond with HTTPU packets that specify a location for the XML description file.

2. **Description:** After the discovery of the XML description file location, the device downloads the XML to discover the different services and actions that the device has available.

3. **Control:** Through the description process, the device learns vital information to interact with the control point. At this point it sends SOAP requests(actions) to the specified control points to execute the different functions on the control point. This is where the actual execution of the actions like AddPortMapping and DeletePortMapping happen.

4. **Eventing:** Control points listen to changes in devices

5. **Presentation:** The referral to an HTML-based user interface for controlling and/or viewing the device status.

Vulnerabilities

The first problem reported for UPnP was a Denial of Service attack reported by Ken from FTUSecurity and applied to the Microsoft Windows 98/ME/XP stack. Afterwards eEye published an advisory for a buffer overflow attack, also on the Microsoft stack. In 2003 Björn Stickler published an information disclosure advisory for the Netgear FM114P, the information disclosure was based on using the GetUserName action of UPnP. Then in 2006 Armijn Hemel reported the vulnerability on remote users being able to use UPnP to forward packets on external hosts. He also published his findings on the www.upnp-hacks.org site, one of the best sources of UPnP hacking information up to date. This flaw highlighted by Armijn is what Umap relies on for the port mapping.

The main workings of Umap rely on the "AddPortMapping" and "DeletePortMapping" actions in the UPnP protocol. They are meant to be used by devices on a LAN that want to traverse a NAT. Unfortunately, these control points are also available on the WAN interfaces of the devices, allowing attackers to add a port map from the external WAN IP to any host desired. The attacker can map a port on the external IP and forward that traffic to another external host. The attacker can also map external ports on the WAN IP to internal hosts behind the NAT of the device. This allows the attackers to scan for hosts inside the NAT, forward traffic to external hosts and forward traffic to internal hosts. Some routers, have an open control point by default. In fact, some routers keep accepting UPnP requests after disabling UPnP WAN requests.

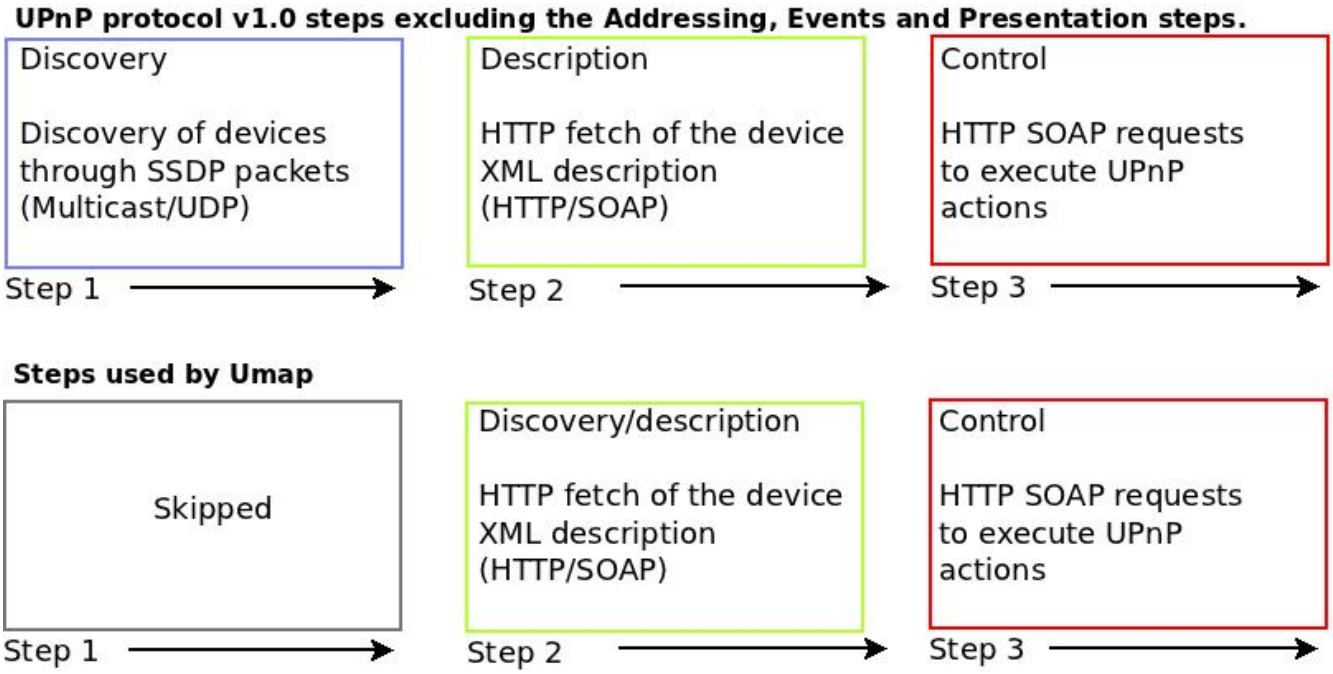
There are many problems besides port mapping: information disclosure, command execution and DoS. For example, another problem that is less intrusive is the disclosure of information regarding the device. On average the minimum information you can get from UPnP IGD devices on the WAN side are the MAC address, serial number and device model. This information could be used by attackers as an identifier to locate modems on dynamic IP pools or just to target.

Umap

Umap is designed to work in different modes:

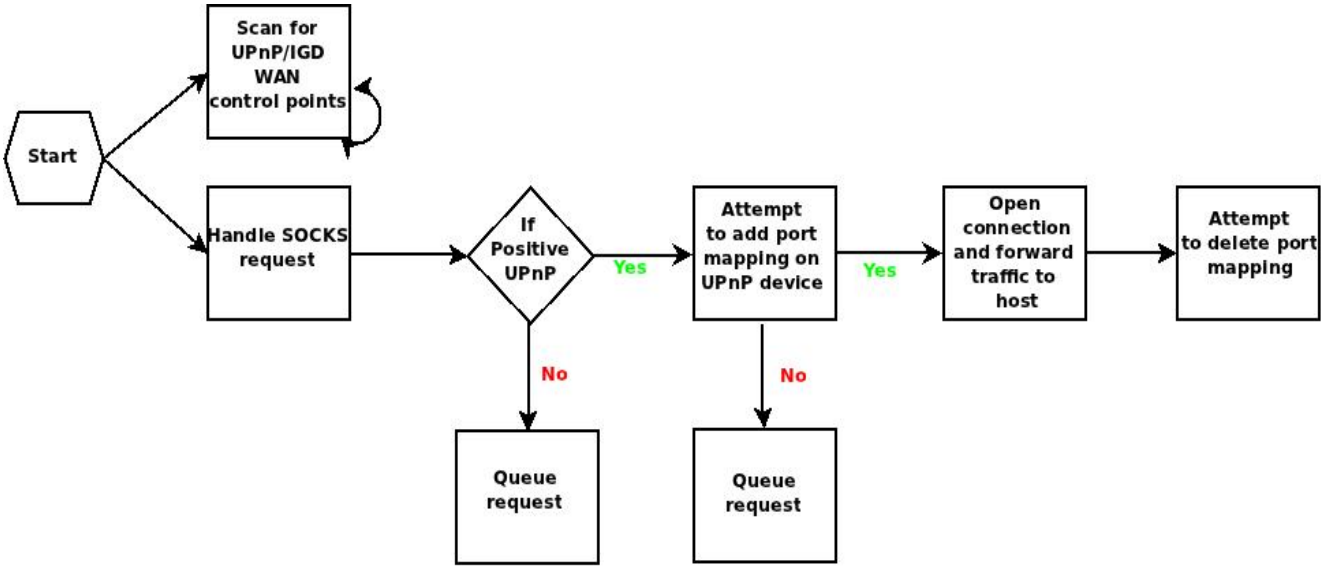
- Scanner for UPnP devices with exposed WAN control points
- SOCKSv4 proxy that forwards traffic through devices with exposed control points
- Scanner/mapper of internal hosts behind a NAT of a device with exposed control points
- Manual TCP/UDP mapping of exposed control points

There is not a lot of PoC on UPnP publicly available. A clever exploit that sends UPnP commands through the execution of javascript on the victim's browser was created by GNUCitizen. There is also a tool available named Miranda by SecuriTeam. Its pretty good and works well manipulating UPnP devices to execute actions. This tool, however, is designed for LAN use only as it relies on SSDP and multicast for the discovery of UPnP devices, which makes a lot of sense since the UPnP protocol v1.0 states that it is the standard way of discovering UPnP devices. Umap, on the other hand, skips this step and simply tries to fetch the XML descriptions of the devices. Relying on the Unicast part of the UPnP transaction makes it suitable for scanning UPnP on WAN scenarios.

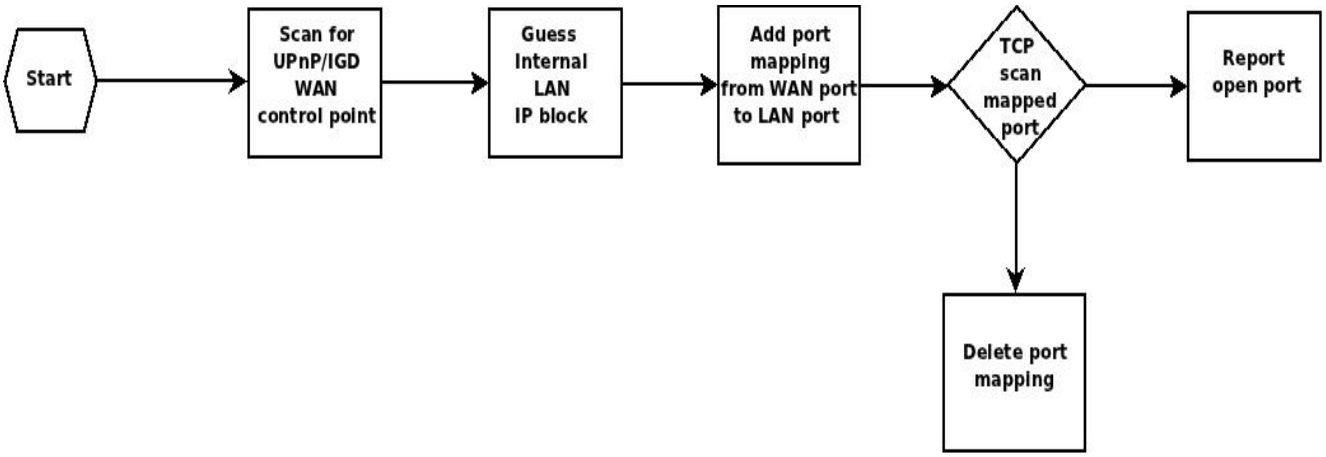


It relies on a database of common locations and ports for XML description files on UPnP devices. After it fetches those description files it tries to execute the AddPortMapping and DeletePortMapping actions. For the internal network scanning, it tries to guess the internal IP set by the device and scans each host for a group of common ports or the ports specified by argument.

Flow diagram on SOCKSv4 mode



Flow diagram on scanner mode



Negative aspects of UPnP mapping

There are many aspects on UPnP mapping that are not favorable. The biggest impact is performance when using other routers. Most UPnP devices are residential gateways/CPEs that have a very limited upload bandwidth. Another factor that affects performance greatly is the unpredictability of the different UPnP stacks on executing the actions for the mapping. Most vendors cap the amount of port mappings in the stack, limiting the amount of mappings. Some devices only allow, 10 mappings at a given time, which lowers the performance of UPnP mapping in heavy connection scenarios like web-browsing.

In terms of the noise made by the attack, some devices actively log the port mappings with the source IP of the request. Unfortunately, residential users do not care/read the logs of their devices. The operators that own the lines for the devices could implement centralized logging solutions which could allow some kind of mitigation for the problem.

Mitigations

The mitigation falls down to two elements: Operators and Users. Users can mitigate by reconfiguring their devices to disallow WAN traffic to a UPnP control point. Some IGD devices only allow enabling/disabling UPnP services, without the ability to indicate if you want to receive WAN traffic to the UPnP control point. Disabling UPnP completely is sometimes troublesome, some devices require UPnP for NAT traversal.

Operators can mitigate either by blocking WAN requests to client devices or by deploying the devices with base configurations that disable the UPnP WAN requests. Using base configuration packages is a better solution because some UPnP stacks rely on port 80 for the transmission of UPnP SOAP requests. Blocking WAN traffic could block user management interfaces for the devices.

Disabling UPnP totally is a nightmare, because a lot of devices use UPnP to traverse. Gaming consoles are the perfect example of devices that need UPnP for better performance. The only reason I would recommend disabling UPnP is if you have a stack that keeps accepting WAN requests even if you specify that you don't want WAN requests.

Affected devices

I have scanned different IP pools around the world looking for different stacks of UPnP devices. During a 1 week period I discovered more than 150,000 devices, just by scanning random DSL IP pools. The speedtouch stack is by far the most common. There may be many other devices vulnerable on-line, but I don't think there has been a lot of research around that subject.

Manufacturer	Model	Version
Linksys	WRT54GX	< 4.30.5
Edimax	BR-6104K	< 3.21
Sitecom	WL-153	< 1.39
Speedtouch/Alcatel/Thomson	5x6	< 6.2.29
Thomson	TG585 v7	< 7.4.3.2