2013 27th International Conference on Advanced Information Networking and Applications Workshops

Determining Home Users' Vulnerability to Universal Plug and Play (UPnP) Attacks

Shadi Esnaashari, Ian Welch
School of Engineering and Computer Science
Victoria University of Wellington
Wellington, New Zealand
{*Shadi.Esnaashari, Ian.Welch*}*@ecs.vuw.ac.nz*

Peter Komisarczuk
School of Computing and Technology
University of West London
London, United Kingdom
peter.komisarczuk@uwl.ac.uk

***Abstract*—Universal Plug and Play (UPnP) technology is used worldwide since it has simplified the installation and management of the devices. As a result, many devices are now equipped with UPnP capabilities. Unfortunately using UPnP in home routers puts routers at risk of abuse. For example, it is easier for hackers to discover the devices and use device vulnerabilities in order to make malicious attacks to cause financial or reputational damage to the users. In this paper, we have analyzed the UPnP protocol and its different vulnerabilities. Furthermore, we have emphasized how common the problem is with the home users' devices. Hence, we suggest a tool to achieve transparency in the health of the Internet by detecting UPnP enabled devices which are likely to be attacked on home networks. The tool will look for UPnP based attacks when people's routers have been compromised. The tool is easy to install and use for novice home users and maintains their privacy too. This project aims not only to implement a tool for a user to determine whether his/her system is vulnerable to a particular attack, but also to measure the prevalence of vulnerabilities at national or global level. Thus a larger framework is required to collect and manage the results from individual users.**

***Keywords*-Security, Network Measurement, UPnP.**

I. Introduction

Most home networks are connected to the Internet via gateway devices (Internet Gateway Devices or IGDs) that use Network Address Translation (NAT) [1] to allow sharing of a single external IP address and implement firewalls rules to limit the visibility of internal devices. Adding new devices to the home network can be done transparently as long as the devices on the home network only make outgoing connections and use permitted ports. However, some devices such as game consoles or Voice-over-IP (VOIP) devices also require the ability to receive incoming connections requiring reconfiguration of the IGD's NAT tables and firewall rules.

Many manufacturers have added Universal Plug and Play (UPnP) support to their IGDs to make the reconfiguration of the IGDs transparent . Universal Plug and Play (UPnP) [2] is an architecture for peer to peer network connectivity from different types of devices such as PCs, wireless devices, and applications. It simplifies the integration of new network devices into a home network by allowing auto-configuration rather than requiring the user to manually configure both the new device and devices with whom it interacts [3].

UPnP makes it easier for users to add new devices to their network but it also adds security vulnerabilities due to manufacturers of UPnP devices not implementing authentication and authorization despite guidance for these being included in the UPnP specification. The usual model implemented is to trust all requests from machines in the same Local Area Network (LAN). This means that if hackers can convince a user to execute code on a machine on the same LAN, the code can reconfigure the router to the advantage of the attacker. For instance, when a request comes from an attacker to open the firewall to allow further use, permission will be given because the computer trusts the machine. The simplest examples of UPnP based attacks are misrouting DNS, changing settings, rebooting, getting and changing usernames/passwords, internal/external forwarding, and code execution.

Of particular concern, is the potential for home users to be subject to a man-in-the-middle attack (perhaps to intercept banking details) implemented using UPnP's ability to reroute connections using port mapping or changing Domain Name Server (DNS) settings.

Although home users who are not using gaming consoles or VOIP devices are often advised to turn off UPnP to secure their network against such attacks, it is likely that many novice users do not do this because they do not have the skills to configure their own routers. In some devices such as Alcatel/Thomson Speedtouch 510, the process of enabling or disabling will be done through the command line which is not easy. It is likely than many home users have UPnP enabled by default despite not requiring it and are leaving themselves vulnerable.

UPnP security vulnerabilities have been known at least since 2001 [4] although initially the focus was on vulnerabilities associated with the implementation of the protocol stack. However, not much is known about the actual prevalence of vulnerable IGDs. Aside from one study (see later), we do not know how many home users have both UPnP turned on and are vulnerable to security attacks.

The goal of our work is to develop a tool that would allow participating home users to test their IGDs for vulnerabilities and allow central collection of this information so we can gauge the prevalence of the vulnerabilities.

978-0-7695-4952-1/13 $26.00 © 2013 IEEE
DOI 10.1109/WAINA.2013.225

The rest of this paper is organized as follows. Section II provides an overview of the UPnP protocol and its vulnerability to man-in-the-middle attacks. Section III outlines what work has been done on the detection of UPnP security vulnerabilities. Section IV provides an overview of the design and implementation of our vulnerability scanner tool. Section V outlines how we have tested the tool. Finally in Section VI we outline future work before concluding in Section VII.

II. Universal Plug and Play (UPnP)

UPnP is an extension to PnP (Plug-and-Play) first implemented by Microsoft in early 1999 and developed by UPnP Forum. UPnP is implemented for different operating systems such as Windows 98/98SE/ME/XP, VxWorks, Linux, and FreeBSD. It is used in different devices such as printers, scanners, file servers, cameras, kitchen whiteware, routers, and firewalls. It is also used in different programs such as Microsoft's MSN Messenger, networking games such as X-Box Live, and Voice over IP (VoIP).

UPnP is a combination of open networking protocols including TCP, IP, UDP, HTTP (RFC 2616), SOAP [5], GENA [6], and XML. The TCP/IP networking protocol stack is used as the base on top of which other UPnP protocols are built. In the UPnP network, devices, services, and control points are the three main components. A UPnP device contains the services. A service consists of a state table, control server and event server. The state table shows the service through the state variable. The state table will be updated when state changes in the network. The control server is responsible for receiving and executing the action request and consequently, making updates on state tables. The event server is responsible for publishing events whenever the state of the service changes. Control points will discover the devices and the available services on them. They also invoke actions on the services and subscribe for the events in the UPnP networks. Devices respond to the action invoked requests and also send the events when changes occur to the state variables.

The typical life-cycle for the configuration of a UPnP device is as follows:

1) When a UPnP device joins the network, it needs to obtain an IP address. An IP can be obtained through the Dynamic Host Configuration Protocol (DHCP) or if the DHCP server is not available, the device will assign an IP address from range 192.254/16.
2) When the device joins the network after obtaining an IP address, it must discover the other UPnP enabled devices on the same network and what services these devices offer. Therefore, it will send the discovery message by using User Datagram Protocol (UDP) to the multicast address 239.255.255.250 on port 1900. All the UPnP enabled devices will send back a UDP unicast to the source of the multicast. This response message contains the location or URL where a XML device description can be downloaded.
3) The XML device description is downloaded and parsed. Included in the device description is a list of all the services provided by a device and information about the model, type, and serial number. A key part of the description is an URL which is crucial for the controlling the device and registering for event notifications from the device.
4) Controlling the device is done by sending action requests represented using SOAP messages. The target of the messages is the URL from the XML description file.

The UPnP specification describes an authentication mechanism made up of two components : SecurityConsole and DeviceSecurity [7]. UPnP forum considers the device with SecurityConsole as a central device and other devices will consult with it to make a security decision [8]. Unfortunately, it is left to the developer of the devices whether to implement security for their devices or not. This has led to the deployment of many devices with non-existent or weak authentication [7]. We should suspect that this is particularly true for home routers.

Without authentication, UPnP devices will accept reconfiguration commands from any other device on the same network. Of particular concern are the possibility that an attacker could reconfigure the router to allow a man-in-the-middle attack where the attacker makes independent connections with the victims and will make them believe that they are talking directly to each other. The attacker will make them believe that they are communicating over a private connection while in fact the entire conversation is controlled by the attacker.

Such an attack can be implemented in two main ways by exploiting the ability to reconfigure the IGD. First, port mappings can be added that allow forwarding of requests between the victim's machine and the router to processes under the control of the attackers. Second, the Domain Name Server (DNS) settings on the IGD can be modified to point at a DNS server under the control of the attacker. This DNS server can be redirect requests for legitimate sites to fake sites under the control of the attacker.

III. Detecting UPnP Vulnerabilities

A number of tools exist for exploring UPnP enabled devices on a network and manipulating them. However, we have only identified one publicly available tool designed to identify vulnerable UPnP devices (UPnP Map) that has been used to do a survey of vulnerable systems.

Tools that allow exploration of UPnP devices on the network and interacting with them include: UPnP Gateway Traffic Monitor [9], HillSoft UPnP Explorer [10] and the Miranda UPnP Administration Tool [11]. Although these allow a skilled user to explore potential vulnerabilities none

of these tools could be termed vulnerability scanners because they do not diagnose specific vulnerabilities.

UPnP Map [12] is the only tool we have identified that can be used to diagnose specific vulnerabilities. UPnP Map focuses on port mapping vulnerabilities and is designed to look for UPnP vulnerabilities exposed to the Internet via the Wide Area Network interface of a IGD. A beta version of the tool is available that provides a command line interface. It attempts to detect if UPnP is enabled, if enabled it attempts to add and remove a port mapping. Being able to do this successfully marks the IGD as being vulnerable to a port mapping and therefore man-in-the-middle attack. In Garcia's presentation at DEFCON-19, he reported that he had carried out a one week scan of IP pools in 2011 revealed 150,000 vulnerable devices. This is only account of a survey of UPnP surveys that we have found.

Although identifying potential for external attacks using UPnP Map is useful, our focus is on identifying vulnerabilities from attacks originating from within the home network. None of the existing tools provide this capability. In addition, our goal is to develop a tool that can be used by home users to run their own analysis and contribute the results of their analysis to allow us to survey the prevalence of vulnerabilities. At the same time we wish to protect user privacy.

IV. Design and Implementation

As discussed, vulnerabilities of UPnP protocol make it easier for an attacker to abuse the whole network in a very simple way. Among all the vulnerabilities, our intitial focus is on identifying IGD that are vulnerable to port mapping attacks. In this Section we provide an overview of the system architecture, structure of the vulnerability scanner, and the implementation of the vulnerability scanner.

A. System Architecture

Figure 1 is the architecture for the tool which shows the server and two clients. In the diagram, there are two main features: Collection and Reporting. The process starts with the home user downloading the vulnerability scanner (step 1) from a trusted site on the Internet. The home user starts the vulnerability detector. The vulnerability detector attempts to discover the IGD on the network and determine if UPnP is enabled or not (step 2). Assuming that it is enabled it uses the UPnP commands to try to add and remove a port mapping. Being able to do this indicates that the IGD both has UPnP enabled and the particular implementation allows the adding and removal of port mappings thereby making man-in-the-middle attacks possible. The result of the vulnerability scan is reported to a trusted server (in our case a server in our lab) by connecting over port 6789 (step 3).

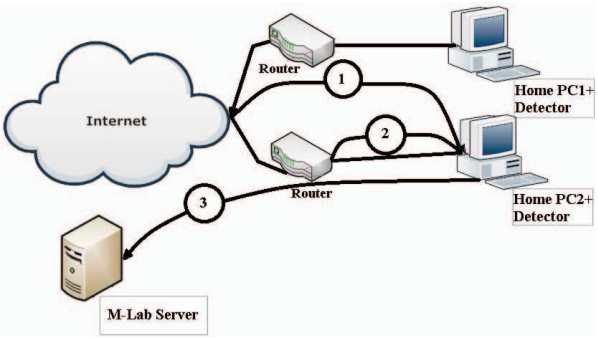

Figure 1. System architecture of vulnerability scanner.

B. Vulnerability Detector Protocol

The protocol used by the vulnerability detector is described in this section, we include how it performs the scans are reports results securely to a server collecting statistics on vulnerabilities. We call the vulnerability scanner the client (C), the IDG is referred to as a device (D) and the server collecting statistics is referred as a server (S). It is assumed that we have created two public-private key pairs. The first is used in the protocol below to protect communications between the client and the server and the second is assumed to a key pair used to encrypt the collected data on the server itself. We assume that the second key is longer than the first because the data is held longterm and therefore an attacker stealing the data would have longer to try and discover the necessary private key in order to reveal the collected information.

Note that in this protocol we do not protect against forgery because the server's public key is well-known so it would be trivial to forge details. However we do not believe that an attacker would gain from doing this instead an attacker has more to gain by finding out which home user's are vulnerable to attack.

A description of the protocol is given below. The interaction between the client and server is in the following sequence:

1. $C \rightarrow D : DiscoveryRequest$

The client sends multicasts Discover Request to all the devices on the LAN.

2. $D \rightarrow C : DiscoveryReply$

If there are any UPnP enabled devices in the LAN, they will send unicast packet back to the requested device.

3. $C \rightarrow D : DescriptionRequest$

Using the URL (Uniform Resource Locator) extracted

from Discovery Reply, a Description Request will be sent to the device.

4. $C \rightarrow D : AddPortMappingRequest$

5. $C \rightarrow D : DeletePortMappingRequest$

In steps 4 and 5, we attempt to modify the device by trying to add and delete PortMapping. The implemented tool first attempts to add portmapping. If adding portmapping is successful, the tool deletes it for the following two reasons. First, it should identify whether the device is vulnerable to deleting portmapping. Second, it tries to revert the changes back to the remote device in order to prevent any miss-settings on the client router. In case the vulnerabilities are found in the users' device, depending on the user decision, the collected information about user's vulnerable device will be reported back to the server.

Finally, is step 6 the results of the scan are reported back to a central server. This is encrypted using the server's public key (KS^{+}). Note that the details reported are abbreviated below and we include a nonce N to support detection of replays:

6. $C \rightarrow S : \{ServiceID_1, ...ServiceID_J, N\}_{KS^{+}}$;

C. Issues of Consent and Privacy

Under New Zealand law, the data gathered from the system is about information on users' systems which could be deemed to be personal data. Therefore, we should ensure full and informed consent and conform to guidelines on appropriate handling of data that respects the New Zealand Privacy Act[1] . We have included an information screen that details the purpose of collecting the data, how long it will be held and how it is protected.

D. Implementation Details

This section explains the implementation of the vulnerability scanner. There are two main components: engine and user interface. In the engine, the program discovers the devices as well as vulnerabilities to add/delete portmapping attacks. In the User Interface, the users use the engine to check the vulnerabilities of their devices.

1) Engine: Engine is responsible for the discovery of devices, retrieving vulnerable device services, attempting to add and delete port mapping, simulating the complete process of discovering the device, lunching the attack in order to check the vulnerabilities, and re-setting of the device to change it to the first step. For implementing the engine by these specific functions Cling [13] library has been used.

[1]Privacy act

2) User Interface: When the program starts, it shows information about the tool, what the tool is going to do, and what information it will collect and show. If the user agrees, he/ she will click the agree button. By clicking on the agree button, the software will automatically find the UPnP enabled devices and check the vulnerabilities by trying to add/delete the portmappings. All attempts have been made to make the tool easy to use for the users and prevent users from being confused. It gives appropriate feedback to the user in a simple and short sentences. It gives all the consents to the users of the tool to prevent ethical failure. When the report on the vulnerable device is ready, it will be given to the user. To inform user about his/ her vulnerable devices, three links will be introduced to the user. Users will get help by reading the suggested documents. If the users are satisfied with the consent and agree to send the data, then the user clicks on the send button and the information will be decrypted and sent to the trusted server.

An important aspect in user interface is content readability of the text. The content readability of the user interface was tested by Flesch-Kincaid Readability test [14]. Flesch-Kincaid examines the level of difficulty of the passage. The result from readability of the content calculation for the informative interfaces of the tool is 72.409. According to Flesch-Kincaid, this is a good result for the stages from an eleven year old student.

There is another test for the purpose of calculating the readability level of the passage. By conducting the Flesch-Kincaid grade level testing, the academic level of the passage can be achieved. The result achieved from grade level of testing of the content for the informative interfaces of the tool is 8.364. The result indicates that the text is understandable by an average student on the 8th grade. The 8th grade means ages around 12 to 14 in the United States of America. Both of the tests confirm that the interface is easily understandable for home users.

V. Testing of the Vulnerability Scanner

We setup a home network without our lab and tested a number of easily obtainable IGDs used within New Zealand. Our internal client machine was a laptop running Windows 7. The client machine was connected to the IGD and each IGD was connected to our School's network to simulate connection to the Internet. The result of our testing is shown in Table I. In the table we indicate the device and whether the adding and deleting of ports was possible. Note that we also carried out functional testing on a network with multiple UPnP devices that included printers as well as IGDs to test whether we could identify IGDs (we do this by blindly adding and removing ports).

Dynalink RTA1320V6 was tested by the program and it had the UPnP enabled feature. It allowed for adding and deleting portmappings. The problem with this device is that

Table I
TESTED WIRELESS GATEWAY DEVICES

Model	Device	Add portmapping	delete portmapping
Dynalink RTA1320V6	Wireless gateway/router	Yes	Yes
D-link DIR-600	Wireless gateway/router	Yes	Yes
D-link DI-704UP	Wireless gateway/router	No	No
D-link SL-504G	Wireless gateway/router	No	No
D-Link DSL-2730U	Wireless gateway/router	Yes	Yes
ZyXEL P-335WT	Wireless gateway/router	Yes	Yes

after each enabling or disabling the UPnP feature on device, the device had to be rebooted.

D-link DIR-600 was tested and had the UPnP enabled. It was vulnerable to adding and deleting portmapping.

D-link DI-704UP had the UPnP feature enabled but was not vulnerable. D-link SL-504G does not have a UPnP enabled feature. So it is not vulnerable to attack.

D-Link DSL-2730U was vulnerable to add/delete portmapping but the problem was that this device had to be connected to the Internet otherwise it could not be found as a vulnerable device. ZyXEL P-335WT has the option for users which allows them to make configuration changes through UPnP. If this option is checked by the user, then this device is vulnerable.

Overall we found that our tool was able to detect if a UPnP device was vulnerable and also that we were able to successfully report the results of the testing back to our central server.

VI. FUTURE WORK

Our current vulnerability scanner focuses on port mapping vulnerabilities. We would like to extend this to also detect possible vulnerabilities such as DNS reconfigurations. We believe that adding extra vulnerability scanning capabilities is important because it is not sufficient to simply determine if UPnP is active as different implementations may have different levels of UPnP support. Therefore having UPnP enabled on a given router might allow port remapping but not changes to the DNS server settings.

Also this is the first step in a larger study. We would like to carry out more user testing to ensure that it can be used by our target audience i.e. home users and actually deploy our tool within New Zealand and/or the United Kingdom.

VII. CONCLUSIONS

In this paper, we described a tool for detecting UPnP vulnerabilities and collecting statistical data as to the prevalence of the vulnerabilities. We described the implementation of the tool and outlined how we tested it within a controlled setting. Our next steps as identified in Future work are to refine the tool and deploy it to collect data on the prevalence of vulnerabilities.

REFERENCES

[1] D. C. Plummer, "IP network address translator (NAT) terminology and considerations, the internet engineering task force, rfc 2663," August 1999, http://tools.ietf.org/html/rfc2663.

[2] I. Al-Mejibli and M. Colley, "Evaluating UPnP service discovery protocols by using NS2 simulator," *2nd Computer Science and Electronic Engineering Conference (CEEC)*, pp. 1–5, 2010.

[3] B. Miller, T. Nixon, C.Tai, and M. Wood, "Home networking with universal plug and play," *Communications Magazine, IEEE*, vol. 39, pp. 104 –109, dec 2001.

[4] Symantec Corporation, "The Microsoft UPnP (universal plug and play) vulnerability," 2002, http://www.symantec.com/connect/articles/microsoft-upnp-universal-plug-and-play-vulnerability.

[5] M. Gudgin, H. Hadley, and N. Mendelsohn, "W3C technology, simple object access protocol, version 1.2," 2007, http://www.w3.org/TR/soap12-part1/.

[6] J. Cohen and S. Aggarwall, "General event notification architecture base: Client to arbiter," 2000, http://tools.ietf.org/html/draft-cohen-gena-client-00.

[7] UPnP Forum, "UPnP Forum homepage," 2012, http://www.upnp.org/. [Online]. Available: http://www.upnp.org/

[8] "Understanding universal plug and play," 2008, www.upnp.org/download/UPNP_understandingUPNP.doc.

[9] "Download UPnP gateway traffic monitor 1.0 - softpedia," http://www.softpedia.com/get/Network-Tools/Network-Monitoring/UPnP-Gateway-Traffic-Monitor.shtml.

[10] "Download HiliSoft UPnP browser 1.5," http://www.download3k.com/Network-tools/Network-monitoring/Download-HiliSoft-UPnP-Browser.html.

[11] "Miranda UPnP administration tool," http://www.sourcesec.com/2008/11/07/miranda-upnp-administration-tool/.

[12] D. Garcia, "Universal plug and play (UPnP) mapping attacks," DEFCON-19, 2011, http://toor.do/DEFCON-19-Garcia-UPnP-Mapping-WP.pdf.

[13] "Cling - Java/Android UPnP library and tools," 2012, http://4thline.org/projects/cling/.

[14] "The Flesch reading ease readability formula," 2012, http://www.readabilityformulas.com/flesch-reading-ease-readability-formula.php.