Poster: Exploiting UPnP Protocol for Botnet Propagation and Control

Di Wu[1,2], Binxing Fang[3,4], Xiang Cui[1,2], Chaoge Liu[1]
1 (Institute of Information Engineering, Chinese Academy of Sciences)
2 (School of Cyber Security, University of Chinese Academy of Sciences)
3 (Beijing University of Posts and Telecommunications)
4 (Institute of Electronic and Information Engineering in Dongguan UESTC)
wudi6@iie.ac.cn

Abstract—**With the development of Internet of Things (IoT), various devices connect to the Internet, which also bring us new security risks. To date, most research workers in the IoT security field focus on analyzing the weakness of devices from communication, configuration, backdoor and system vulnerability. However, with the increase of devices and protocol types, large-scale controlling is becoming more difficult. To change this situation, we studied the communication technology among devices and determined that the Universal Plug and Play (UPnP) protocol has the ability to identify IoT devices and distribute commands. Consequently, we propose an UPnP-based botnet, implementing bot propagation and control by exploiting the UPnP protocol. Moreover, we set up a re-infection mechanism to enhance the resilience. In general, the botnet, which has good accuracy in device discovery and status monitoring, is efficient and stable. The results of preliminary experiments indicate that our approach can be supported by the standardized parameters and protocol features of UPnP devices.**

I. Introduction

A botnet is a group of compromised computers that is remotely controlled by botmasters via command and control (C&C) channels. Traditional botnets are composed primarily of PCs and mobile phones, but in recent years, botnets are moving in the direction of device diversification. The hybrid botnet has gradually become a new trend and shown some new characteristics. However, research workers are paying less attention on it, and there are many issues on building and managing this type of botnet at present.

UPnP is a bunch of protocols designed to allow networked devices, such as routers, smart phones, printers, or home appliances to seamlessly find each other and establish connection, and has been widely used in IoT devices. It uses Simple Service Discovery Protocol (SSDP) to search devices, and controls devices by means of Simple Object Access Protocol (SOAP). In addition, UPnP comes with a solution named Internet Gateway Device (IGD) protocol for NAT traversal by mapping port in the gateway devices. But UPnP also exists some safety risks[1], for example, it makes internal IP addresses behind the NAT expose to the public network and can be accessed from external IP addresses. Besides, UPnP has serious flaws in authentication. If UPnP devices do not implement relevant protection measures, certain UPnP services can tamper the configurations such as port mapping and DNS setting without any permission.

Due to these features of UPnP, in this paper, we design a botnet based on it and make three contributions. First, we improve the accuracy and efficiency of target discovery. For the difficulty of probing and identifying devices in the IoT environment, bots use SSDP to obtain the detail description of targets in the propagation. Second, we enhance the resilience of botnet by mutual monitoring among bots. The state of UPnP devices can be subscribed with General Event Notification Architecture (GENA) protocol. Thus, we establish a re-infection mechanism through this portocol, and re-infect the target when it becomes invalid. Third, we propose a C&C channel from the botmaster to bots. The control method can be set up by utilizing the weakness of UPnP authentication. More specifically, bots will use SOAP to add a corresponding port mapping in gateway devices in order to receive the commands.

II. Design

In our work, we mainly discuss the propagation and control of botnet which based on different UPnP devices on the same LAN. Botmaster need to get some vulnerabilities of target devices in order to take control them, and compile relevant bot programs on different platforms. The process of botnet propagation and control (Figure 1) mainly consists of three parts: scan, infection, and control.

A. Scan ("①" and "②" in Figure 1)

Assuming that we have infected a device, in order to discover potential weak UPnP devices on the same LAN, the bot first sends several SSDP discovery requests with different "ST" field to "239.255.255.250:1900" by multicast. The "ST" field in request packet is used to refer the type of search target such as media renderer or printer. By setting this value, bot can scan devices in the specified range initially.

If the response is received, it means that there are interested targets alive. The "Location" field in response packet gives the URL of device description (i.e., 192.168.0.1:49152/des.xml), and bot will get the description file through it. The description is stored in XML format and usually includes more detail information about the device, such as device type, service type, uuid, manufacturer name, and model name. These parameters can help the bot to further determine whether the target is vulnerable. In addition, there are two key tags in the description file named “controlURL” and “eventSubURL”, which are used to control and subscribe device, respectively. We will discuss about their usage later.

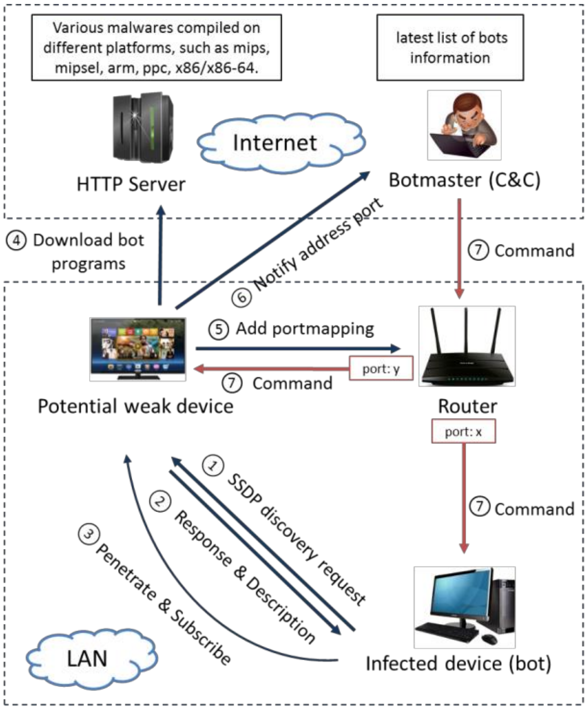

Figure 1. The process of botnet propagation and control

B. Infection (“③” and “④” in Figure 1)

Once bot has determined the target, the next step is trying to penetrate it with the included exploit code and password dictionary. In case of success, target will connect to the server of botmaster through the router. There are various bot programs and malwares compiled on different platforms in the server, and the target will reporting its own operating system and CPU architectures to download appropriate programs.

Beyond that, since the resilience of malware in most such devices is not strong, to solve this issue, we can choose a reliable device on the LAN like PC to be a “subscriber” and establish a re-infection mechanism. Each UPnP device has some state variables to show the running status, when they changed, device will multicast the events. Thus, we can choose a major event that is able to indicate whether the target is active or not as a reference, and let subscriber use GENA protocol to monitoring it through the “eventSubURL”. The target will be deemed invalid if the subscriber has not received the event signal for a long time, and then subscriber will re-infect it and notify the C&C of its state. Through the mechanism of mutual supervision, we can increase the overall survival on the LAN.

C. Control (“⑤”, “⑥” and “⑦” in Figure 1)

In order to take control of devices, we can let the botmaster to access to bots besides polling commands from the botmaster. Due to the weakness in authentication, the UPnP IGD profile usually allows any client on the LAN to tamper certain configurations of gateway devices. Therefore, infected device uses AddPortMapping SOAP action to request a Port Mapping to forward from the IGD WAN interface to bots, and then bot will notify the C&C server of its address port. If the LAN IP address of bot gets changed by DHCP, the bot should send again, which ensures C&C server maintains a latest list of bots information and status. In this way, botmaster can transmit commands to bots through specified ports on the router. Compared to the way of polling commands from the botmaster, this control method will be more flexible.

III. Preliminary Results

To prove the feasibility of the proposed botnet, we firstly investigated services and the corresponding state variables of certain kinds of UPnP devices. According to the official standardized documents, UPnP devices simultaneously support the required, optional, and non-standard (implemented by vendors) parameters. Therefore, we chose and verified some type of devices and their representative required properties for propagating and subscribing bots. Table 1 displays the examples that we specified for botnet propagation and control. For example, we can use “Printer” or “PrintBasic” combined with the information of printer model and manufacturer to recognize the target, and subscribe to the event about the variations of “JobID” to determine whether the bot is alive.

TABLE I. The Parameters of UPnP Devices

Device-Type	Service-Type	Variable Name
BinaryLight	SwitchPower	Target
MediaRenderer	RenderingControl	LastChange
Printer	PrintBasic	JobId

In addition, we realized major construct operations in embedded Linux devices by using lib files of miniupnp[2], such as sending control commands to the router and adding a designated port mapping. Experiments show that many routers exist such UPnP security problems, which proves that our approach is feasible in practice.

IV. Conclusion

On the basis of the observation that UPnP protocol has good controllability, flexibility, and application universality, in this paper, we study the major features of UPnP and design a botnet based on it. Our approach is able to discover and control UPnP bots accurately by taking advantage of the communicating mechanism and weakness of UPnP protocol.

Acknowledgment

The authors would like to thank the anonymous reviewers for their helpful comments for improving this paper. This work is supported by the Industry-University-Research Cooperation Project of Guangdong Province (No. 2016B090921001).

References

[1] Squire J. Universal Plug and Play IGD-A Fox in the Hen House (white paper), 2008.

[2] MiniUPnP project. https://github.com/miniupnp/miniupnp, 2014.