Universal Plug and Play Eventing Vulnerabilities

Joeri Blokhuis

February 4, 2009

Joeri Blokhuis Universal Plug and Play Eventing Vulnerabilities

イロト イヨト イヨト イヨト

Research question

Introduction

Eventing

Research

Conclusion

< □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □



- What vulnerabilities can be found in UPnP Eventing?
 - Can a Callback URL contain any URL and thereby attack a website everytime a state variable changes?
 - Can the subscribers list be filled in such a way that it can't accept any new subscriptions?
- Testing is done on two devices using Intel's UPnP library

イロト イポト イヨト イヨト



- UPnP Forum formed in 1999
- > 800 members
- Various industries
 - Computers, networking, consumer electronics, mobile products
- Define UPnP standards

イロン イヨン イヨン イヨン

UPnP

- Simplify home networking
 - Auto-configuration of devices
 - No user interaction
- Using existing standards:
 - TCP, IP, UDP, HTTP, SOAP and XML
- Simple architecture can be defined by:
 - Devices, services and control points

伺 ト イヨト イヨト



UPnP has several phases:

- Addressing
- Description
- Control
- Eventing
- Presentation

イロト イヨト イヨト イヨト



- Keep state of variables
- Will notify registered entries when a variable changes
- Publisher/subscriber model
 - Publisher (service)
 - Subscriber (control point)
 - General Event Notification Architecture (GENA)

(本間) (本語) (本語)



- Uses HTTP as transport
- Three new HTTP methods
 - SUBSCRIBE
 - UNSUBSCRIBE
 - NOTIFY



・ロト ・回ト ・ヨト ・ヨト

Э

Callback URLs

A subscription request has the following format SUBSCRIBE /upnp/control/WANIPConn1 HTTP/1.1 HOST: 192.168.2.1:52869 CALLBACK: <192.168.2.100/test> NT: upnp:event TIMEOUT: Second-1800 A successful subscription will respond with HTTP/1.1 200 OK DATE: Sat, 01 Jan 2000 22:30:45 GMT SERVER: Linux/2.4.18-MIPS-01.00, UPnP/1.0, Intel SDK for UPnP devices /1.2 SID: uuid:16766c80-1dd2-11b2-a2ce-e7182fbea8a1 Timeout: Second-1800 ・ 同 ト ・ ヨ ト ・ ヨ ト



Any URL is accepted as a callback

- Any IP address (not LAN only)
- No domains
- Same callback URL can be registered
- Bad submitted URLs respond with
 - 501 Method Not Implemented

伺 ト イヨト イヨト



- Service won't delete subscriptions
- Contradicts with Intel's specifications
 - "Avoid unnecessary consuming of resources"

同 と く き と く き と

UUID

- Universally Unique IDentifier
- Generated on every successful accepted subscription
- Only known to
 - Publisher
 - Subscriber
 - Control point
- UUIDs can be sniffed to
 - cancel a subscription

回 と く ヨ と く ヨ と



A cancellation message has the following format
UNSUBSCRIBE: /upnp/control/WANIPConn1 HTTP/1.1
HOST: 192.168.2.1:52869
SID: uuid:16766c80-1dd2-11b2-a2ce-e7182fbea8a1

Denial of service

- Both devices were prone to a denial of service
- Caused by sending to much subscriptions
- Submitting subscriptions with an infinite timeout
- All services(UPnP) will stop working
 - no event notifications
 - no responses to discovery messages
 - no control

- 4 同 6 4 日 6 4 日 6

Denial of service[2]

- Done by creating a while-loop
- Sleep function to slow down subscriptions
 - Edimax: 18000 subscriptions and 42 minutes
 - Sitecom: 14000 subscriptions and 1,5 hour

伺 ト イヨト イヨト

Denial of service[3]

- Devices use a maximum number of subscriptions as resources allow
 - this is set when UPnP is enabled
- Suspected that
 - more resources are being used due to handling the load of subscriptions
 - maximum number will than be lower
 - causing a DoS
- Explains why the number of subscriptions aren't consistent to cause a DoS

イロト イポト イヨト イヨト



Intel's UPnP Eventing stack is found to be reasonably solid

- Any Callback URL is possible
- UUIDs used for communication
- Denial of Service

イロン イヨン イヨン イヨン



Questions

Joeri Blokhuis Universal Plug and Play Eventing Vulnerabilities

・ロト ・回ト ・ヨト ・ヨト