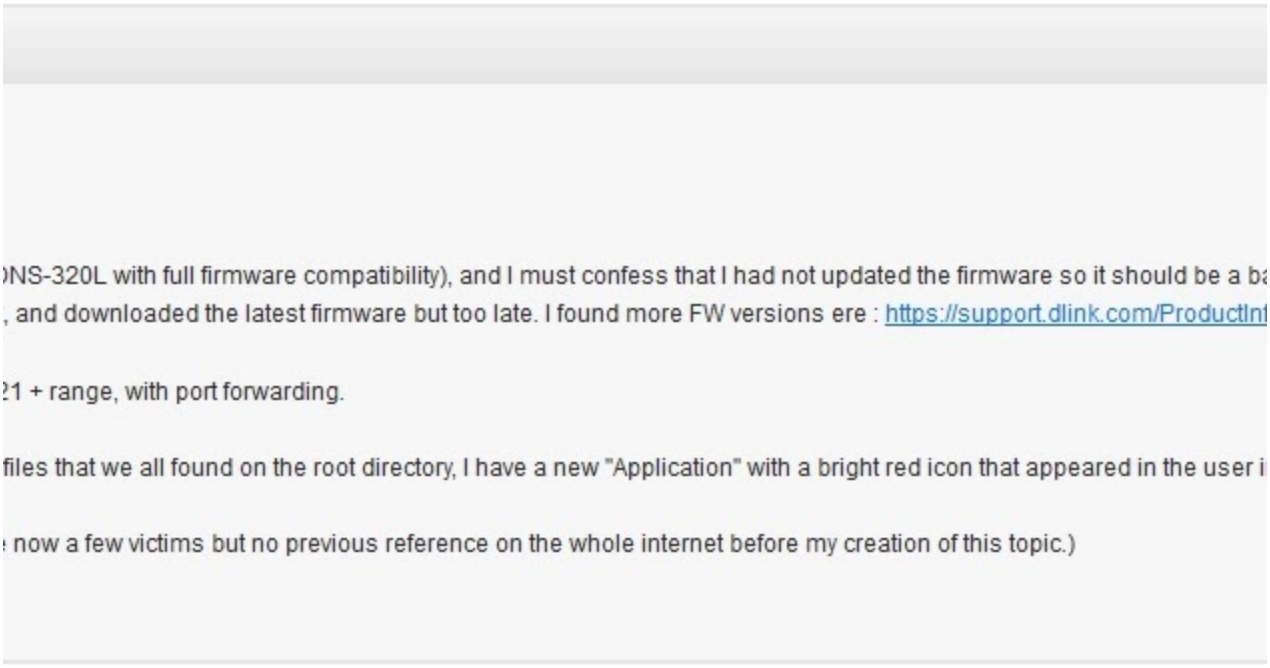
D-Link DNS-320 NAS Cr1ptT0r Ransomware ARM Dynamic Analysis - QEMU and Raspberry PI VM

resolverblog.blogspot.com/2019/02/d-link-dns-320-nas-cr1ptt0r-ransomware.html



Hi Everybody,
a few days ago I saw a tweet from @Amigo_A_ asking for help about a new ransomware which was affecting a D-Link 320 NAS.
The first thought was directed to the historical disabling of dlink to make sufficiently secure firmware and their willingness not to support updates. Those facts made me to think about an attack conducted over the net targetting all the devices exposed on internet itself.

Apparently was the right hypothesis.
All the users with D-Link 320XX are nowdays are at very high risk.
TURN OFF THE DEVICE AND DISCONNECT IT FROM WAN. On BleepingComputer's forum I asked to the affected users to check their own firmwares and trying to grab the malware. Someone did and shared the ELF on VirusTotal.
Thanks to Michael Gillespie @demonslay335 I was able to have a copy of that sample.
Hash: 9a1de00dbc07271a27cb4806937802007ae5a59433ca858d52678930253f42c1

(very few) years ago I had experience on some router exploiting and reversing (Italian ISP company named Telecom Italia and their ADSL routers), they were based on MIPS with a very good OS (Jungo OpenRG) always trivial to exploit. But this is another story, I'lve spent a lot of time on those devices learning some useful stuffs which today apparently become a good knowlege.

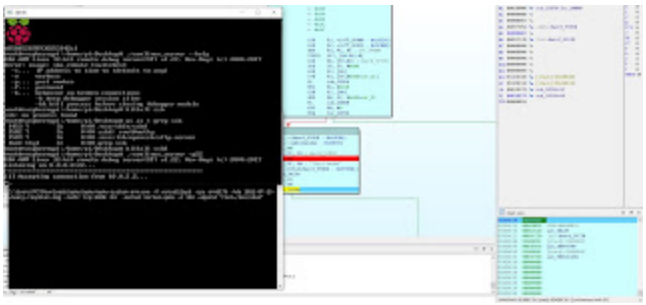
Since the fact that this ransomware is stripped (with removed debugging informations!) and statically compiled, the static analysis is very hard to do since the fact that any calls appear to be just a sub_XXXXX because of the stripped ELF.
Because of this, we have few options to make our life less complicated:
1) do a dynamic analysis
2) create IDA pro FLIRT signatures

Starting from the first point we faced a new problem: where to run the ARM malware?
2 opportunities: the first on a QEMU VM, the second on a D-Link device (of course lol).
I do not buy D-Link stuffs, so I had only one opportunity: QEMU.
Since I'm lazy (and the executable is statically linked), I decided to try with my Kali x64 VM and qemu-static-arm with -g parameter which enable the gdb debugger.
I really dont know why, but something go wrong bringing ida pro to crash. LOL 😔

It was a fail, so I've started to look at a new easy path and I thought about raspberry pi.
On Sourceforge there is a nice prebuild, preconfigured QEMU Raspberry emulator:
https://sourceforge.net/projects/rpiqemuwindows/
On the run.bat file I've added a parameter to be able in order to upload with FileZilla over sftp protocol the malware and the remote debug server, and then killing the sshd, I've used the same port to connect with IDA. Smart lazyness😀

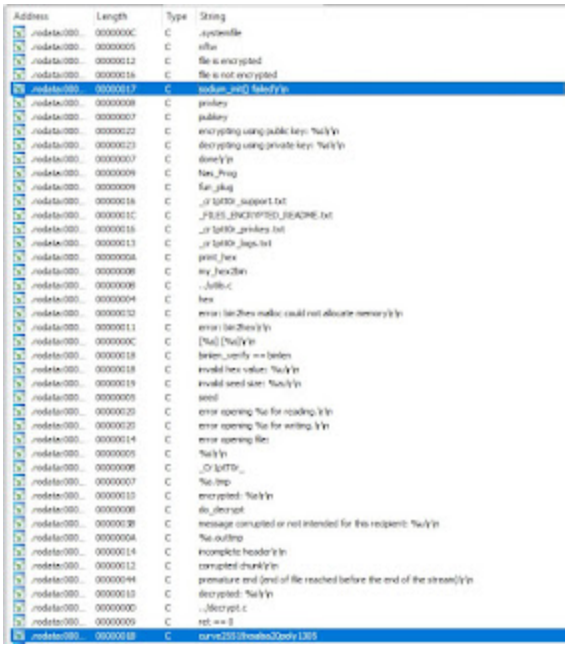
qemu-system-arm.exe -M versatilepb -cpu arm1176 -hda 2012-07-15-wheezy-raspbian.img **-redir tcp:2200::22** -kernel kernel-qemu -m 192 -append "root=/dev/sda2"

Starting again the IDA Pro remote debugging, the following stuffs comes up!!!!! YEAH IT WORKS!!!! 😍😍



So far we know few things about such ransomware which are:

in addition, the strings shows up that he also use crypto routines from Libsodium library https://libsodium.gitbook.io/doc/



Like I said before, the - hardest - next step is to create a IDA FLIRT signature, by cross compiling some example from Libsodium repo (hoping that it will use the same functions as the malware), extract the signatures by using FireEye idb2pat tool https://www.fireeye.com/blog/threat-research/2015/01/flare_ida_pro_script.html to have an understandable static analysis to MAYBE retrive the private key and decrypt the files, or at least have a reduced keyspace to make possible a brute force attack.

Follow me on Twitter and I'll keep you updated.
Cheers
RE Solver