

Security of WPA2 PSK with the Elcomsoft Wireless Security Auditor

Author: Kristian Burfeindt
ID: 881203N170
Course: Wireless Communication Systems
Supervisor: Urban Bilstrub
2nd period, WS 2010/11

Table of Contents

1. Preface	1
2. Theory	1
3. Praxis	3
4. Analysis of the results	6
5. References	6

Table of Figures

FIGURE 3.1: RESULT OF AN ATTACK.....	3
FIGURE 3.2: TEST RESULTS	4
FIGURE 3.3: FULL MUTATION RESULT.....	5
FIGURE 3.4: CALCULATED TIMES FOR BRUTE FORCE.....	5

1. Preface

The Elcomsoft Wireless security auditor is claimed being a program to be able to crack WPA2 PSK secured wireless Networks. It is widely known this was possible before with intelligent dictionary- or brute force attacks. The problem is that these attacks work, but also need much performance and time. So one would need cloud computing which provides extreme performance or much time to calculate. What the Wireless Security Auditor does is using Graphics Cards for calculation. The idea behind this is to take advantage of the parallel processing in Graphics Cards.

The aim of this project is to find out whether this extra performance is that high that even private persons can easily crack WPA2-PSK keys with this tool or not. There are several tests with different hardware starting from mobile AMD Athlon MV-40 processors up to High-end Desktop Graphics cards to see how this tool scales.

2. Theory

The function of the Elcomsoft Wireless Security Auditor is quite simple. It tries to sniff a hash and afterwards to find out the password of the hash with a dictionary- or brute force attack. In words this means that it simply checks every single word whether it fits to the hash or not. These words are saved into *.dic files and can easily be exchanged. An English dictionary is included in the software which contains 242.964 words. If one needs to do a brute force attack only the dictionary has to be exchanged because the workflow of both attacks is the same, only the “words” are different. Another specialty of the program is being able to “mutate” the words of the given dictionary. In the Elcomsoft documentation [1] the mutation options are described as following:

All mutations are divided into a several 'classes' (described below). The program can set the mutation 'level' for every type, that allows selecting between speed and efficiency. With the minimum level, the program checks only lowercase passwords, and performs basic mutations only: e.g. Border mutation uses not all special characters, but only digits, and only at the end of the password. For an intermediate level, more special characters are being used (both as prefix and as a suffix); and uppercase characters are also tested. At the maximum level, even more advanced prefixes and suffixes are added, but of course, it runs much slower (as far as more variations are checked).

- *Case mutation: the program checks all variations of uppercase/lowercase characters.*
- *Digit mutation: adding several digits to the work (from the dictionary) as prefix and suffix.*
- *Border mutation: similar to the above, but adding not only digits, but also most commonly used combinations like 123, \$\$\$, 666, qwerty, 007, xxx etc.; in addition, adding some chars at both end of the word, e.g. #password#, \$password\$ and more.*
- *Freak mutation: replacing some characters (one or more); for example, the word password will also generate p@ssword, p@\$s\$word and p@\$s\$w0rd.*
- *Abbreviation mutation: some commonly-used abbreviations like ihateyou - ih8you, loveyou - loveu, foryou - 4u etc.*
- *Order mutation: reversing the order (password - drowssap), repeating the word (password - passwordpassword), adding the reversed word (password - passworddrowssap).*
- *Vowels mutation: playing with vowels, e.g. psswrđ, PaSSWoRD, pAsswOrd etc.*
- *Strip mutation: removing one char, e.g. assword, pssword, pasword...*
- *Swap mutation: replacing some characters, e.g. apssword, psasword, password...*
- *Duplicate mutation: duplicating the characters, e.g. ppassword, paassword, passsword, passwword etc.*

- *Delimiter mutation: adding delimiters between characters: p.a.s.s.w.o.r.d, p+a+s+s+w+o+r+d, p-a-s-s-w-o-r-d.*
- *Year mutation: adding the year (four digits) at the end of the word: password1973, password2002.*

The workflow of the program is very simple: The password of the Wireless network is contained in every successful handshake which means that only one handshake needs to be sniffed by the program. The program itself can only sniff networks with AirPCap adapters. Because these are very special it is also possible to import hashes from other Tools like airodump-ng, OmniPeek, CommView for Wi-Fi, Wireless Snif, AirMagnet, AirDefense, Sniffer Global [1 S. Capturing network packets]. After the hash is recognized in the Wireless Security Auditor it can attack like described before.

Another feature is that the program is able to produce hashes itself. This means that the user enters the SSID and password of a network manually and a hash for a successful handshake for this network is created. This hash doesn't differ to sniffed hashes (for the same network and password), because the program is simply going the way vice versa. Since the project is not hinted to really attack or check networks but to find out how much faster the Security Auditor works by using GPU computing it wouldn't make sense to install extra software for packet sniffing. This needs to be done with normal dictionary- or brute force attacks too. Another advantage of this feature is that the passwords and SSIDs can be changed very quickly without having to sniff new hashes every single time a new test is done. Adding these advantages there is no reason to use extra software for sniffing in this project leading to use this feature.

Another advantage of this attack is that it is completely invisible for the network. Most of the time is taken by trying to crack the hash locally.

3. Praxis

The program is only available for Testing as version 3.1.0.69 standard version. This means that only two CPU cores can be used and only one GPU. There is also a free Demo version from Elcomsoft that allows using the most up-to-date GPUs but also only 2 CPU cores. The mutation is also disabled in the demo version. To stay comparable between the versions a simple password without need of mutation was chosen. "password" is a good choice because it is quite at the end of a dictionary and not very hard to crack.

A hash is created under 'File -> Add WPA/PSK Hash Manually'. One only has to enter the SSID and the password and the program creates the hash. Under Options the mutations are adjusted and also the CPU Cores and Graphics Cards which shall be used can be chosen. Finally the attack is started and while it is running performance data is shown.

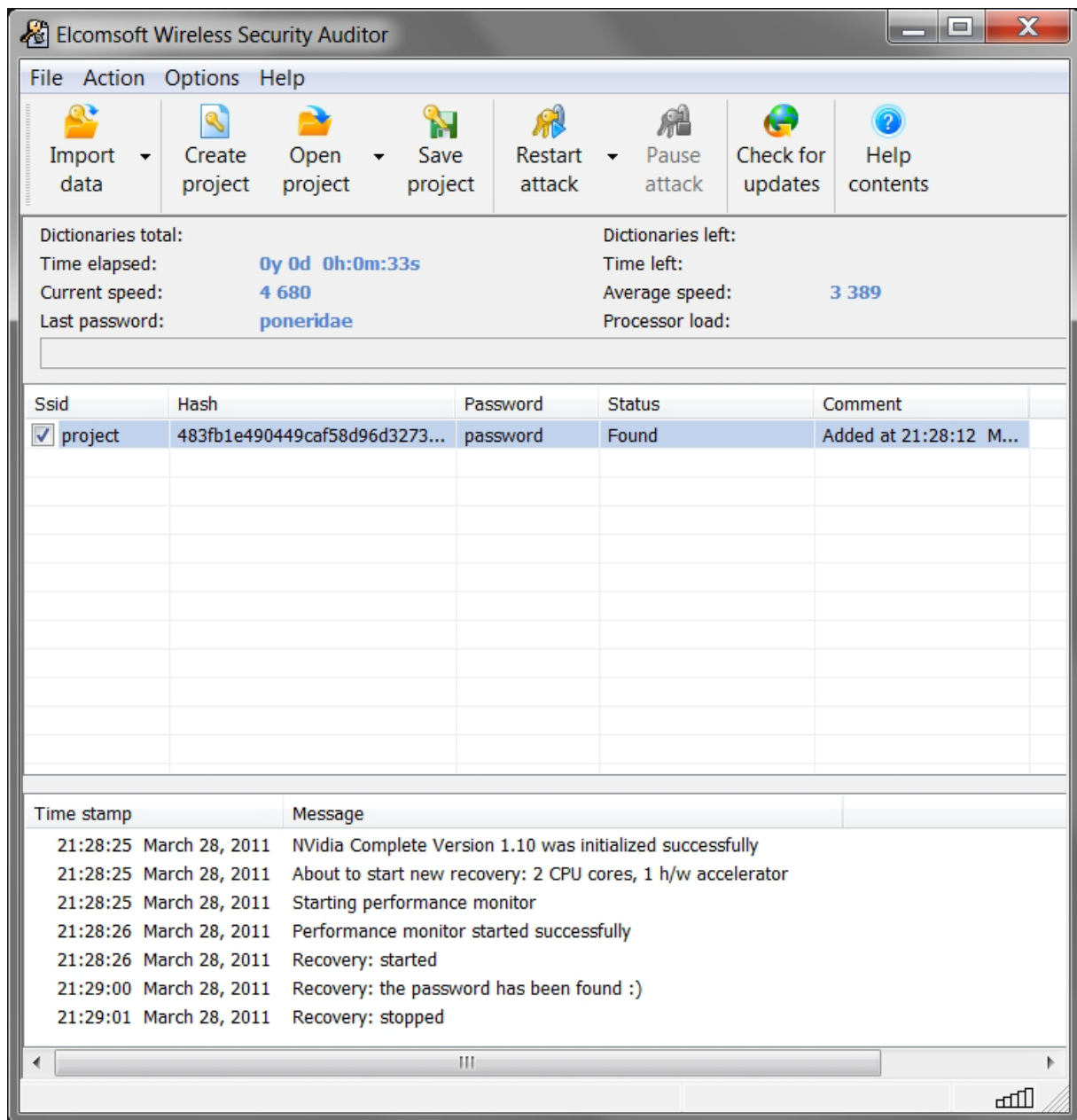


Figure 3.1: Result of an attack

shows the result of an attack where the password “password” was found after 33 seconds. This time was achieved without mutation and a mobile i5-520m plus a NVIDIA GT330M. For later states on scaling of this software a wider test was done with different computers and the same options from Figure 3.1. To acquire more test results the computerbase.de community was asked [2] (source in German) to help out to do the same test with the demo edition of the Elcomsoft Wireless Security Auditor. The results are merged in Figure 3.2.

Because some of the CPUs used where overclocked it is necessary to give the clock rates of most of the processors. The 3DMARK06 was chosen as a reference value for later conclusions on scaling.

CPU	Graphics Card	3DMark06	Average speed (s)	Time (hh:mm:ss)
Intel i5-2300@3,4GHz		5892	2400	00:00:42
Intel i5-2300@3,4GHz	Radeon HD6970	23622	68000	00:00:02
Intel i7-860@2,6GHz		5048	2076	00:00:50
Intel i7-860@2,6GHz	NVIDIA GTX460	18534	19562	00:00:06
Intel i5-760@3,15GHz			2107	00:00:48
Intel i5-760@3,15GHz	Radeon HD5770		22220	00:00:04
Intel i7-620m@3,06GHz		3214	1500	00:01:06
Intel i7-620m@3,06GHz	NVIDIA NVS3100	3316	2412	00:00:44
Intel i5-520m@2,65GHz		2752	1524	00:01:04
Intel i5-520m@2,65GHz	NVIDIA GT330M	6862	3469	00:00:33
AMD Phenom II 955 BE@3,6GHz		4870	1953	00:00:52
AMD Phenom II 955 BE@3,6GHz	HD4850	18494	22171	00:00:06
AMD MV-40		604	163	00:10:40

Figure 3.2: Test Results

For testing purpose a dictionary attack with full mutation was done on this device with a cryptic password like “csda!ho32059vsvo,jh2z09”. This of course cannot be cracked with a dictionary attack so the result is the full time an attack would take if it is needed to run through the whole dictionary. Figure 3.3 is the result and means that it would take about 12:48 hours.

If one would now like being able to crack any password, brute force attacks are needed. In the following brute force is explained. First of all one needs to know that simply mutating an easy dictionary like the english.dic helps with easy mutations like editing just some letters or putting some words of the dictionary after each other. This is not enough for cryptic passwords so that a dictionary with any possible combination is needed. Mutation is not used in this case. WPA-PSK supports 8 to 63 printable ASCII characters [3]. This means that there are 95 characters available for every single character in the WPA-PSK.

$$Combinations = 95^{Number\ of\ characters\ used} - 95^7$$

$$Needed\ Time\ (s) = \frac{Combinations}{Words\ per\ Second}$$

If one would use every single of these 63 characters the possible combinations would be $3,95 \cdot 10^{124}$. To prove this we assume that there is only the english.dic from the dictionary attack in Figure 3.2 available with only 242.964 words. The calculated time for the system from Figure 3.1 is ~70 seconds to run through the whole dictionary. This is almost exactly the

time which is needed in reality keeping in mind that the password 'password' is found before the whole dictionary is run through.

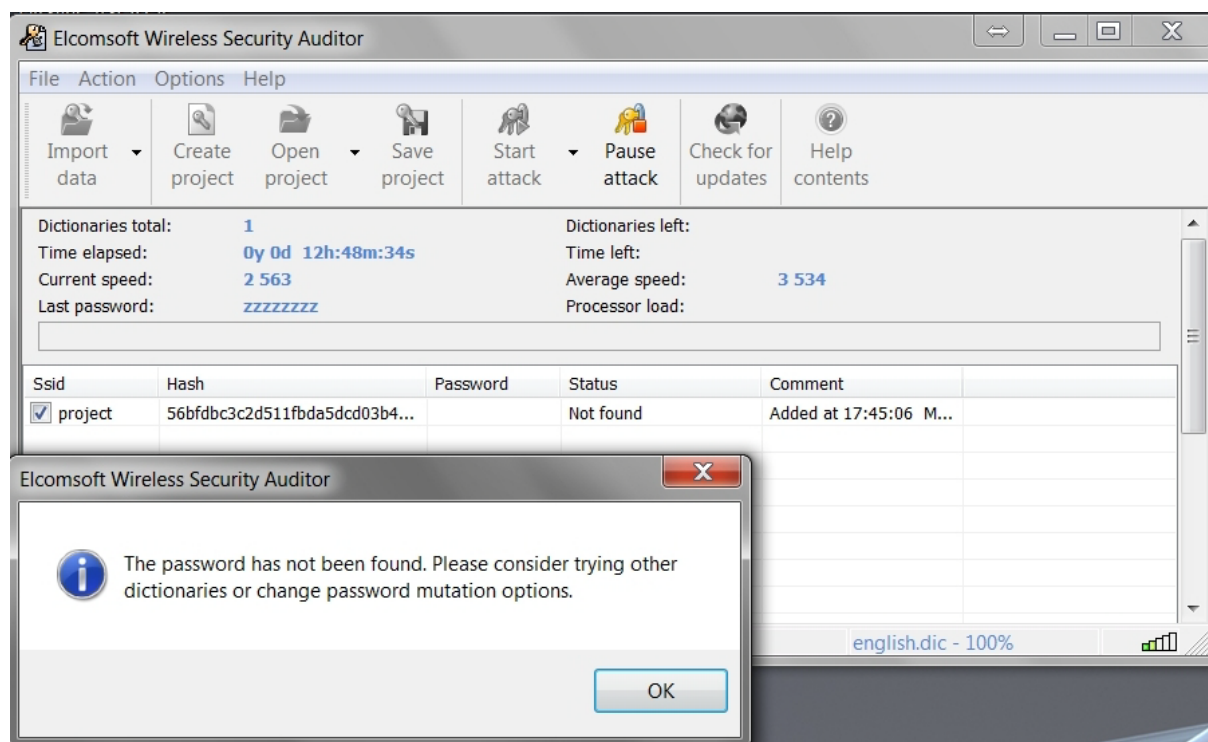


Figure 3.3: Full mutation result

After proving that the calculated time is mostly the same as the real time it is easy to give statements on the time which is needed for a brute force attack that is able to find any password.

Characters used	Possible combinations	Average Speed (s)	Calculated time			
			Seconds	Hours	Days	Years
8	6,56E+15	3.469	1,89E+12	5,26E+08	2,19E+07	6,00E+04
16	4,40E+31	3.469	1,27E+28	3,52E+24	1,47E+23	4,02E+20
63	3,95E+124	3.469	1,14E+121	3,16E+117	1,32E+116	3,61E+113
8	6,56E+15	68.000	9,65E+10	2,68E+07	1,12E+06	3,06E+03
16	4,40E+31	68.000	6,47E+26	1,80E+23	7,49E+21	2,05E+19
63	3,95E+124	68.000	5,81E+119	1,61E+116	6,72E+114	1,84E+112
8	6,56E+15	68.000.000.000	9,65E+04	2,68E+01	1,12E+00	3,06E-03
16	4,40E+31	68.000.000.000	6,47E+20	1,80E+17	7,49E+15	2,05E+13
63	3,95E+124	68.000.000.000	5,81E+113	1,61E+110	6,72E+108	1,84E+106

Figure 3.4: Calculated times for brute force

4. Analysis of the results

The results give two outcomes of this project. On one hand if a WLAN is secured with one or several words from a dictionary that are only little mutated it is very easy to find out the password with an attack like the one in this project. Even small laptops can do this in little time. Talking about performance there is no really good indicator to make a good prognosis of the average speed of the Elcomsoft Wireless Security Auditor. While the CPUs scale pretty good compared to their performance in 3DMARK06 graphic cards are extremely dependent on the number of Stream processors. Especially AMD Graphics Cards with their high number of stream processors take advantage in GPU computing. This is easily shown with two cards. The AMD HD5770 with 800 stream processors is normally about ~20% slower than a NVIDIA GTX460 with 336 stream processors [4] but in this program it is ~12% faster (Figure 3.2). This causes that one cannot really say how a special Graphics Card would scale in this program just that in general more stream processors are better than high clock rate.

On the other hand the results of brute force attacks indicate that there is no chance to crack a cryptographic password with this program. The fastest computer in this project was a high end gaming machine with an AMD Radeon HD6970 (Figure 3.2) and even this would take more than 3000 years for an 8 character long cryptographic password.

One can say that GPU computing is a 'nice to have' for faster testing of networks whether they are too easy to crack but really secure WPA-PSK can only be cracked with cloud computing. As a simulation for cloud computing a value 1.000.000 times faster than the fastest computer in this project was chosen. With this scenario any password with 8 characters can be cracked within 1 day (Figure 3.4).

As end of this project one can say that there is no chance to crack passwords with 16 arbitrary chosen characters within a human lifetime at this point.

5. References

- [1] **Co.Ltd., ElcomSoft.** EWSA Help >> Elcomsoft Wireless Security Auditor. [Online] 2010. <http://www.elcomsoft.co.uk/help/ewsa/index.html>.
- [2] **computerbase.de.** *Projekt: WPA2-PSK Sicherheit und GPU Computing.* [Online] 15.03.2011. <http://www.computerbase.de/forum/showthread.php?t=872873&highlight=elcomsoft>.
- [3] **Wikipedia.** <http://en.wikipedia.org/wiki/WPA-PSK>. [Online] 12.03.2011.
- [4] **Andermahr, Wolfgang.** Test: Nvidia GeForce GTX 460. *computerbase.de.* [Online] 12.07.2010. http://www.computerbase.de/artikel/grafikkarten/2010/test-nvidia-geforce-gtx-460/20/#abschnitt_performancerating_ohne_aaaf.