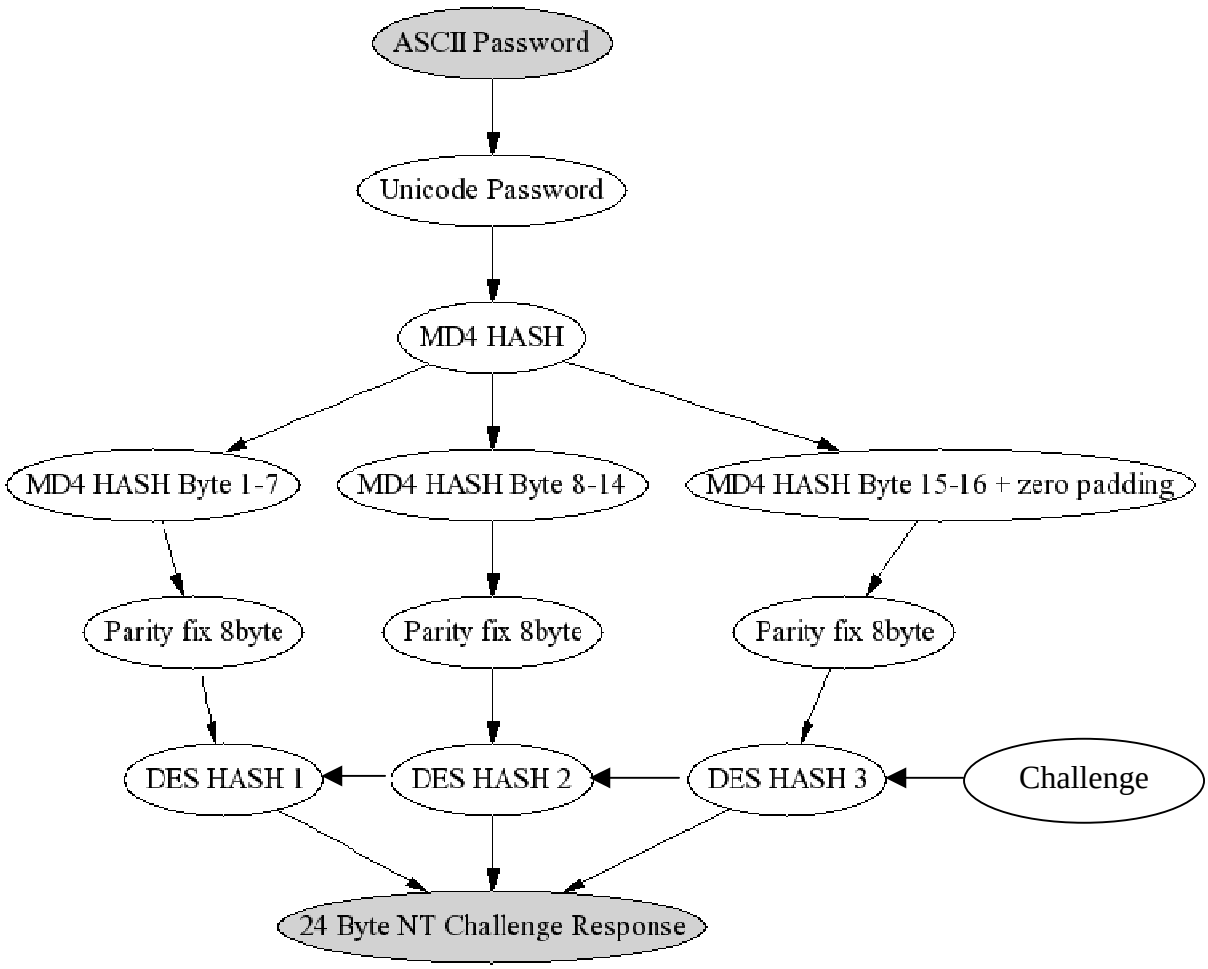
LEAP Attack tools:

Intro:

I hate writing or reading docs, so I tried to keep it short and clear, hopefully not too short.

LEAP-Cracker Tool:

The Cisco LEAP protocol use the NTChallengeResponse authentication method. Unfortunately, this Authentication is vulnerable against offline brute force attacks. The NTChallengeResponse authentication first translates the ASCII password into Unicode, than it generates an MD4 Hash out of the given Unicode password and splits this hash value into three parts. First part: byte 1-7, second part: 8-14 and last part: byte 15 and 16 zero padded to 7 bytes length. These three parts of the MD4 hash value are getting parity fixed and are used as keys to DES encrypt the random challenge (that was chosen and transmitted by the other site of the communication, see below 'A general description of the LEAP'). The three 8 byte results are merged into the 24 byte NTChallengeResponse Hash that is sent in clear text (see Pic.1).



Pic. 1

A general description of the LEAP protocol (taken from the Free Radius Mailing list):

AP = AccessPoint and RS = RadiusServer, these messages are more or less forwarded to the wireless client. The important part is that it's similar to a CHAP authentication. The servers sends a random challenge to the client and the client uses this random value to encrypt the password (only steps 1 - 3 are used by our tool). Actually, this is the reason why the victim is disconnected from the network for a while – until the timeout is over and the authentication process is re-initiated by the client (the client is using the wrong challenge (our spoofed challenge) for the rest of the authentication process ☺.)

1. AP->RS: Radius Request/EAP Identity, containing the name of the user to be authenticated

2. RS->AP: Radius Challenge/EAP Request/LEAP,containing a 8 octet random MSCHAP Peer Challenge (PC)

3. AP->RS: Radius request/EAP Response/LEAP, containing the 24 octet MSCHAP response(NtChallengeResponse) to the challenge in 2 above (PR).

4. RS->AP: Radius Access-Accept/EAP Success

5. AP->RS: Radius Request/EAP Request/LEAP, containing 8 octet Access Point Challenge (APC).

6. RS->AP: Radius Access-Accept/EAP Response/LEAP, containing 24 octet response to the challenge in 5 above (APR), plus a session key sent in a cisco-avpair vendor-specific attribute.

The first weakness the tool uses, is that the input of DES Hash 3 has only a range of 2^{16} possible input values. So it brute force crack the last two characters of the MD4 Hash. After this is done, we are starting to generate MD4 Hashes out of the given ASCII password input (depending on the option of the tool you have chosen - password generator or wordlist). Too save some processing time, it compares the last two bytes of the generated hash with the cracked last two bytes of the sniffed NTChallengeResponse MD4Hash. If they are matching it generates the DES HASH 1 and compares it, if it is also matching it compares the DES HASH 2. If both are matching we found the original password or at least one that results in the same MD4Hash (birthday vulnerability of MD4). The most time consuming part is, of course, the generating of all the MD4 hashes. That's why the MD4 function is speed optimised – we are using a 'short cut version' of the original MD4 padding routine, unfortunately this only works for passwords up to 16 characters length. The Unicode translation is also not a real Unicode translation, we are just inserting 00 between the characters, but both optimisations should work for most cases.

Password cracking options of leap cracker tool:

Password Generator: you can use the [–l] and [–a] (or [-w]) function to generate passwords. For example, " -l 3 –a abcdef " would generate a password with 3 characters length and all possible combinations out of 'abcdef'. The way in which order the passwords are generated, depends on the order of your input. The algorithm is a number system algorithm, so your alphabet characters are the number system members :

-a 01 -l 3 means the passwords are generated like this: 000,001,010,011,100,101,...
-a abc -l 3 means the passwords are generated like this: aaa,aab,aac,aba,abb,abc,...
-a cba -l 3 means the passwords are generated like this: ccc,ccb,cca,cbc,cbb,cba,...

The –w is the same option like –a, the only difference is that it accepts the following wildcards:

a-z	=	abcdefghijklmnopqrstuvwxyz
A-Z	=	ABCDEFGHIJKLMNOPQRSTUVWXYZ
0-9	=	0123456789

Additional you can use a prefix for the password generator with the option '-p', e.g. "–p cisco" –l 3 –w 0-9" would generate passwords from "cisco000 – cisco999".

Wordlist: you can use an ASCII password wordlist instead of the password generator. The option would be '-f <filename>' and the format of this wordlist is :

Password1
Password2
Password3
…

You can use the 'wordgen' tool to generate wordlists. See wordgen –h for details.

Brute force attack*:* the third option is to use a pre-compiled binary 'password/DESHash'-file. You can use the 'passwords_convert2bin' tool to compile this file out of a ASCII password file and a given challenge. So you have a binary file with the NTChallengeResponse and the corresponding password. Now, the tool don't have to generate the Hashes anymore, it just has to compare the given NTChallengeResponse and the values in the file. This is, of course, 'slightly' faster than the other options. The option is '-b <bin-filename>'.

NTChallengeResponse - Input options of leap cracker tool:

Manual input. You can use one NTChallengeResponse Hash as input (-t) and optional the corresponding Challenge (-c) (default Challenge is 'deaddeaddeaddead') for one single user and one of the three password cracking methods from above.

or ...

Userlist input. You should use an ASCII user list file with the format:

USERNAME CHALLENGE NTCHALLENGERESPONSE.
USERNAME CHALLENGE NTCHALLENGERESPONSE
USERNAME CHALLENGE NTCHALLENGERESPONSE
...

This is the same file format that the getleap tool uses for the output file.

GET LEAP Tool:

The basic idea of the getleap tool is to spoof the LEAP response from the AP that includes the random challenge value, so the victim is using our given challenge to calculate the NTChallengeResponse.
The advantage of this is that you have the same challenge for all users and you are able to use the 'leapcracker'- tool with the bruteforce and userlist option. That means you can use one pre-compiled 'password/DESHash'-file for all of them. I guess the chance to find at least one user with a weak password is quite high.

The getleap tool can either sniff for users on the WLAN or you can give it a MAC address (-m) of a special user you want to attack. The only option you have to set is the MAC address of the AP in your WLAN. See getleap –h for more details.

Depending on the state in which the users wireless card is in the moment, it's behaviour could be an other than the tool expects, but you can restart or abort the sniffing process at any time with CTRL-C. Due to an bug in the Airjack driver (see below) , it's also possible that you have to use the reset function (r) . See the online help (-h) for more details.

The getleap tool needs the AirJack driver by Abaddon. This is a special driver for PrismII wireless cards (and others) that gives you direct access to the Layer 2, so you are able to generate every wireless packet you like . The tool was tested with version 0.6.6b of AirJack and the old version of the SMC 2632W wireless card. (Hint: this version of AirJack doesn't work with Cisco cards !). For more details please see :

http://802.11ninja.net/airjack/ or http://sourceforge.net/projects/airjack/

Other useful links:

Linux Wireless cards:
http://www.linux-wlan.org/docs/wlan_adapters.html

Kismet link list:
http://www.kismetwireless.net/links.shtml

Other Tools and Scripts:

deauth-all	- send deauthentication frames to all or one special user
passwords_convert2bin	- convert ascii password file to binary list (see above)
wordgen	- ASCII wordlist generator (see above)
resetcard.sh	- reset your Wireless Card
td_showbeacons.sh	- shows all beacons in tcpdump

Conclusion:

The main security issue is not the LEAP algorithm, but the strength of the user password, if this password is a good strong password, you will have a hard time to crack this password.

Recommendation:

Please see Cisco: *Dictionary Attack on Cisco LEAP* paper:
http://www.cisco.com/en/US/tech/tk722/tk809/technologies_security_notice09186a00801aa80f.html

Author:

DeX7er / THC (dexter@thc.org)
www.thc.org