Global Information Assurance Certification Paper

- Do not fall ASLEAP -

GIAC Certified Incident Handler (GCIH)
SANS Practical Assignment Version 3
By Claudio Silotto

Submitted on July 19th 2004

Abstract

This paper provides information about offline dictionary attacks against the Cisco LEAP protocol, as well as an incident handling process based on the SANS 6 step method.

Cisco LEAP (Lightweight Extensible Authentication Protocol) is an authentication protocol supported by a large number of wireless devices, including Cisco Aironet Series and Cisco Compatible Extensions equipments.

This subject is relevant due to the release of the tool called "asleap"[1] by Joshua Wright on April 6, 2004 that simplifies the process of capturing and discovering passwords. The tool is able to perform millions of password checks per second on commonly available hardware.

[1] Wright, Joshua. "asleap home page". URL: http://asleap.sourceforge.net

INDEX

1. Statement of Purpose

In this paper I will be using a fictitious scenario created specifically to illustrate the possible weaknesses of a Cisco LEAP implementation. All the tests were performed on a dedicated lab without any private information on it.

The corporate network of a new but fast growing consulting company will be the scenario of this paper. This company will be referred to from this point on, as "Fictcius Co." or simply "Fictcius".

At the moment, Fictcius Co. has only one site where about one hundred employees work. Some of them are of course performing back office activities, others management, marketing, human resource tasks and so on, but about fifty of them are delivering consulting services to the clients. Fictcius has also some technical people responsible for maintaining their web site as well as their IT infrastructure. Recently, a few positions were made available to expand the local help desk and support team responsible for the first contact with employees and customers but they have been only partially filled.

An important information about this company, and relevant to this scenario, is that due to the success and fast growth of the business the infrastructure started to become an issue and changes were needed to address points like shared cubicles and mobility to the consultants who needed access to the network from different areas within the company. So it was decided that an 802.11b wireless Access Point would be used.

Because the technical people had already heard about the innumerous cases of misuse of the wireless network by "hackers", some security features were implemented. Cisco LEAP was chosen as the authentication method, dynamic WEP keys were used and the SSID broadcast was disabled (cloaking the network).

Using this technology and features Fictcius Co. believed that they were safe and had now a solution for most of their current problems.

That is when the incident took place.

It will be demonstrated here how a person may discover an authentication pair (username/password) simply because he/she can listen to (sniff) the wireless traffic with the right software.

In this case, asleap is the "right software".

Asleap is the main tool presented in this paper and it is well detailed on the next section (Section 2) but, to complement, some directions on other utilities

normally used to analyze wireless networks are also included here, mainly to update the reader with the latest features available by the time of this writing.

In the last, but far from least, section of this document (Section 5) the handling process of an incident of this kind will be made, following SANS' six phase method and applying it to this fictitious scenario.

2. The Exploit

The exploit covered in this paper is based on the fact that Cisco LEAP, like any other password-based authentication protocol, is vulnerable to brute force attacks.

This becomes an issue because of the simplicity of wireless LAN sniffing techniques and the availability of tools capable of performing millions of tries per second on common hardware to offline crack these passwords in seconds.

2.1. Name

The exploit in this case was referred by SANS as "Cisco LEAP Brute Force Password Cracker" on one issue[2] of the @Risk newsletter.

Bugtraq created the "ID 8755"[3] for this exploit and some good information can be found in there.

No CVE entry has been created so far for this issue but US-CERT did publish a "Vulnerability Note"[4] to cover this attack.

2.2. Operating System

LEAP is a Cisco proprietary authentication method and the exploit is based on the expected behavior of the protocol. Because of that, the vulnerable operating system is any Cisco Systems equipment with an implementation of Cisco LEAP. Normally it will be a Cisco Aironet Access Point running Cisco IOS or VxWorks software and a few clients running some wireless configuration tool on Windows or even Linux.

2.3. Protocol

Also called EAP-Cisco Wireless, LEAP is a Cisco authentication type that uses usernames and passwords to perform authentications between wireless

[2] SANS Institute. "@Risk: The Consensus Security Vulnerability Alert" April 19, 2004 Vol. 3. Week 15. URL: http://www.sans.org/newsletters/risk/vol3_15.php

[3] Security Focus. "Vulnerabilities BID#8755". URL: http://www.securityfocus.com/bid/8755

[4] US-CERT. "Vulnerability Note VU#473108". URL: http://www.kb.cert.org/vuls/id/473108

client devices and an authentication server. LEAP has also the ability to generate dynamic, per user and per section WEP keys.

LEAP authentication relies on a “shared secret”, the user password; and to prevent from sending it in clear text over the network, LEAP uses two values: a challenge value and a response value.

The challenge is created by the authentication server and sent in clear text over the network.

The response is first created by the client by a one way mathematical function, called “hash function”, of the password which is then used as the key of a DES encryption of the received challenge string.

Once the authentication server receives the response sent by the client, it repeats the same steps (made by the client) to create its own response (using the stored user password) to verify if the responses match and therefore authenticating the user.

The user password is stored in the form of a Windows NT key, which is a Message Digest Algorithm 4 hash (MD4) of an MD4 hash.

Since LEAP encryption mechanism was based on the Microsoft Challenge Handshake Authentication Protocol (MS-CHAP) it has basically the same vulnerabilities.

Cisco presented the following LEAP challenge-response protocol in its paper called “Cisco Response to Dictionary Attacks on Cisco LEAP[5]” providing some extra details about the process:

1. The wireless client sends an authentication request.
2. The access point relays the authentication request to the RADIUS server.
3. The RADIUS server acknowledges the client's request and sends an 8-byte challenge.
4. The wireless client computes a response:
a. The password is hashed with MD4. A 16-byte hash is produced.
b. The hash is padded with 5 nulls. 21 bytes are produced.
c. The resulting 21 bytes are split into 7-byte units.
5. DES encrypts the challenge as plain-text with the 7-byte unit as the key.
6. The resulting cipher text is concatenated producing 24 bytes.
7. The resulting 24 bytes are sent from the wireless client to the server as the challenge-response.

[5] Cisco Systems, Inc. “Cisco Response to Dictionary Attacks on Cisco LEAP”. URL: http://www.cisco.com/warp/public/cc/pd/witc/ao350ap/prodlit/2331_pp.pdf

8. The RADIUS server MD4 hashes the stored NT hash for the user (user's password is MD4 hashed twice) and repeats steps 4-7 to calculate the expected client response.

9. If the client and RADIUS responses match authentication is complete.

Note that on step 4, five null bytes are concatenated to the MD4 hash output. This will have an impact on the strength of the encryption and will be used by our exploit (more information can be found on Section 2.5 of this paper).

Like other EAP authentication algorithms, LEAP was designed to run on top of the 802.1X authentication framework, it is EAP type 0x11, but apparently it does not conform to the IEEE 802.1X specification and follows a Cisco implementation of 802.1X where Access Points may change packets in transit. However, we are not getting into the details in here.

2.4. Variants

We cannot find many Cisco LEAP exploits but we can find other tools that, like asleap, exploit the vulnerabilities presented here. THC-LEAPcracker[6] is one of them.

Asleap is so far the most complete tool for LEAP exploits so what we may see is different tools but not different methods or exploits.

Also, a large number of brute force attacks exist today and all those strategies they use may be implemented or added to tools like asleap to improve the exploit somehow.

2.5. Description

As said before, password-based authentication mechanisms rely on shared secrets, in our case on the users' passwords. If a password is compromised anyone with the credentials is able to authenticate successfully and obtain access to the resources.

So, it is possible to exploit this weakness and "guess" the password, in other words, brute force the password. For that, there are two options, online and offline attacks.

Normally the number of possible combinations (possible passwords) is not small so online attacks are not viable because they would eventually disturb the normal state of the network, raise the attention of system administrators, block the users after invalid tries and even be considered a slow process.

[6] Hacker's Choice, The. "THC-LEAPcracker". URL: http://www.thc.org/releases.php

On the other hand, offline attacks may be totally transparent and fast. Because it is easy to capture traffic on wireless network such attacks become a reality.

As previously said, Joshua Wright wrote a proof-of-concept program called asleap and this will be the tool used in this paper.

This tool is able to perform an offline dictionary attack to brute force the password and to use some weaknesses of the protocol to improve performance thus making the recover of weak passwords trivial.

Version 1.0, available by the time of this writing, runs on Linux with all features and on Windows with restricted functionalities.

The tool basically works on two phases. On the first one, it helps the attacker to capture the challenge/response messages exchanged between a valid wireless client and the access point (and the authentication server). Since asleap is capable of reading files on different formats this step can be performed by other tools, like Airopeek[7] or Kismet[8] (libpcap file).

On the second phase, the actual brute force attack happens. The tool searches through the captured packets looking for a valid and successful LEAP authentication sequence and then it starts the dictionary attack.

In order to recover the password, asleap will look for a match between the captured challenge/response and an entry found on the pre-computed dictionary file.

This pre-computed dictionary file can be created by a small program that is distributed with asleap called “genkeys”. “Genkeys” takes a plain-text file with a word per line and creates two output files. The first one is a new dictionary file with NT hashes of each word and the second one is an index-file for the NT hashes.

We are able to create and use these pre-computed files because of a design flaw of the authentication protocol where no salt is used to create the NT hashes.

Another flaw of MS-CHAPv2, which LEAP is based on, allows a great reduction of the number of possible character combinations. In fact this is what makes the difference and makes the offline process viable and so fast.

This flaw occurs because of the use of “null” bytes as part of the seeds to the DES algorithm generating cryptographically week results. MS-CHAPv2 splits the

[7] WildPackets Inc. “Airopeek NX”. URL: http://www.wildpackets.com/products/airopeek_nx
[8] Kismet Wireless. “Kismet”. URL: http://www.kismetwireless.net/

NT hash output of the user password, a 16 byte string, in three parts to use them as seeds to DES, but since each seed needs to be 7 bytes long, 5 “null” bytes are concatenated to the last seed.

Even with these flaws, everything still relies on the dictionary file and it may be possible to increase even more the chances of matching a password if a “good dictionary” is used. A “good dictionary” may be a more complete one or may be a targeted one, created specifically to your scenario.

2.6. Signature

It is really hard to identify this kind of attack before it is too late.

Since all a person needs to “crack” a password is to listen to the wireless traffic and then do an offline brute force attack, the only time that an organization will be able to identify that some credential has been compromised is when the attacker decides to use the username and password to connect (associate) to the network.

At that point some of the signatures may be:

- Identification of a new device association;
- Simultaneous use of an account.

But this is not an easy scenario to observe and it would require a well structured organization from a security point of view (with good infrastructure, trained personnel, policies, etc).

It is more likely to identify the intruder later on because of one of their actions after the credential has been used. Such actions may be:

- Unusual behavior;
- Out of business-hours access;
- Another signature triggered by some other action/attack.

Besides that, there is one option available on asleap that does not make this exploit totally unnoticeable and it is used to speed up the capturing of challenge/ response packets. By using that option asleap will send deauthentication packets forcing LEAP users to reauthenticate, which will allow the tool to capture the authentication sequence. This feature is better explained on section 4.3 of this paper. Even though it may not be considered a signature of this exploit because many other tools use the same strategy of deauthenticating users, it provides a way of detecting unfriendly activity on the network and may trigger alerts.

3. Platforms and environments

3.1. Victims

In this case, the victims are wireless users authenticating to the network by using valid LEAP credentials. They are vulnerable at the login time, when the LEAP challenge/response takes place.

To simulate the connections of our fictitious company users, I used the client software called "Odyssey for HP[9]" version 1.10 from Funk Software running on Microsoft Windows XP Professional.

Also, some tests were performed using two other client programs: a "Cisco Aironet Client Utility[10]" v. 6.3.011 running on Microsoft Windows XP Professional and a LEAP client running on a HP iPaq 4150[11] on Microsoft Pocket PC version 4.20.1081.

These two last programs seem to use some kind of LEAP authentication sequence variation in a way that asleap cannot automatically find the challenge/response, at least not in asleap version 1.0. For the exploit to work with them, some changes are necessary on the source code (probably at the findleapexch() function of asleap.c) but that is not covered in this paper. Consider the default client software as being the Odyssey v. 1.10.

3.2. Source

The source device used for this attack is a standard Intel notebook running Linux (Fedora Core 1) on one partition of the hard drive and Windows XP Professional on another. The PCMCIA 802.11b wireless card used is an Orinoco Gold card.

The main software used is, as previously commented, asleap.

There is no secret to compile asleap. The only change I wanted to make was on the "Makefile" file to assure it was compiled with the –D_LINUX flag. The lines below were copied from the "Makefile" after my change and show how it should look. After that, simply type "make".

```
...cut...
CFLAGS        = -g3 -ggdb -pipe -Wall -D_LINUX
#CFLAGS       = -g3 -ggdb -pipe –Wall
...cut...
```

[9] Funk Software, Inc. "Odyssey Client for HP". URL: http://h18007.www1.hp.com/support/files/EvoNotebook/us/download/17817.html

[10] Cisco Systems, Inc. "Cisco Aironet Client Utility". URL: http://www.cisco.com/public/sw-center/sw-wireless.shtml

[11] Hewllet-Packard. "HP iPaq 4150". URL: http://www.hp.com/go/ipaq

Of course, before compiling asleap it makes sense to have the wireless card correctly installed. In order to do so, many installation guides are available on the Internet and for sure they will provide the requirements for asleap, such as "libpcap".

To make the scanning of networks and packet capture more realistic, Kismet will be briefly used in this paper.

Since Kismet is not the focus tool of this paper (it is only used to illustrate and complete the attack scenario), I will not get into the details of the installation. If you need more information, please check the excellent work done by Ritchie on his How-to[12] paper covering the installation and configuration process including all needed drivers and patches.

The exploit program will also need a dictionary file. A good start to create one would be using the list available at rootexploit.net[13]. This is actually one of the resources I used to come up with my test files for this paper. Commercial versions of dictionary files are also available but are not covered in here and as for any brute force test, a more sophisticated and well targeted dictionary should be used on real assessments.

Another option is to create our own dictionary file, including every combination of characters we wish. I used a merge of files created by the "Brute Force Dictionary Maker"[14] tool v.1.0.

This merged file has 62,193,780 "words" using up to 5 characters from "a" to "z" and "0" to "9" and even though it is about 1.3Gb, asleap can test all these words in less than a second on my Pentium-M notebook. This is what makes this brute force attack so scary; time is not really an issue and all you need is a good (read complete) dictionary.

3.3. Target network

The wireless network of our fictitious company is composed by a Cisco Aironet 1100 Series[15] with Cisco IOS version 12.2(15)JA. This is the core of our target network.

[12] Ritchie. "Red Hat Linux 9.0 + Kismet HOWTO". URL: http://www.tipsybottle.com/technology/wireless/RedHat8-Kismet-HOWTO.shtml
[13] rootexploit.net. "Dictionary list". URL: http://rootexploit.net/docs/dictionaries/
[14] Astalavista Group. "Brute Force Dictionary Maker". URL: http://www.astalavista.com/index.php?section=dir&cmd=file&id=1622
[15] Cisco Systems, Inc. "Cisco Aironet 1100 Series". URL: http://www.cisco.com/en/US/products/hw/wireless/ps4570/index.html

A RADIUS server is running locally on the Access Point to authenticate LEAP users. This is not a requirement for our test but it is presented like this in order to simplify the network design.

The table below shows a few usernames and passwords that I created to simulate Fictcius Co. user's accounts.

Table 1 – Fictcius Co. Usernames and Passwords.

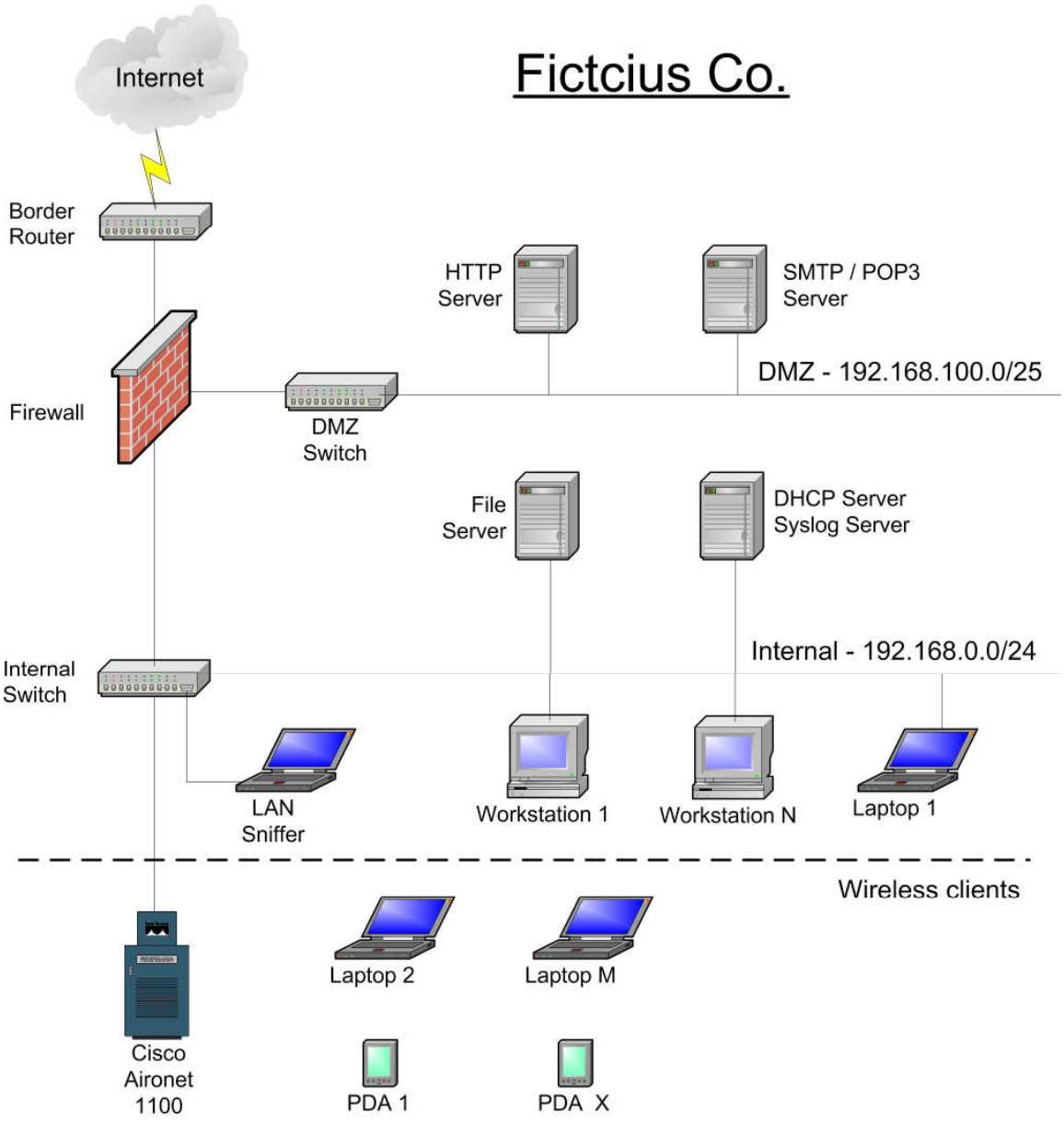
Account #	Username	Password*
1	james	j4mes
2	mary	love
3	david	IdkW2PtT!
4	susan	scott

* These passwords are only examples and should not be used on real systems.

Some extra hosts are active on the internal network as well and they are shown on the next diagram. More information on each of them will be provided when needed during the next sections of this paper.

3.4. Diagram

A logical representation of the network design used on Fictcius Co is presented below.



4. Stages of the attack

4.1. Reconnaissance

This is how the attack begins and it is a very important stage since some sensitive information may be gathered here which can strongly impact the overall success of the attack.

The necessary reconnaissance for this wireless exploit may differ a little from what is presented by SANS on track 4 seminar, but of course, the idea behind those actions is still the same.

Things like whois-results analysis and some web-based tools may not be really useful in this case since there is no need to find IP addresses or open ports on the external hosts of a company. But it is not hard to find useful information for our needs. One example would be finding out the address of one company that is posting messages on a group regarding LEAP configuration problems. All we would have to do is go to their web site and get the address of one or more sites and then we would be able to go there and start the wireless capture; that is it, you found your real target.

This stage does not have to be made as the first step of the attack or to finish soon either. It can be easily performed within the next steps, for instance, in the Scanning phase. Let's say that the attackers do not have a target yet for their next activity. All they know is that they need to find a wireless network in the neighborhood with LEAP authentication to test the new tool. They can drive around the city scanning for wireless network (war driving) and then choose the one they like the most. After they find out where a vulnerable network is located, they can start the reconnaissance phase, now with a defined target.

Some information that may be really relevant for our exploit and that can be obtained in this stage, is regarding users' password. If we are able to access or create users' profiles we can use more accurate dictionary files based on their habits, likes and personal information and that will definitely impact the success of the exploit.

Also, since this exploit is only focused on getting access to the network, any extra information that is found about the network may be useful after the access is granted. If you already know the internal network structure it may be easier for you to keep your access, cover your tracks or even perform the actions for which you needed the network access in the first place. For that matter, things like a non-split DNS architecture may disclosure some critical information.

For this exploit it would make sense to check the area for places to stay and sniff the wireless traffic as well as to check for the proximity of the building to

public areas, parking places and maybe even for a coffee shop next to the target company where it would be possible to comfortably sit down while waiting for the desired packets. These, among many others, are the things covered on this stage of the attack that may increase the chances of success.

4.2. Scanning

Wireless network scanning has been already covered in many papers and it was for some time considered a hot topic by the media (TV, newspaper, web sites…). It is not hard to find information about it and it is not a new topic, so I will be very brief about it.

If we want to scan for wireless networks we will eventually end up with one of these two softwares: Netstumbler[16] or Kismet.

Netstumbler version 0.4 was recently released with some good new features but even though it is easier to install and use I will stay with Kismet.

I am going to use Kismet in this paper because it is probably what is going to happen in real situations since it is not really convenient to use asleap to search for wireless network (even though it is possible).

Kismet is able to write all the information it captures to a file that can be imported later on into asleap. So, with Kismet, we are able not only to find wireless network but also to record their packets. There is only one thing missing which is to identify the networks that in fact use LEAP.

At this time, there is no flag or column on Kismet that shows whether some kind of EAP authentication is used on the network. The next version may contain one.

So, one way of doing this identification is opening the captured file (Kismet .dump) on a protocol analyzer looking for EAP packets (LEAP is type 0x11).

Now that the attacker knows which network is the target, all he/she needs to do is to find a nice place where he/she can have a good reception of the wireless traffic and sniff the packets for some time with any tool (like asleap, Kismet or even commercial ones like Aeropeek) until some challenge/responses are acquired.

One important detail here is that we should avoid hopping channels at this stage in order to increase the probability of capturing a full LEAP authentication sequence (use channel hopping only before, to find the wireless network, once found, lock on that channel to capture full negotiations).

[16] Netstumbler.Com. "Netstumbler v0.4". URL: http://www.netstumbler.com/

As said before this stage may also be performed by asleap by reading packets directly from an interface (option “-i”). This functionality will be discussed in the next section.

4.3. Exploiting the systems

After succeeding on the scanning of the network (or even on the capturing of a valid challenge/response authentication), it is possible to move on with the attack and actually crack the passwords.

Before running asleap the dictionary file needs to be put in a specific format.

We cannot simply feed asleap with a “word-per-line” source file. We need NT hashes of these words and for that we use the tool called “genkeys”. Some comments about this tool were presented in section 2.5 of this document, but this is how it should be executed:

```
[root@reality asleap-1.0]# ./genkeys
genkeys 1.0 - generates lookup file for asleap. <jwright@hasborg.com>

usage: ./genkeys wordlist outfile.dat indexfile.idx

e.g. "./genkeys words words.dat words.idx"
```

A real example, using the “allwords.txt” file from rootexploit.net, would be:

(Input file: “allwords.txt” and output files: “allwords.dat” and “allwords.idx”)

```
[root@reality asleap-1.0]# ./genkeys allwords.txt allwords.dat allwords.idx
genkeys 1.0 - generates lookup file for asleap. <jwright@hasborg.com>
Generating hashes for passwords (this may take some time) ...Done.
53082 hashes written in 0.22 seconds:  245350.17 hashes/second
Starting sort (be patient) ...Done.
Completed sort in 343012 compares.
Creating index file (almost finished) ...Done.
```

Now, having all the needed files, we can move on and recover the passwords. The following options may be passed to the main program:

```
[root@reality asleap-1.0]# ./asleap
asleap 1.0 - actively recover LEAP passwords. <jwright@hasborg.com>
asleap: Must supply an interface with -i, or a stored file with -r
Usage: asleap [options]

        -i      Interface to capture on
        -f      Dictionary file with NT hashes
        -n      Index file for NT hashes
        -r      Read from a libpcap file
        -w      Write the LEAP exchange to a libpcap file
        -a      Perform an active attack (faster, requires AirJack drivers)
        -c      Specify a channel (defaults to current)
        -o      Perform channel hopping
        -t      Specify a timeout watching for LEAP exchange (default 5 seconds)
        -h      Output this help information and exit
        -v      Print verbose information (more -v for more verbosity)
        -V      Print program version and exit
```

First I am going to use the “-r” option to make asleap read a “Kismet .dump file” containing only one successful authentication on it. The dictionary file in this case will be the “allwords” files created a while ago.

```
[root@reality asleap-1.0]# ./asleap -r Kismet1.dump -f allwords.dat -n allwords.idx
asleap 1.0 - actively recover LEAP passwords. <jwright@hasborg.com>
Using the passive attack method.

Captured LEAP exchange information:
        username:   mary
        challenge:  34e6c904f58ab4c9
        response:   ca2c03f92bbe212ce2729f1b95e9f6670ebde30a76749048
        hash bytes: 0716
        NT hash:    85deeec2d12f917783b689ae94990716
        password:   love
```

Since the user’s password was contained in the dictionary file it was recovered by the program.

If you want to capture traffic directly with asleap, use the flag “-i” which allows asleap to capture the authentication packets from an interface and crack the password in “real time”. By using this flag no Kismet dump is needed.

I have included not only the asleap command line but also some extra ones that are required for the tool to run well in this mode (see the execution below).

The first line simply shows the moment when the card is inserted and the module driver is loaded. The second command puts the wireless card on monitor mode and the third line brings the interface to an “up” state. Note that these commands may be different from system to system; so please adjust the commands to provide similar functionality according to your system.

Finally, the fourth command line runs asleap, and once again it is very straight forward.

```
[root@reality asleap-1.0]# cardctl insert
[root@reality asleap-1.0]# iwpriv eth1 monitor 2 1
[root@reality asleap-1.0]# ifconfig eth1 1.1.1.1
[root@reality asleap-1.0]# ./asleap -i eth1 -f a-z_0-9_one_to_five.dat -n a-z_0-
9_one_to_five.idx
asleap 1.0 - actively recover LEAP passwords. <jwright@hasborg.com>
Using the passive attack method.

Captured LEAP exchange information:
        username:   james
        challenge:  f44089e70e5855db
        response:   3ebeb2bcbd4fa0ad604bccee8ff73e30cb5280f4ee656249
        hash bytes: 8d9a
        NT hash:    2559aba735d6d14f917df937d2428d9a
        password:   j4mes

Captured LEAP exchange information:
        username:   susan
        challenge:  c4a4463afe866ac1
        response:   f42d30516b69d00d64a6bfd7cb967720e3582cfadadf5e46
        hash bytes: 8cbc
        NT hash:    0661a2cfddabc279bd51a10e961e8cbc
        password:   scott
```

```
Closing pcap ...
```

The example above was given using the dictionary file I wrote about in section 3.2 with more than 62 million words.

Simultaneous packet capture and password crack using large dictionary files is possible for asleap because of the flaws of the protocol and the use of an index file.

To complete the scenario, the output below shows asleap reading a Kismet dump with four successful LEAP authentications on it (Kismet4.dump).

Each of these authentications belongs to one user of the fictitious company presented before in section 3.3.

Once again the dictionary password is the “a” to “z”, “0” to “9” and up to five characters file.

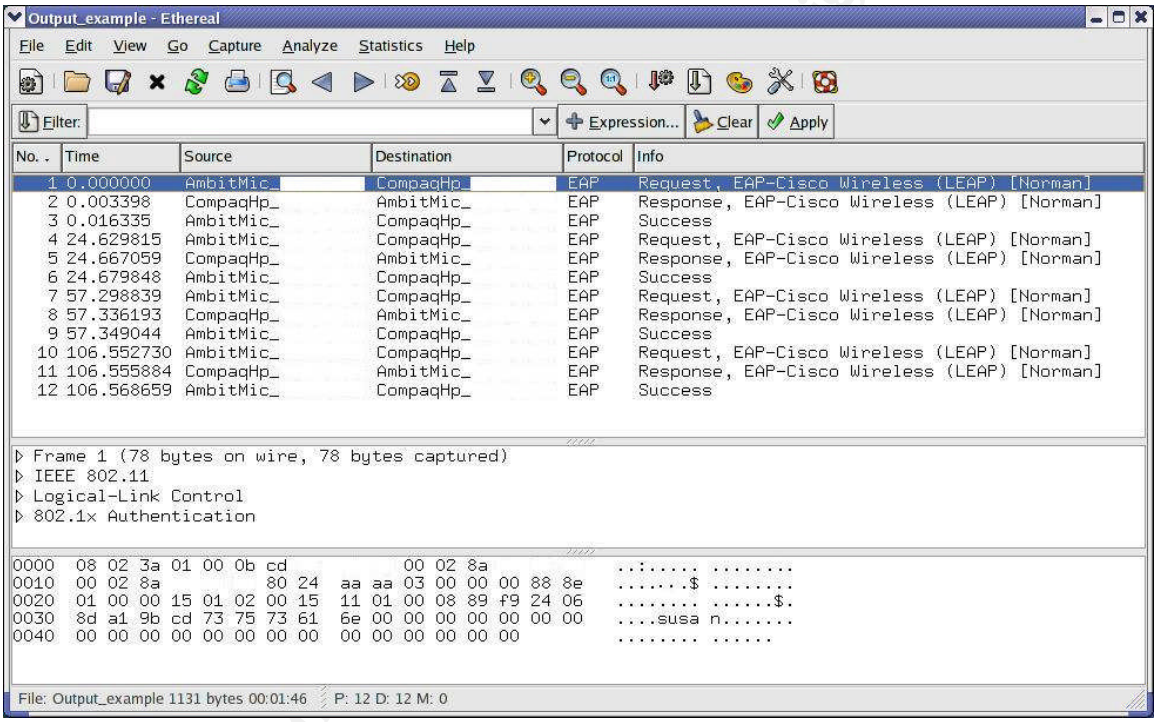
```
[root@reality asleap-1.0]# ./asleap -r Kismet4.dump -f a-z_0-9_one_to_five.dat -n a-z_0-
9_one_to_five.idx
asleap 1.0 - actively recover LEAP passwords. <jwright@hasborg.com>
Using the passive attack method.

Captured LEAP exchange information:
        username:   susan
        challenge:  89f924068da19bcd
        response:   6e8cf0bfaecc259b72df0f17464be14a666ec5e7c0e666f9
        hash bytes: 8cbc
        NT hash:    0661a2cfddabc279bd51a10e961e8cbc
        password:   scott

Captured LEAP exchange information:
        username:   mary
        challenge:  20b46a442f21cc94
        response:   8efdc3e9b8d330bdc3f07391b65beaa521fe3be69f44f4b9
        hash bytes: 0716
        NT hash:    85deeec2d12f917783b689ae94990716
        password:   love

Captured LEAP exchange information:
        username:   james
        challenge:  aee720d1850466b3
        response:   1b15bc5edc778e5ea464a913b13f70df3bc2855dbb436e6d
        hash bytes: 8d9a
        NT hash:    2559aba735d6d14f917df937d2428d9a
        password:   j4mes

Captured LEAP exchange information:
        username:   david
        challenge:  037614201eb1e487
        response:   9be01ab859092025a2d4c0832f75f214a58398446a4b95ee
        hash bytes: 0654
        Could not find a matching NT hash.  Try expanding your password list.
        I've given up.  Sorry it didn't work out.
Closing pcap ...
```

Note that only the last exchange could not be cracked, since “david” is using a strong password that is not part of the dictionary.

This last execution was completed in less than 1 second.

Asleap also has some other useful flags and they are pretty much self explanatory.

The “-w” flag writes to a file only the LEAP exchange found on the input. This input can be a “libpcap” file or an interface. This is a great feature to reduce the size or filter a “libpcap” file. The next screenshot (sanitized) is an example of one of this files opened with ethereal[17]. The source file was the Kismet4.dump used a while ago.



A common flag used in most programs is the “-v”. In asleap this option displays extra information about the LEAP packets found. An example of an output from a command line with this flag is as follows:

```
[root@reality asleap-1.0]# ./asleap -r Kismet1.dump -f a-z_0-9_one_to_five.dat -n a-z_0-
9_one_to_five.idx -v
asleap 1.0 - actively recover LEAP passwords. <jwright@hasborg.com>
Using the passive attack method.


Captured LEAP challenge:

         0802 d500 000b cd8c fc29 0002 8a0e 37bb  .........)....7.
         0002 8a0e 37bb 8071 aaaa 0300 0000 888e  ....7..q........
         0100 0014 0102 0014 1101 0008 34e6 c904  ............4...
         f58a b4c9 6d61 7279 0000 0000 0000 0000  ....mary........
         0000 0000 0000 0000 0000 0000 0000 0000  ................
         0000                                     ..
```

[17] Ethereal. “Ethereal 0.10.5”. URL: http://www.ethereal.com/

```
Captured LEAP response:

         0801 d500 0002 8a0e 37bb 000b cd8c fc29  ........7......)
         0002 8a0e 37bb 0001 aaaa 0300 0000 888e  ....7...........
         0100 0024 0202 0024 1101 0018 ca2c 03f9  ...$...$.....,..
         2bbe 212c e272 9f1b 95e9 f667 0ebd e30a  +.!,.r.....g....
         7674 9048 6d61 7279 0000 0000            vt.Hmary....

Captured LEAP auth success:

         0802 d500 000b cd8c fc29 0002 8a0e 37bb  .........)....7.
         0002 8a0e 37bb 9071 aaaa 0300 0000 888e  ....7..q........
         0100 0004 0302 0004 0000 0000 0000 0000  ................
         0000 0000 0000 0000 0000 0000 0000 0000  ................
         0000 0000 0000 0000 0000 0000 0000 0000  ................
         0000                                     ..


Captured LEAP exchange information:
        username:   mary
        challenge:  34e6c904f58ab4c9
        response:   ca2c03f92bbe212ce2729f1b95e9f6670ebde30a76749048
        Attempting to recover last 2 of hash.
        hash bytes: 0716
        Starting dictionary lookups.
        NT hash:    85deeec2d12f917783b689ae94990716
        password:   love
Reached EOF on pcapfile.
```

One last flag that must be commented here is the “-a”. This flag can speed up the process of getting users credentials by making asleap send disassociation packets over the wireless network. To use this flag, an Airjack[18] interface must be used.

It is not really easy to find much information about Airjack anymore since the main web site is down (at the time of this writing). But copies of the files can be found on some mirrors on the Internet, or on sourceforge.net[19]. Since the installation can be tricky, I recommend looking for a file called “AirjackSetup.txt” on some search-engine cache before installing the software.

Once the driver is installed and an interface called “aj0” is present in the system you can use that interesting feature of asleap.

Just as a last comment, it may also be useful to install the Airjack Tools available only in the early versions of the driver in order to correctly set the wireless channel and monitor the network.

4.4. Keep access

By using only this exploit there is no guarantee of keeping network access since all it can do is to discover valid username and password combinations.

[18] Abaddon. “Airjack”. URL: http://802.11ninja.net/airjack/

[19] SourceForge.Net. “Airjack Project”. URL: http://sourceforge.net/projects/airjack/

What is possible to do though is to collect more than one credential allowing access even in the case of a password change made by the real user or company personnel.

Of course, at this point, having access to the network may allow the use of other exploits that take advantage of other vulnerabilities present in the network.

One option would be the installation of backdoor programs in some systems or to perform what is often called a "phone-home" communication which is a technique used to make programs open a communication channel back to one machine on the outside (starting from the inside). A good example of this technique was presented a couple of years ago at DefCon[20] X by Chris Davis and Aaron Higbee and it is still my favorite example of a phone-home implementation.

4.5. Covering Tracks

Once again there is not much to do in this part if we are only looking at the LEAP exploit. This exploit may be pretty much passive all the way to the password discovery which is great from the perspective of an attacker since he/she does not have to worry about covering tracks, but it is also really scary for any security or incident handling team.

Of course, if asleap is running with the "–a" flag, disassociation packets are sent and this may be logged by some kind of wireless IDS. In this case, the attacker may want to erase the logs to prevent the disclosure of the attack and possibly a password reset or change.

The actions that may need to be hidden or erased are only the ones performed by the attacker after he/she decides to start using the credentials to connect to the network.

5. Incident Handling Process

5.1. Preparation

I heard my SANS instructor saying, maybe not exactly with these words, that the best way to handle an incident is by preventing it from happening in the first place. At that moment I got a whole new vision on how to handle incidents, and everything makes more sense now.

[20] Davis, Chris and Higbee, Aaron on DefCon X. "DC Phone-Home". URL: http://www.defcon.org/images/defcon-10/dc-10-presentations/dc10-higbee-davis-bootable/dc10-higbee-davis2.ppt

Preparation is a critical step and if done well it will prevent most incidents from happening and also help on the next steps of the handling process.

Looking back at our fictitious scenario we can see some good actions performed by the company to avoid or prevent most attacks. However, some critical pieces were missing.

Let's start by looking at the positive points first.

As we said before, the security team worked well configuring the Access Point following most best-practices about wireless configuration, including the use of dynamic WEP keys, cloaking the network, disabling some management services on the access points and exporting logs to another server via syslog. In addition to this, they were running the latest version of Cisco IOS available for that hardware and they chose to use LEAP to authenticate users instead of only authenticating devices via MAC filters or something like that.

Unfortunately, that last choice opened the door to our exploit especially because of one crucial mistake: there was no Password Policy in place at the company. This is the first mistake made at the preparation phase and probably the most critical one.

Without a password policy users can, and probably will, choose weak passwords allowing our offline brute-force attack to succeed.

Now that we could identify one negative point left behind, we can point out a few more. A poor network design was used, connecting the WLAN directly to the internal network. No IDS was used and only some limited logging was available. Host or personal firewalls were used on some devices but not mandatory for every system on the company.

These are probably the critical mistakes and configuration actions taken that are related to incident prevention. So let's quickly look at what was done in this phase to actually handle an incident.

First, a local incident handling team was created. Some employees were invited to join the team and all those willing to help were selected.

The list below shows the team members and their roles in the company. It is important to note that people from different departments were included so the team would be able to have a better understanding of the company's business and also ensure that decisions would be made without surprises or damages to specific areas of the company.

Table 2 – Fictcius Incident Handling Team

Name*	Position	Department
David D.	Security engineer	IT

Susan S.	System administrator	IT
James J.	Network administrator	IT
Larry L.	Support Supervisor	Help Desk
Sarah S.	H.R. Administrator	Human Resources
John J.	Project Manager	Consulting
Roger R.	Corporate Attorney	Legal
Richard R.	Security Coordinator	Physical Security
Pending – To be defined	N/A	Public Affairs

* These are only fictitious names used to illustrate.

The team realized they needed a person to deal specifically with public affairs. However, there was no one available at the moment so Roger, the corporate attorney, agreed on doing this job if needed.

After the team was selected, the first version of a security incident response plan was written. It still needed some improvement but it was already a good start.

A security awareness campaign was also launched, starting to alert users about security and the existence of a group of people within the company able to help anyone to identify or simply understand different kinds of events happening in the company. This campaign was closely followed by the help desk personnel since it was chosen to be the focal point of contact for end users. Banners were posted on the elevators hall and cafeteria and emails were used to promote the campaign.

A recognition method was approved by senior management to reward incident handling team members after exhaustive or well performed analysis. Also, as part of the security awareness training a smaller but still nice gift would be send to any user helping in the identification of incidents within the organization.

The team members developed an emergency communication plan to help them reach each other whenever necessary. The plan contained information about partners, local authorities, media and even some especial clients. It included the name of the people, their relation to the company as well as phone and a cell phone numbers that could be used during or after business hours. Pager numbers were included when available.

As a last step, the incident team received authorization to create a “jump kit” to be used when necessary and to store (in a safe location within a sealed envelop) a list of the administrative accounts and passwords of every business critical equipment.

5.2. Identification

When this incident occurred at Fictcius the incident handling team was a few months old and only small events had been previously analyzed. Even though some of team members were a little anxious about working on a “real” incident handling case, it was clear to all of them the importance of remaining calm and acting in an organized and structured way since any mistake could compromise the whole incident handling process.

Here, I’m going to start describing what happened and how the sequence of events occurred at Fictcius Co. Note that each member used a previously created form to document every action and piece of information obtained. Those forms were based on SANS incident handling Sample Forms[21]. The information presented in this paper is not in the same format.

Everything started on what I call here day 1 when a call was received by a help desk attendant.

The call was made at about 4:45pm by a technical consultant who was having some problems to understand why his personal firewall was blinking so much and showing some strange messages. He also said that in fact he was not the only one having that problem and that at least two or three other employees sitting close to him saw the same behavior. The attendant, following a corporate process, asked a few questions to verify the user’s identity and record a log of this call in the help desk system.

Since the help desk attendant could not conclude much from the user’s diagnostics, he used Microsoft Netmeeting to access the remote host. After a few minutes, the attendant was looking at the firewall log on the user’s machine and could see a large number of “TCP Syn” packets being dropped by the software; all of them targeting port 25. But what called his attention was the source IP address of those packets, 192.168.101.188, an IP address used in the internal network of the company.

Trying to have a better view of the situation the attendant opened a command prompt and checked which ports were open on that host. No service was using port TCP/25. So he tried to explain to the user what he had done and told him not to worry because that was probably some misconfigured machine on the internal network trying to connect to other hosts; The machine was not going to be affected anyhow since the firewall was working well. As a last action, he started the antivirus update feature on the user’s machine to improve security.

After the call, the attendant decided to do a quick search on the internet to know if any new virus using TCP port 25 had been identified, but no matches were found.

[21] SANS Institute. “Sample Incident Handling Forms”. URL: http://www.sans.org/incidentforms/

Just after 5:00pm Larry, the help desk supervisor, was getting ready to call off the day when that attendant asked him if he knew something about a new virus. Larry wasn't aware of any new virus hitting the internet lately and being an enthusiastic member of the incident handling team, asked for the whole story. Immediately Larry identified it as an event and started taking some notes.

After that, following the emergency action plan, Larry informed management about the possible incident and decided to call David, the security guy who worked hard to create the team. At this time it was already 5:12pm. By phone Larry explained what he knew so far and David decided that they should investigate the situation deeply and assumed the coordination position to handle the incident. Incident mode was on!

David knew that they would need some help from the network group so he put James, who was also a member of the incident handling team, on a conference call with Larry to better understand this new "virus thread".

It was 5:20pm and James was not answering his phone so they checked the emergency communication plan to reach James on his cell phone. James was half way home but after realizing that David was serious about handling this situation he returned to the company.

It was 5:40pm when these three technicians where sitting in front of a notebook (which was part of the jump kit), connected to the internal switch, running Linux and ethereal. They wanted to analyze and record the network traffic. So far, they knew that they had to look for any non-standard traffic as well as communications on port 25. Also, they knew the IP address of the first "infected machine" since it was recorded by the help desk.

Immediately they could see a large volume of TCP packets coming from a high port of the infected machine and going to the corporate SMTP server on the DMZ on port 25. So far there was nothing critical because that could simply mean that the employee was sending some emails. The payload of the packets did not make much sense to them, but then again, it could be attached files.

At 5:45pm James opened the firewall log while the sniffer was acquiring more data then something more intriguing appeared. Lines and lines of dropped packets were logged, all of them coming from that specific infected machine looking for hosts and ports on different network addresses. It is not a common or authorized employee's behavior to scan the network. David wanted to know who had done that.

James logged into the internal switch at 5:52pm to check the MAC tables for a port identification of that host. It turned out to be behind the wireless access point. The infected machine was in fact a wireless client.

Larry, the help desk supervisor, finished a phone call to their anti-virus software provider about that same time. There was no news about a new virus with that behavior so far. Their corporate anti-virus should be able to identify old threads.

They tried to contact the system administrator of the SMTP server, but she could not be reached. Since she was also a member of the incident handling team, they checked her alternative contact phone on the emergency communication plan but she was not available on that number either.

Even though the team had access to the administrator account on that system (through the use of the sealed envelope) no one had a good knowledge of that service in order to perform a good troubleshooting. That was not the time to test commands.

The sniffer started showing more traffic originating from that host but this time it was scans on ports 139 and 445 TCP and more email messages now with unknown source and destination email address. It was 5:58pm when David realized that they had assumed the virus idea too soon and they could be dealing with a user with bad intentions.

James logged into the access point and checked the list of current clients at 6:04pm. Only 3 wireless clients were associated at the time and all of them were in the state EAP-Associated, meaning that they had provided valid credentials to associate to the access point. Unfortunately it was not possible to know which username was used by each client and the only information available on the access point regarding user authentication was the number of successes and failures with some limited details, like failure reasons including wrong username or wrong password. It was not possible to map the IP address (or even MAC) to the username.

At that point, 6:10pm, they decided to take a small break to review what they knew. Basically they had: an active client scanning the network and using the SMTP server. The client was connected to the wireless network. The IP address of that machine was 192.168.101.188, MAC address 00:02:2D:XX:XX:XX* (*sanitized) and it used a valid username and password to associate via EAP (LEAP). The probability that they were dealing with a virus infected machine was now low. The first event that occurred related to this incident was reported to the help desk by an employee who had been included in the port scan. Firewall logs, network traffic and network devices were being analyzed and recorded. They were missing the extra information about the SMTP server. Those logs could reveal important clues at that time but there was not much to do about that.

David decided that it was important to keep upper management updated so he asked Larry to write a small report. Larry agreed on that but before doing it he joked and said that the first thing that management would do after reading that

report was to ask them to disconnect that machine. That was when James had an idea: what if they did disconnect that machine from the wireless network, but not completely, only momentarily. That should be enough for them to get more information about the authentication.

David reminded them that the last thing they wanted was to show the attacker that they had noticed his/her presence since it could send him/her away leaving the incident handling team with almost nothing.

Again, the three agreed on that, but James explained that they could only send one disassociation packet, requiring the attacker to re-authenticate and they could monitor the number of authentications per username on the access point. For the attacker, it would look like a short lost of signal on the wireless network; something pretty common that the client software should handle without big alarms.

At 6:20pm with management approval they did that. James connected again to the access point, took notes on the number of successful authentications per username and sent the disassociation packet to that client. The access point log showed that it took only 6 seconds for the client to reauthenticate. Checking the local RADIUS server statistics they saw one more success for the username "mary".

They checked Mary's full name on a list of RADIUS-user's owners and got her extension number from the intranet of the company, but there was no answer. So they called Sarah, from human resources, who was also a member of the incident handling team. They explained the situation and asked for Mary's cell phone number or, if not available, her manager's phone. It turned out that Mary did have a corporate cell phone and it was listed in the corporate database. They called Mary to know if there was something wrong with her computer and, for everyone's surprise, she was home at that time and had left the company at about 5:15pm. Someone else was using her account.

5.3. Containment

At that time they had a pretty good view of what was going on (even without knowing yet how or why). So identification phase was complete and it was time to contain the incident.

The first thing they did was to update management with the latest information and then, since most of the incident team members were not at the company at the time, they made a conference call to get everyone's opinion on what to do and how each action would impact business.

It took a while, but they finally got everyone connected expect for the system administrator that could not be found. David explained the situation and it was

clear for all of them the need to contain this incident. Many ideas on how to stop this “intruder” were presented and they were all discussed.

Of course, some of those ideas were too extreme for that time and they were left aside; other turned out to be good.

The most important decision made was regarding the wireless connectivity. Wireless access inside the company was considered non-business-critical since most people could manage to work using the wired network. It was decided then that it could be turned off until a reliable solution was implemented.

So, basically they came up with the following plan to contain the incident:

It was not efficient to block the intruder at the access point (via ACL, MAC filter or user authentication) so the wireless access point would be turned off. Mary’s password should be changed since it was compromised. They should watch network traffic for a while to see if any “phone-home” software was already present on the inside network as well as monitor router and firewall logs for any non-standard activity. And finally, start monitoring the SMTP server very closely as soon as possible.

After the call, the action plan was documented as being the team recommendation. It was signed by David and taken to management for final approval. After it was approved, they started working on it.

For the access point, the output of the commands “show tech” and “show config” was recorded as a backup of the configuration and the device state. Making backups before any changes was a rule for the IH team. So before turning the access point off they used some new tapes from the jump kit to backup the SMTP server. It was 7:45pm when the wireless network was turned off.

Mary’s password was changed on the RADIUS server. She was informed over the phone about the new password and the need of a new password change on her next logon.

No unusual packets were observed for a couple of hours after that. Apparently no backdoors were installed on the network.

With the situation under control the team could rest until the next day to focus on eradication.

5.4. Eradication

It was time to understand how the incident happened so it would be possible to prevent it from happening again. It was also important to restore the

environment to its full operational condition (if it was possible and acceptable, otherwise, clearly state the new architecture).

Reviewing every note and information gathered on the previous phases, they realized that everything started because of a compromised password being used on the wireless network. In this way, at least two points should be improved somehow: the user's password policies and the wireless network implementation.

That morning the system administrator arrived to the company and as soon as she was informed about the last day events she started checking the SMTP server.

Researching on wireless security best practices they decided it would be good to reduce the power used on the access point to prevent the signal from reaching unneeded areas. Most of the other recommended configurations were already in place.

That's when the team found out about the LEAP vulnerabilities.

Reading some good documentation about the topic, they understood the flaws of the protocol and decided to recommend the implementation of a corporate Password Policy forcing the users to have strong passwords.

SANS' Password Policy[22] template was used in the creation of the new corporate policy template that was going to be proposed. Even though the complete policy will not be presented here, the definition of "strong passwords" copied from the SANS template is the following:

Strong passwords have the following characteristics:

• Contain both upper and lower case characters (e.g., a-z, A-Z)
• Have digits and punctuation characters as well as letters e.g., 0-9, !@#$%^&()_+|~-=\`{}[]:";'<>?,./)*
• Are at least eight alphanumeric characters long.
• Are not a word in any language, slang, dialect, jargon, etc.
• Are not based on personal information, names of family, etc.
• Passwords should never be written down or stored on-line. Try to create passwords that can be easily remembered. One way to do this is create a password based on a song title, affirmation, or other phrase. For example, the phrase might be: "This May Be One Way To Remember" and the password could be: "TmB1w2R!" or "Tmb1W>r~" or some other variation.

NOTE: Do not use either of these examples as passwords!

[22] SANS Institute. "Password Policy". URL: http://www.sans.org/resources/policies/Password_Policy.pdf

The SMTP analysis did not add much except for the confirmation of a number of spam messages that were sent originating from that IP address.

By the end of day 2 a new meeting was scheduled. The vulnerability was presented as being the possible exploited point and the new policy was proposed as a solution. The impact of bringing up again the wireless access with the new level of protection (due to strong passwords) was discussed. It was accepted by the team to bring up again the service in that way while the IT team researched on new authentication mechanisms, like EAP-FAST[23].

As a last action, they decided that the security awareness training for the employees should be updated and, with a strong support from upper management, it could be now better deployed on the site.

Once again this decision was documented as being the team recommendation and the final decision was left for management.

5.5. Recovery

Upper management was able to understand and approve the recommendations based on the document created by the incident handling team.

The new Password Policy was implemented on day 4 and all employees were informed about it by email. Every user of the wireless network had their password changed immediately by the IT team and they could only start using the service again after calling the help desk and performing a new password change (meeting the new standard). With this process in place (which was possible due to the company size) the access point was once again active.

The password policy was good for every corporate system, not only for the wireless network. Every employee should change the password on each owned account while the system administrators should implement mechanisms to force the use of strong passwords.

The IT team was still monitoring for backdoors for some days and the new wireless authentication mechanism was being analyzed.

5.6. Lessons Learned

The next morning after the recovery phase was completed the incident handling team met again to record what they had learned from that incident while the information was still fresh for them.

[23] Cisco Systems, Inc. EAP-FAST. URL: http://www.cisco.com/warp/public/cc/pd/witc/ao1200ap/prodlit/eapfs_qa.pdf

First they concluded that they had found out about the LEAP flaws too late, which made the company vulnerable and without any idea of the risk it was exposed to. As an action for that, the security team would analyze and propose the utilization of a vulnerability analysis service performed by an external organization with the required knowledge.

The IT team had a new task which was to study the viability of the implementation of alternative authentication mechanisms for the wireless network. They should have a special focus on EAP-FAST so they defined a deadline for a proposal.

They realized that the preparation-phase work regarding incident handling had been very useful but they needed to update the emergency communication plan since the system administrator could not be found when needed. They found themselves actually lucky to have had such a small impact in the handling process since it could have been much worse.

The SMTP configuration was also reviewed and by a suggestion of that own system administrator, they decided it would be better to change it a little to prevent relay even from the inside network. They also planned to update the server to require user authentication on port 25/TCP. This last change should be synchronized with an awareness program to inform help desk and end users. Management approval for those changes should be easily obtained.

The good actions taken were also pointed out in order to make them habits.

After the lessons-learned meeting was over and the final document was presented to management, the team received the deserved recognitions (specially the three guys that worked harder, onsite after business hours). The technical consultant that made the first call to the help desk identifying an unusual behavior on the network was not forgotten either and his small recognition was also used as part of the advertising for the updated security awareness program.

6. References

6.1. Related Links

Abaddon. “Airjack”. URL: http://802.11ninja.net/airjack/

Astalavista Group. “Brute Force Dictionary Maker”. URL: http://www.astalavista.com/index.php?section=dir&cmd=file&id=1622

Cisco Systems, Inc. “Cisco Aironet 1100 Series”. URL: http://www.cisco.com/en/US/products/hw/wireless/ps4570/index.html

Cisco Systems, Inc. “Cisco Aironet Client Utility”. URL:

Cisco Systems, Inc. “Cisco Response to Dictionary Attacks on Cisco LEAP”. URL: http://www.cisco.com/warp/public/cc/pd/witc/ao350ap/prodlit/2331_pp.pdf

Cisco Systems, Inc. “Cisco Visio Stencils”. URL: http://www.cisco.com/en/US/products/prod_visio_icon_list.html

Cisco Systems, Inc. EAP-FAST. URL: http://www.cisco.com/warp/public/cc/pd/witc/ao1200ap/prodlit/eapfs_qa.pdf

Davis, Chris and Higbee, Aaron on DefCon X. “DC Phone-Home”. URL: http://www.defcon.org/images/defcon-10/dc-10-presentations/dc10-higbee-davis-bootable/dc10-higbee-davis2.ppt

Ethereal. “Ethereal 0.10.5”. URL: http://www.ethereal.com/

Funk Software, Inc. “Odyssey Client for HP”. URL: http://h18007.www1.hp.com/support/files/EvoNotebook/us/download/17817.html

Hacker’s Choice, The. “THC-LEAPcracker”. URL: http://www.thc.org/releases.php

Hewllet-Packard. “HP iPaq 4150”. URL: http://www.hp.com/go/ipaq http://www.cisco.com/public/sw-center/sw-wireless.shtml

Kismet Wireless. “Kismet”. URL: http://www.kismetwireless.net/

Netstumbler.Com. “Netstumbler v0.4”. URL: http://www.netstumbler.com/

Ritchie. “Red Hat Linux 9.0 + Kismet HOWTO”. URL: http://www.tipsybottle.com/technology/wireless/RedHat8-Kismet-HOWTO.shtml

Rootexploit.net. “Dictionary list”. URL: http://rootexploit.net/docs/dictionaries/

SANS Institute. “@Risk: The Consensus Security Vulnerability Alert” April 19, 2004 Vol. 3. Week 15. URL: http://www.sans.org/newsletters/risk/vol3_15.php

SANS Institute. “Password Policy”. URL: http://www.sans.org/resources/policies/Password_Policy.pdf

SANS Institute. “Sample Incident Handling Forms”. URL: http://www.sans.org/incidentforms/

Security Focus. “Vulnerabilities BID#8755”. URL:
http://www.securityfocus.com/bid/8755

SourceForge.Net. “Airjack Project”. URL: http://sourceforge.net/projects/airjack/

US-CERT. “Vulnerability Note VU#473108”. URL:
http://www.kb.cert.org/vuls/id/473108

WildPackets Inc. “Airopeek NX”. URL:
http://www.wildpackets.com/products/airopeek_nx

Wright, Joshua. “asleap home page”. URL: http://asleap.sourceforge.net