WEP Attacks and Solutions

Jack Neilson

April 24, 2018

School of Computing Science and Digital Media

Faculty of Design and Technology

Coursework Assignment

Student Name*	Jack Neilson
Matriculation Number*	1506801
Contact Number (in case of urgent need)	07990 850 875
Course	CS Hons Year
Stage	BSc
Time taken to complete (hours)	1-4 5-9 10-14 15-19 20+
Lecturer	Ian Harris
Module Name	Ethical Hacking
Module Number	CM4103
Coursework Title	WEP Attacks and Solutions
Coursework Part	Coursework
Handout Date	26th Mar 2018
Due Date	23rd Apr 2018
Submission Method	Via campusmoodle assessment dropbox

*Declaration ** This **MUST** be affirmed by signing below*

I confirm

- **That the work undertaken for this assignment is entirely my own and that I have not made use of any unauthorised assistance.**
- **That the sources of all reference material has been properly acknowledged.**

Student Signature*	J. Neilson
Date Submitted*	

For Office Use

Marker' Comments

Marker	Grade

Contents

1 Non-Technical Report

1.1 WEP Overview

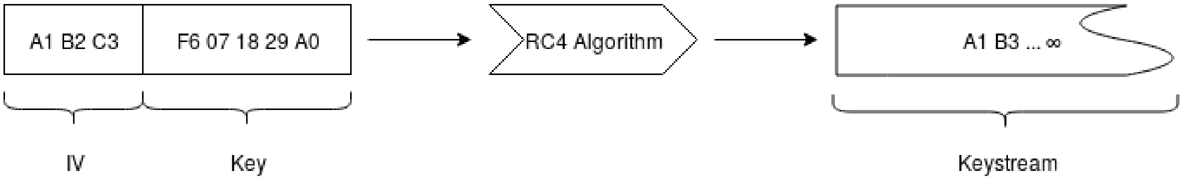
Wired Equivalent Privacy (WEP) is a method of securing over-the-air network traffic. It's used when devices connect to a Wi-Fi access spot to make sure that traffic between an access spot and an end device (such as a laptop or mobile phone) is encrypted, and isn't sent ”in the clear” as human-readable, unencrypted text. It is important that traffic is encrypted in this manner, as otherwise an attacker could position themselves between an end device and an access point and read every communication sent between the two devices, capturing sensitive information in the process.

1.2 WEP Implementation

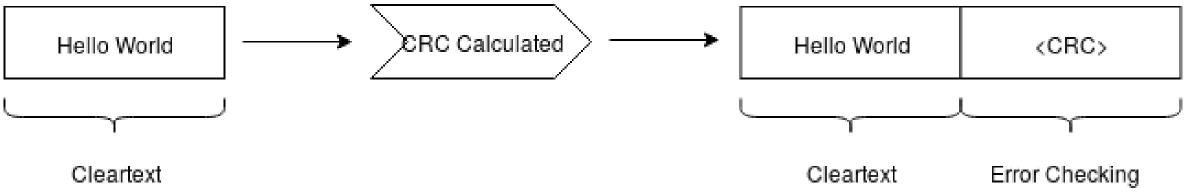
At it's core, WEP encryption consists of two components:

- An ”initialisation vector” (IV). This is a random number that is generated for each message sent over the network. It is used in conjunction with the secret key to generate a stream of pseudo-random bits that are suitable for encryption.
- A secret key. This is known to both the access point and end device, and is used when encrypting and decrypting traffic.

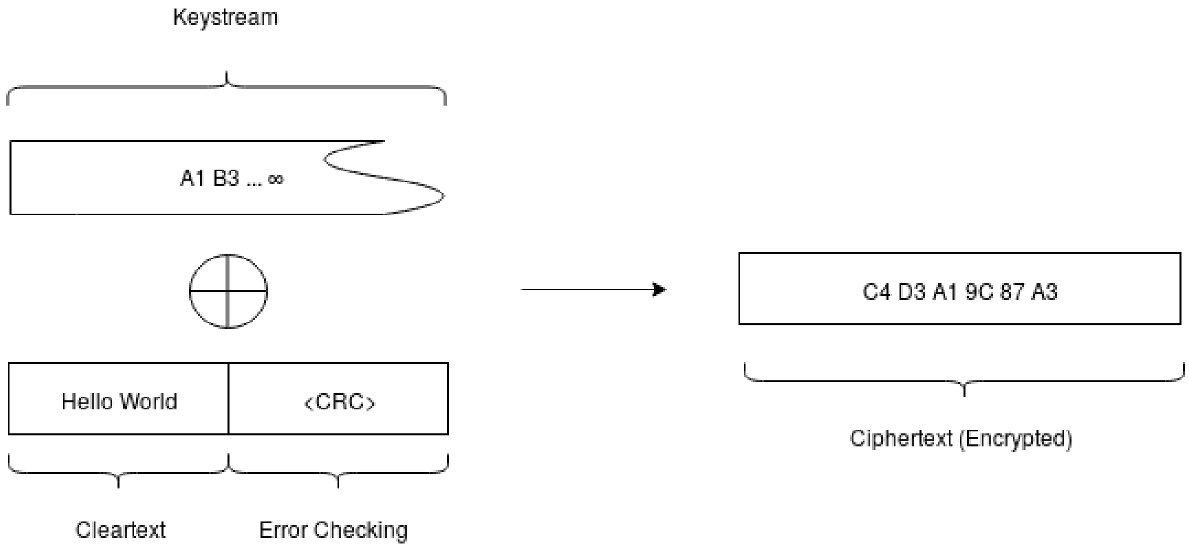
First, the IV and the secret key are joined together to give the key for the message. This is then put through the RC4 algorithm, which generates a stream of pseudo-random numbers.



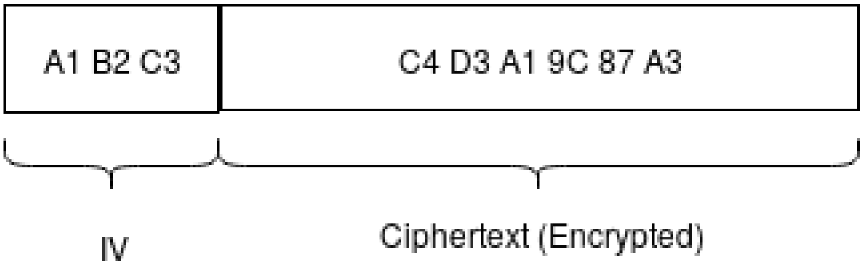
A digest (CRC) of the message is then generated to allow to recipient to check for errors.



The message and the CRC is then encrypted using the keystream.



Finally, the IV is added so that the recipient can do the entire process in reverse to decrypt to message.



1.3 Weaknesses of WEP

WEP was superceded by WPA in 2003, and was deprecated due to security concerns in 2004.

In 2006 a paper was released that showed fundamental flaws in how the WEP standard was designed, allowing attackers to gain access to the key of even the most secure (104-bit) implementations in under 60 seconds (Tews et al., 2007). This is particularly damning as the key allows the attacker to decrypt all traffic he or she captures.

1.4 Business Impact

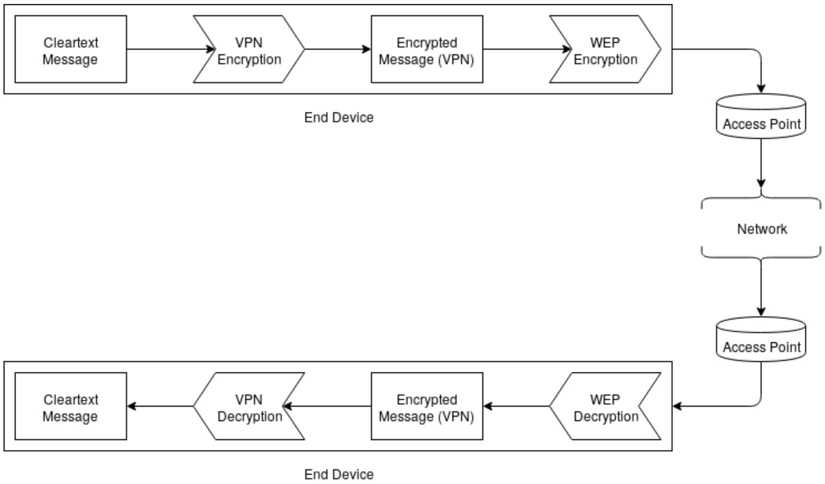
The business impact of using WEP with no other mitigations could potentially be very large. Should an attacker come in range of a wi-fi access point they could potentially gather traffic to and from multiple users. If this occurs the massive negative press could potentially bankrupt the business, legal implications notwithstanding. Industry regulators could force the business to close if customer medical or financial information is disclosed.

2 Proposed Solution

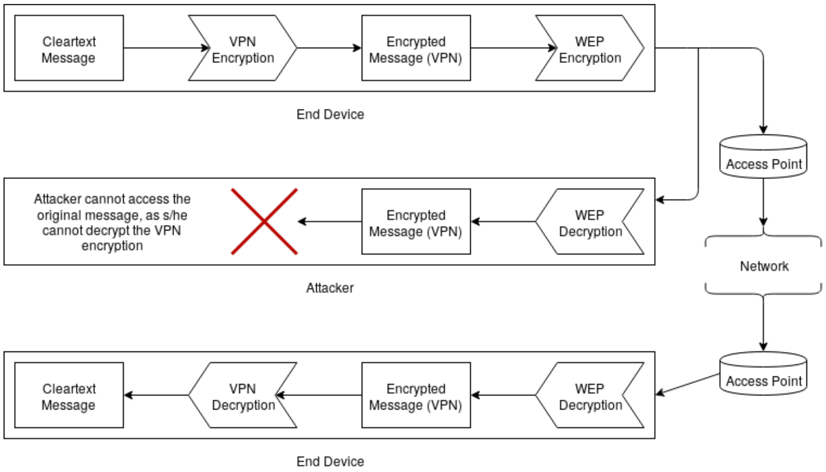
2.1 VPN

2.1.1 Description

A virtual private network (VPN) allows two private networks to be connected over an unsecured connection by encrypting traffic between two end points. In this case it would add an additional level of encryption that is ”tunneled” past WEP. It is routinely utilised by users concerned with privacy and security. If an adversary executes a man-in-the-middle attack, all they will capture is cypher text since the actual message being encrypted by WEP has already been encrypted in the VPN. A diagram of how this could be implemented is included below:

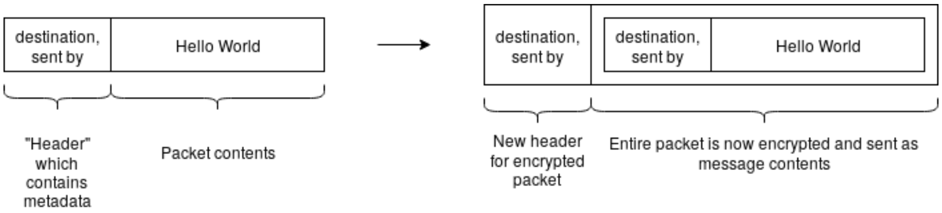


Should this be implemented as suggested, an attacker will no longer be able to read traffic gathered from the network even if the WEP key is compromised.



2.1.2 IPSEC

A VPN is not suitable in this case as it connects two trusted networks together over an insecure link, rather than two trusted machines over an insecure network. Instead, the IPSEC protocol should be used as it encrypts traffic between two machines over an untrusted network. It works by taken a packet (a message to be sent over the network), encrypting it, and then encapsulating it within a second packet as the diagram below illustrates:


destination, sent by
Hello World
"Header" which contains metadata
Packet contents
destination, sent by
destination, sent by
Hello World
New header for encrypted packet
Entire packet is now encrypted and sent as message contents

This solves the problem of an untrusted network in much the same way a VPN does, as the message can only be decrypted by the intended recipient.

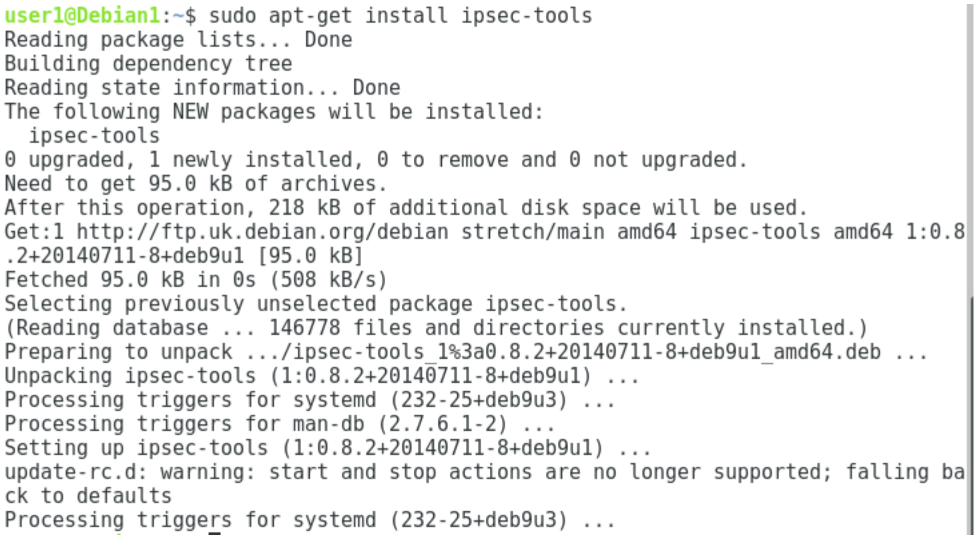
IPSEC is implemented by setting up a database of ”security associations” wherein one IP address is associated with another, along with the secret key and the encryption algorithm to be used when they communicate with each other. Policies can then be made using these associations, and stored in a security policy database.

2.1.3 Implementation

Since the we are attempting to secure the communications to two endpoints over an untrusted network rather than communication between two networks, peer-to-peer encryption must be used. This is most commonly implemented using IPSEC. This assumes that both machines have operating systems that support IPSEC. Should this not be the case (e.g. when supporting older machines) a custom solution should be used instead.

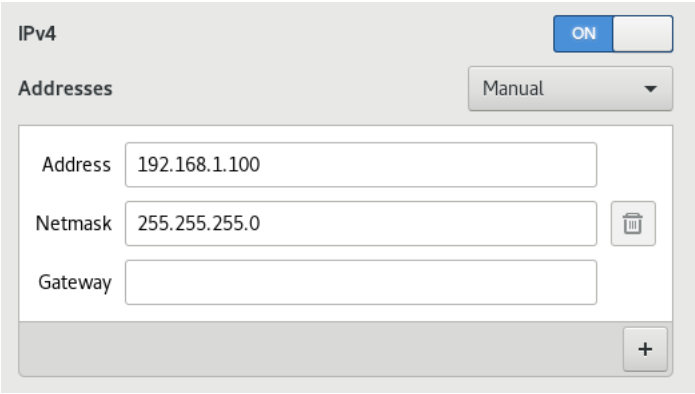
Below are instructions to set up an example IPSEC VPN:

1. Install ipsec-tools on both machines



```
user1@Debian1:~$ sudo apt-get install ipsec-tools
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following NEW packages will be installed:
  ipsec-tools
0 upgraded, 1 newly installed, 0 to remove and 0 not upgraded.
Need to get 95.0 kB of archives.
After this operation, 218 kB of additional disk space will be used.
Get:1 http://ftp.uk.debian.org/debian stretch/main amd64 ipsec-tools amd64 1:0.8
.2+20140711-8+deb9u1 [95.0 kB]
Fetched 95.0 kB in 0s (508 kB/s)
Selecting previously unselected package ipsec-tools.
(Reading database ... 146778 files and directories currently installed.)
Preparing to unpack .../ipsec-tools_1%3a0.8.2+20140711-8+deb9u1_amd64.deb ...
Unpacking ipsec-tools (1:0.8.2+20140711-8+deb9u1) ...
Processing triggers for systemd (232-25+deb9u3) ...
Processing triggers for man-db (2.7.6.1-2) ...
Setting up ipsec-tools (1:0.8.2+20140711-8+deb9u1) ...
update-rc.d: warning: start and stop actions are no longer supported; falling ba
ck to defaults
Processing triggers for systemd (232-25+deb9u3) ...
```

2. Assign machine 1 a static IP address (192.168.1.100)



IPv4 ON

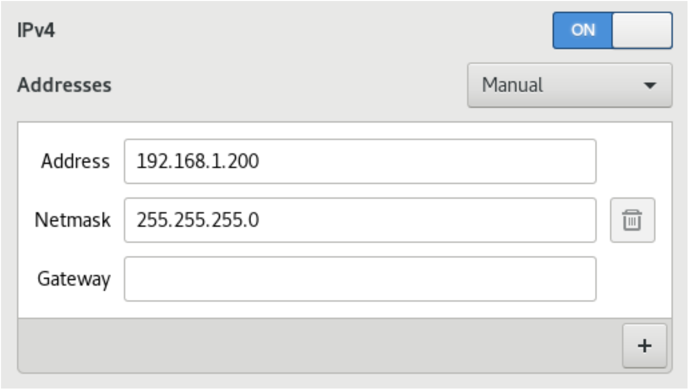
Addresses Manual

Address 192.168.1.100

Netmask 255.255.255.0

Gateway

3. Assign machine 2 a static IP address (192.168.1.200)



IPv4 ON

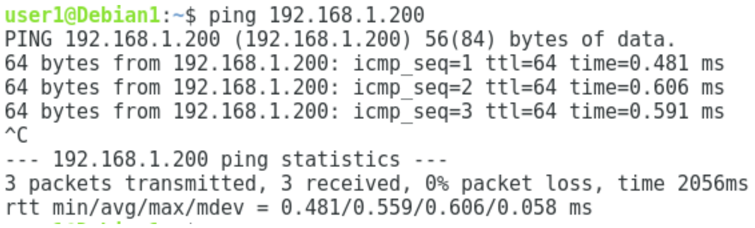
Addresses Manual

Address 192.168.1.200

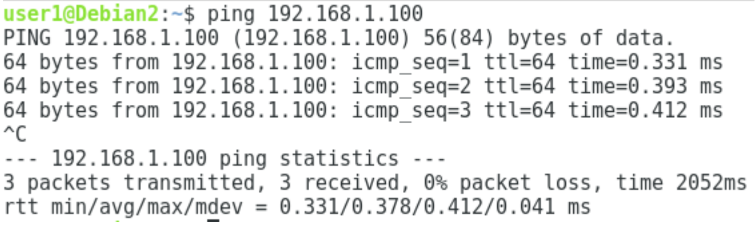
Netmask 255.255.255.0

Gateway

4. Test connectivity between the two machines

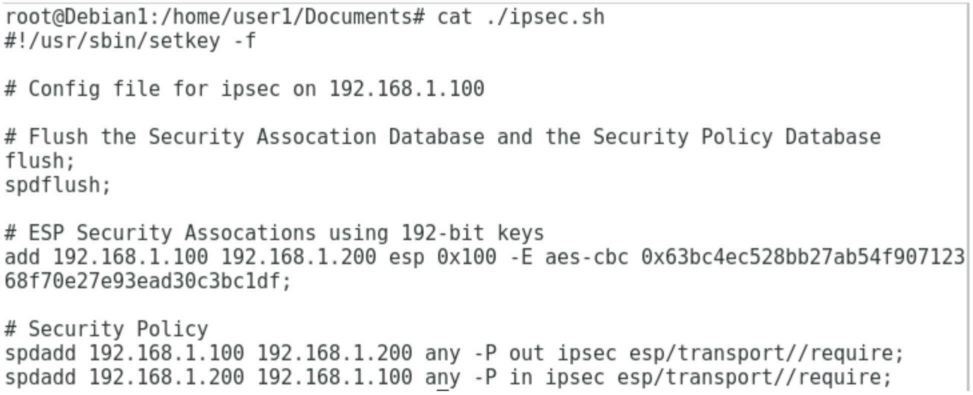


```
user1@Debian1:~$ ping 192.168.1.200
PING 192.168.1.200 (192.168.1.200) 56(84) bytes of data.
64 bytes from 192.168.1.200: icmp_seq=1 ttl=64 time=0.481 ms
64 bytes from 192.168.1.200: icmp_seq=2 ttl=64 time=0.606 ms
64 bytes from 192.168.1.200: icmp_seq=3 ttl=64 time=0.591 ms
^C
--- 192.168.1.200 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2056ms
rtt min/avg/max/mdev = 0.481/0.559/0.606/0.058 ms
```



```
user1@Debian2:~$ ping 192.168.1.100
PING 192.168.1.100 (192.168.1.100) 56(84) bytes of data.
64 bytes from 192.168.1.100: icmp_seq=1 ttl=64 time=0.331 ms
64 bytes from 192.168.1.100: icmp_seq=2 ttl=64 time=0.393 ms
64 bytes from 192.168.1.100: icmp_seq=3 ttl=64 time=0.412 ms
^C
--- 192.168.1.100 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2052ms
rtt min/avg/max/mdev = 0.331/0.378/0.412/0.041 ms
```

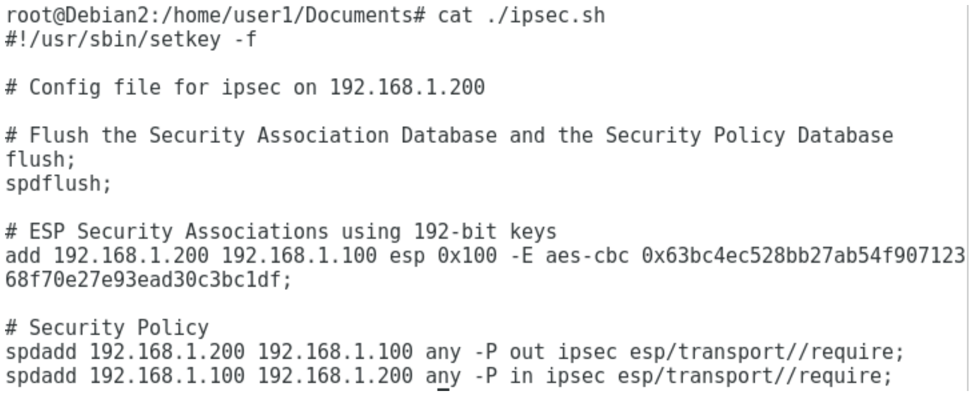
5. Run the ipsec configuration script on machine 1



```
root@Debian1:/home/user1/Documents# cat ./ipsec.sh
#!/usr/sbin/setkey -f

# Config file for ipsec on 192.168.1.100

# Flush the Security Assocation Database and the Security Policy Database
flush;
spdflush;

# ESP Security Assocations using 192-bit keys
add 192.168.1.100 192.168.1.200 esp 0x100 -E aes-cbc 0x63bc4ec528bb27ab54f907123
68f70e27e93ead30c3bc1df;

# Security Policy
spdadd 192.168.1.100 192.168.1.200 any -P out ipsec esp/transport//require;
spdadd 192.168.1.200 192.168.1.100 any -P in ipsec esp/transport//require;
```

The script first deletes any security associations that have been entered in to the database, and any policies that have been made using those associations. Then, it adds a new security association between 192.168.1.100 (machine 1) and 192.168.1.200 (machine 2). The ”esp” argument means that the packet should be encrypted, not just signed. The arguments following the -E switch are the encryption method to use (AES-CBC), and the secret key. Finally, two policies are made for incoming and outgoing traffic using the association.

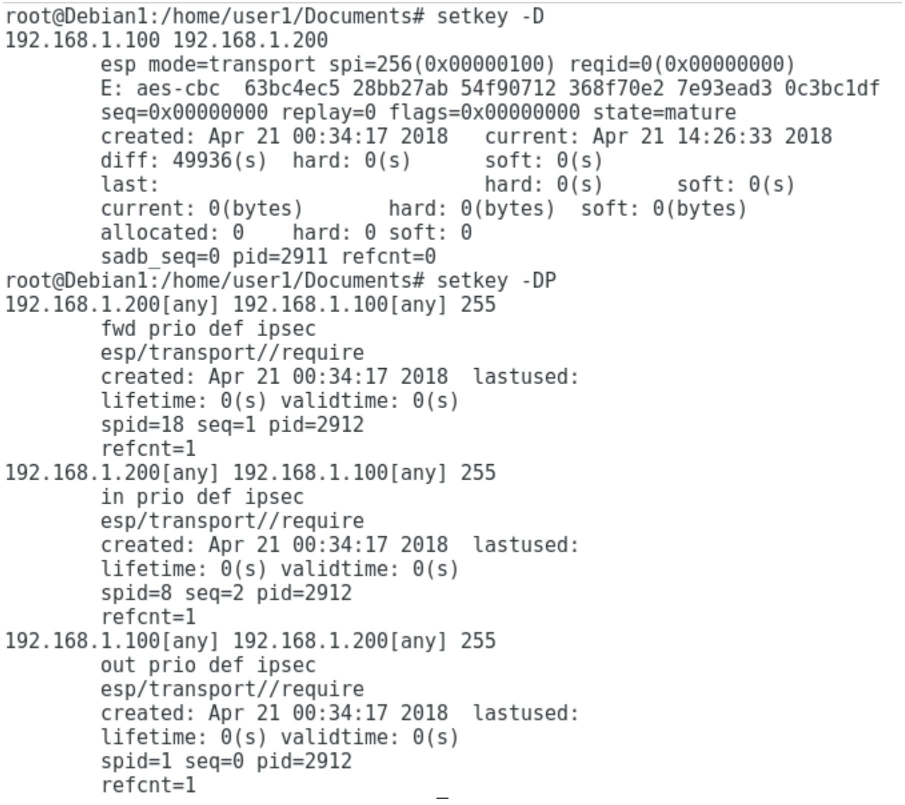
6. Run the ipsec configuration script on machine 2



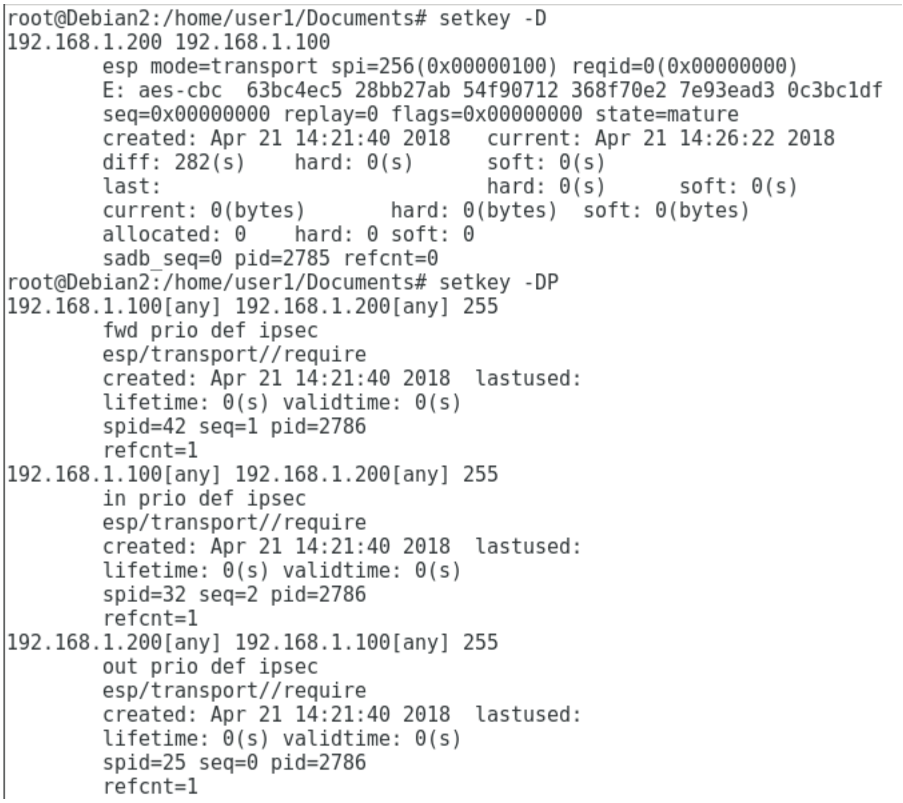
```
root@Debian2:/home/user1/Documents# cat ./ipsec.sh
#!/usr/sbin/setkey -f

# Config file for ipsec on 192.168.1.200

# Flush the Security Association Database and the Security Policy Database
flush;
spdflush;

# ESP Security Associations using 192-bit keys
add 192.168.1.200 192.168.1.100 esp 0x100 -E aes-cbc 0x63bc4ec528bb27ab54f907123
68f70e27e93ead30c3bc1df;

# Security Policy
spdadd 192.168.1.200 192.168.1.100 any -P out ipsec esp/transport//require;
spdadd 192.168.1.100 192.168.1.200 any -P in ipsec esp/transport//require;
```

This script is almost identical to the script in the previous step, however the direction of travel has been switched in the security policy.

7. Ensure both machines have been configured correctly

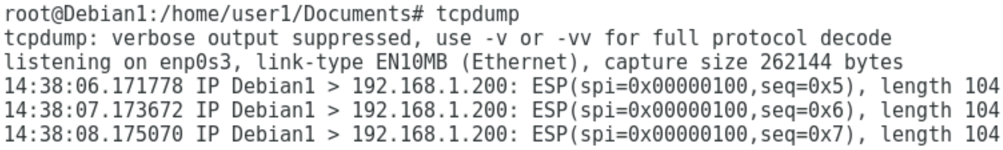


```
root@Debian1:/home/user1/Documents# setkey -D
192.168.1.100 192.168.1.200
        esp mode=transport spi=256(0x00000100) reqid=0(0x00000000)
        E: aes-cbc  63bc4ec5 28bb27ab 54f90712 368f70e2 7e93ead3 0c3bc1df
        seq=0x00000000 replay=0 flags=0x00000000 state=mature
        created: Apr 21 00:34:17 2018   current: Apr 21 14:26:33 2018
        diff: 49936(s)  hard: 0(s)      soft: 0(s)
        last:                           hard: 0(s)      soft: 0(s)
        current: 0(bytes)       hard: 0(bytes)  soft: 0(bytes)
        allocated: 0    hard: 0 soft: 0
        sadb_seq=0 pid=2911 refcnt=0
root@Debian1:/home/user1/Documents# setkey -DP
192.168.1.200[any] 192.168.1.100[any] 255
        fwd prio def ipsec
        esp/transport//require
        created: Apr 21 00:34:17 2018  lastused:
        lifetime: 0(s) validtime: 0(s)
        spid=18 seq=1 pid=2912
        refcnt=1
192.168.1.200[any] 192.168.1.100[any] 255
        in prio def ipsec
        esp/transport//require
        created: Apr 21 00:34:17 2018  lastused:
        lifetime: 0(s) validtime: 0(s)
        spid=8 seq=2 pid=2912
        refcnt=1
192.168.1.100[any] 192.168.1.200[any] 255
        out prio def ipsec
        esp/transport//require
        created: Apr 21 00:34:17 2018  lastused:
        lifetime: 0(s) validtime: 0(s)
        spid=1 seq=0 pid=2912
        refcnt=1
```

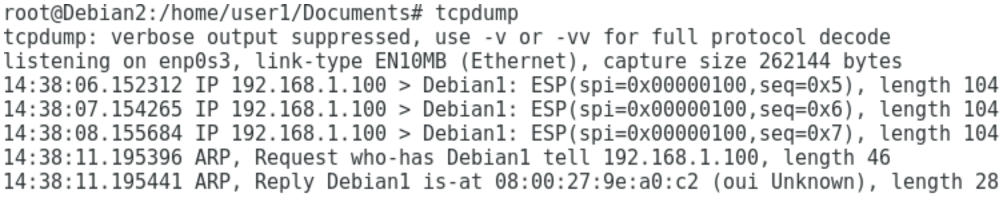


```
root@Debian2:/home/user1/Documents# setkey -D
192.168.1.200 192.168.1.100
        esp mode=transport spi=256(0x00000100) reqid=0(0x00000000)
        E: aes-cbc  63bc4ec5 28bb27ab 54f90712 368f70e2 7e93ead3 0c3bc1df
        seq=0x00000000 replay=0 flags=0x00000000 state=mature
        created: Apr 21 14:21:40 2018   current: Apr 21 14:26:22 2018
        diff: 282(s)    hard: 0(s)      soft: 0(s)
        last:                           hard: 0(s)      soft: 0(s)
        current: 0(bytes)       hard: 0(bytes)  soft: 0(bytes)
        allocated: 0    hard: 0 soft: 0
        sadb_seq=0 pid=2785 refcnt=0
root@Debian2:/home/user1/Documents# setkey -DP
192.168.1.100[any] 192.168.1.200[any] 255
        fwd prio def ipsec
        esp/transport//require
        created: Apr 21 14:21:40 2018  lastused:
        lifetime: 0(s) validtime: 0(s)
        spid=42 seq=1 pid=2786
        refcnt=1
192.168.1.100[any] 192.168.1.200[any] 255
        in prio def ipsec
        esp/transport//require
        created: Apr 21 14:21:40 2018  lastused:
        lifetime: 0(s) validtime: 0(s)
        spid=32 seq=2 pid=2786
        refcnt=1
192.168.1.200[any] 192.168.1.100[any] 255
        out prio def ipsec
        esp/transport//require
        created: Apr 21 14:21:40 2018  lastused:
        lifetime: 0(s) validtime: 0(s)
        spid=25 seq=0 pid=2786
        refcnt=1
```

8. Network traffic between the two machines is now encrypted



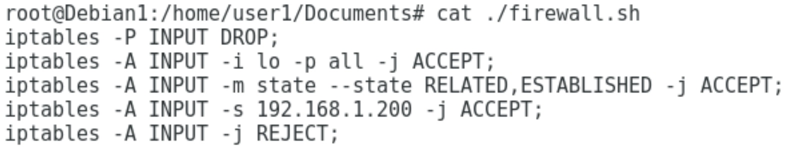
```
root@Debian1:/home/user1/Documents# tcpdump
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on enp0s3, link-type EN10MB (Ethernet), capture size 262144 bytes
14:38:06.171778 IP Debian1 > 192.168.1.200: ESP(spi=0x00000100,seq=0x5), length 104
14:38:07.173672 IP Debian1 > 192.168.1.200: ESP(spi=0x00000100,seq=0x6), length 104
14:38:08.175070 IP Debian1 > 192.168.1.200: ESP(spi=0x00000100,seq=0x7), length 104
```



```
root@Debian2:/home/user1/Documents# tcpdump
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on enp0s3, link-type EN10MB (Ethernet), capture size 262144 bytes
14:38:06.152312 IP 192.168.1.100 > Debian1: ESP(spi=0x00000100,seq=0x5), length 104
14:38:07.154265 IP 192.168.1.100 > Debian1: ESP(spi=0x00000100,seq=0x6), length 104
14:38:08.155684 IP 192.168.1.100 > Debian1: ESP(spi=0x00000100,seq=0x7), length 104
14:38:11.195396 ARP, Request who-has Debian1 tell 192.168.1.100, length 46
14:38:11.195441 ARP, Reply Debian1 is-at 08:00:27:9e:a0:c2 (oui Unknown), length 28
```

2.2 Firewall

The machine on the network that requires WEP should have every port not in use disabled, and should only be able to communicate with the trusted machine. The firewall should therefore reject all traffic from devices other than 192.168.1.200.



```
root@Debian1:/home/user1/Documents# cat ./firewall.sh
iptables -P INPUT DROP;
iptables -A INPUT -i lo -p all -j ACCEPT;
iptables -A INPUT -m state --state RELATED,ESTABLISHED -j ACCEPT;
iptables -A INPUT -s 192.168.1.200 -j ACCEPT;
iptables -A INPUT -j REJECT;
```

2.3 IDS

Since ths system is likely to be old (it only supports WEP), it will likely have unpatchable vulnerabilities. Since these cannot be fully mitigated an intrusion detection system (IDS) should be installed on the machine. There are two kinds available, network-based and host-based. If the suggestions given in this report are implemented the network should be a difficult target to attack. Therefore, it is suggested that a host-based detection system be installed to monitor traffic on the machine and report should the machine become compromised. Several commercial products are available.

2.4 Update Schedule

If the machine does not already have a set update schedule, one should be implemented to ensure it remains protected against attacks the manufacturer has created fixes for. If the machine is end-of-life or does not have significant manufacturer support (likely in the case that it only supports WEP), audits should be performed at regular intervals to find and mitigate potential vulnerabilities.

2.5 Security in depth

It should be noted that while the mitigations listed above will increase security, they are not a one-time fix. The solutions provided should be reviewed at regular intervals.

3 Ethics

As security professionals we have a responsibility to use our knowledge only within strict ethical guidelines. While it might seem common sense to some, only attempting to access machines with express permission of the owner as an unbreakable rule is perhaps not as commonplace as it should be. The skills and tools at our disposal should only be used to allow for the effective protection against hostile actors and should not be used to attack themselves.

3.1 Legal

There law that governs the majority of activities likely to be undertaken by a penetration tester (in the UK) is the Computer Misuse Act, 1990. It was originally drafted to make unauthorised access to a computer illegal, after two men who gained unauthorised access to the BBC's systems were acquitted after being charged with fraud. It prohibits a person from accessing or attempting to access a system or information contained within a system to which they do not have authorisation to access. From the text;

1. A person is guilty of an offense if-
   (a) he causes a computer to perform any function with intent to secure access to any program or data held in any computer, or to enable any such access to be secured;
   (b) the access he intends to secure, or to enable to be secured, is unauthorised; and
   (c) he knows at the time when he causes the computer to perform the function that that is the case.

The law also prohibits unauthorised access with intent to commit further offences under section 2 with heavier penalties, as well as denial-of-service attacks under section 3 (”Unauthorised acts with intent to impair, or with recklessness as to impairing, operation of computer, etc.” (*Computer Misuse Act*, 1990)).

Obviously, this single act does not cover all ethical issues within the field of information security. There still exists many actions which may be legal but are clearly unethical.

3.2 Ethical Framework

The overarching principle that should be adhered to except under specific circumstances when attempting to codify an ethical system is that an individual should never attempt to gain access to anything which is not theirs. This includes things which some may not consider, such as hardware or software which has only been licensed to the end user.

The circumstances in which an individual may attempt to gain access to a system are as follows:

1. With the express permission of the owner of the system;
2. With the express permission of a maintainer of the system, acting under the authority of the owner;
3. Within the terms set under a legally binding engagement document which outlines which systems a person may attempt to gain access to, who is liable for any damage caused and what methods of attack may be used.

It is hard to imagine a scenario in which an unethical action could be taken when following this framework (at least, in the context of information security). Penetration tests, for example, fall under section 3. Bug bounties fall under section 2.

3.3 Professional Standards

There are many professional standards that a person may follow in an information security career. Two of the most desirable certifications have been detailed below:

3.3.1 Certified Information Systems Security Professional

The Certified Information Systems Security Professional (CISSP) certification is a globally recognised professional standard that is ratified by the International Information System Security Certification Consortium, known also as $(isc)^2$. It requires candidates to pass an examination, have at least 5 years of full-time information security employment, answer questions about their criminal history and accept the CISSP code of ethics (*CISSP Code of Ethics*, n.d.).

3.3.2 Offensive Security Certified Professional

Seen by many as the "gold standard" for information security professionals, the Offensive Security Certified Professional (OSCP) certification aims to closely mirror a real-world engagement. It is offered by Offensive Security, the company behind the Kali distribution of Linux (one of the most widely used collection of tools for information security professionals). It is renowned for being extremely difficult, as candidates are expected to gain access to an unfamiliar system, document the process used and prove that access was gained by recording "flags" which change per-exam.

References

CISSP Code of Ethics (n.d.), https://www.isc2.org/ethics/. Online, accessed 24th April 2018.

Computer Misuse Act (1990), http://www.legislation.gov.uk/ukpga/1990/18/. Online, accessed 24th April 2018.

Tews, E., Weinmann, R.-P. and Pyshkin, A. (2007), Breaking 104 bit wep in less than 60 seconds, *in* 'International Workshop on Information Security Applications', Springer, pp. 188–202.