Wireless Security

Alessandro Barenghi

Dipartimento di Elettronica e Informazione
Politecnico di Milano

barenghi - at - elet.polimi.it

June 15, 2011

Recap

By now , you should be familiar with...

- System administration and userspace system programming
- Network administration and userspace network programming
- Linux kernel module programming and Netfilter tinkering

Lesson contents

Overview

- Introduction to security concepts
- Wired Equivalent Privacy and its issues
- WPA/WPA2 and its security warranties

Wireless security

Overview

- Wireless communication take place on a shared medium
- Communications may be eavesdropped by anyone
- Injection of forged packets or data corruption is not out of question too
- Ideally, we would like to have wireless links at least as secure as wired ones

Security properties

Overview

- Confidentiality: The data must not be disclosed to anyone without rights to access it
- Integrity: The data must arrive in its full, unaltered form to the recipient
- Authentication: The identity of an actor (or both) of the communication must be validated
- Non Repudiation: The author of a message cannot deny to have produced it

Symmetric Ciphers

Overview

- Symmetric ciphers are the tool employed to warrant confidentiality
- A symmetric cipher acts on a piece of clearly readable data, the plaintext
- Binds them together with a secret information the key, obtaining the ciphertext
- The same key can be employed to derive back the plaintext from the ciphertext
- Retrieving either the key or the plaintext from the ciphertext is not computationally feasible if the cipher is sound
- The best attempt to break a sound cipher is to try all the possible keys : a brute-force attack

Stream Ciphers

Overview

- Stream ciphers employ the key as a seed to a pseudo-random bit sequence generator
- The output of the P-R sequence is combined via a bitwise XOR with the plaintext obtaining the ciphertext
- The stream of P-R bits is commonly called keystream: even knowing that, retrieving the master key should be unfeasible
- The decryption is as simple as re-generating the same keystream and XOR-ing it back with the ciphertext
- Stream ciphers are very fast, and do not pose any constraint on the length of the text to be encrypted

Block Ciphers

Overview

- Block ciphers combine directly a small portion of plaintext (64-128 bits) with the key obtaining a block of ciphertext
- The decryption is performed running the same encryption algorithm backwards
- The combination must involve non-linear relations and the block of plaintext and ciphertext should ideally have zero correlation
- Block ciphers impose the constraint that the plaintext length is a multiple of the block size, if needed, padding is added

Cryptographic Hashes

Overview

- Cryptographic hashes produce a fixed length digest, given a message of an arbitrary size[a]
- The digest should change catastrophically even with a single bit change in the message :this property can be exploited as a corruption check
- A hash collision is defined as two messages having the same hash
- It should be computationally unfeasible, given a digest h , to find a message m that hashes to that digest $hash(m) = h$ (First preimage), or given a fixed message m_1, to find a message $m2$ that hashes to that digest $hash(m_1) = hash(m_2)$ (Second preimage)

[a] so, many messages actually have the same hash

Asymmetric Ciphers

Overview

- Asymmetric ciphers work with two keys: each one inverts the effect of the other
- Usually, one of the keys is publicly broadcasted and is known as the public key, while the other (the private key) is kept secret
- Encrypting with the public key, allows only the owner of the private key to decrypt the message (Encryption)
- Encrypting with one's private key, allows everyone to check that the message was sent from the private key owner (Signature)

A first attempt

WEP

- 1997: IEEE introduces the Wired Equivalent Privacy protocol
- The symmetric cipher of choice was RC4, employing 64-128(-256) bit keys
- 24 bit (the IV) of the key are sent on the transmission and change continuously
- Implicit authentication via knowing the secret key
- Integrity of the message was handled via CRC-32

A first attempt

Some concerns

- Back in 1997 WEP could not be cracked, but...
  - The most lightweight configuration uses only 40 effective key bits
  - RC4 was concerning cryptanalysts since 1995 due to a couple of tiny cracks
  - Finding a collision, first and second preimage for CRC-32 can be done with pen and paper
- The one of the key design criteria in WEP was the performance figure, so we cannot blame the designers[a] for what happened next....

[a] ok, not too much at least

Confidentiality issues

Keystream recovery

- A 24 bit IV was deemed enough to avoid encrypting two identical packets with the same keystream
- Birthday Paradox : one in $2^{12} = 4096$ packet has been encrypted with the same keystream we are looking for
- Some particular level 3+ packets are well recognizable even if encrypted (size, rate, destination...)
- Obtain a known content packet (e.g. an ARP request) and obtain a piece of the keystream
- Use the keystream to decrypt everything you eavesdropped: get one in 4096 packets on average :)

Confidentiality issues

Keystream reuse

- Since we only recovered a snippet of the keystream, we are not able to recover the master key
- ... but we can inject valid packets in the network stream!
- Eventually, it is possible to build a whole dictionary of valid keystreams and ignore the master key (takes a long time)
- Injected packets, however, may lead to hijacking (ARP poisoning) and partial DoS attacks

Integrity issues

CRC-32

- CRC-32 is computed only via a series of XOR of the message words and constant values and shifts
- It is trivial to find collisions and preimages for a CRC32 as only linear operations are involved
- Messages can be mangled locally without being detected as corrupted
- Although this is not directly an issue, it voids the integrity warranty of the protocol

RC4 cracks down

FMS

- In 2001 Scott Fluhrer, Itsik Mantin, and Adi Shamir discover an issue with the RC4 cipher
- The attack is based on the availability of many ciphertext messages encrypted with keys sharing
- A possible scenario: a large number of messages encrypted with a key composed of a *fixed unknown* part and a randomly chosen known part
- Through statistics on the messages it is possible to retrieve the whole master key of RC4
- But no one in his sane mind will ever reveal a rolling part of the encryption key of his messages....

RC4 cracks down

Attack evolutions

- After less than two weeks, the theoretical attack by FMS is applied to WEP encryption layer
- In 2001 it was possible to retrieve the whole RC4 with $\sim 100k - 1M$[a] packets with the IV belonging to a peculiar set known as weak-IVs
- The statistical biases pointed out by FMS have been enhanced in these years: the last development cracks the keys with $\sim 9k$[b] data packets
- Remember that we can induce extra traffic with the keystream recovery+packet forging trick

[a] That is, with a 2272 standard MTU, 200MB-1GB of traffic

[b] i.e. 18MB of data traffic, a matter of minutes on an average network

Practical feasibility of the attacks

Attack evolutions

- At the present moment, a regular network card is sufficient to sniff packets from a WEP protected network
- The `airodump` tool provides the means to save either all the traffic or only the relevant weak-IV packets
- The `airoreplay` tool performs all the packet reinjection attacks useful to enhance the traffic volume
- The `aircrack-ng` takes care of performing the latest improvement on the FMS attack on either live captures or dumps
- Moral : do not use WEP. Period.

WPA

Patching up the security

- As soon as the FMS attack was published, there was the sudden need to patch up things
- The first alternative to WEP was the Wi-Fi Protected Access (WPA), introducing :
  - A counter in each packet to avoid trivial replay attacks (no arbitrary replay)
  - The minimum key length was pushed up to 128 bits (no bruteforce, 40 bits are too low a margin)
  - The key management mechanism does not employ the same simple relation among different master keys for different packets (no FMS attack)

WPA

Legacies

- However, a large number of WEP-enabled network card were already sold....
- Willing to preserve their usefulness, a couple of features were kept from WEP
  - The symmetric cipher is still RC4 for performance reasons
  - The message integrity check is still done via CRC32
- Sadly, these two backward compatibility choices proved troublesome for WPA

WPA

Chopping packets

- 2008: Martin Beck and Erik Tews combine the two legacy features vulnerabilities into the ChopChop attack
- The CRC32 algorithm acts as 32-bit wise XOR sums, and the AP automatically drops corrupted packets
- The attacker tries to guess the value of the last keystream byte, and decrypts the last byte of the packet
- After that, he is able to subtract the value of the decrypted byte from the checksum and forge a 1-byte shorter packet
- Trying this for all the possible 256 values of the key byte will yield only a single response from the AP

WPA

Chopping packets

- The ChopChop Attack “only” allows the injection of packets in small timeframes
- Contrary to the FMS attack , it requires to generate a reasonable amount of traffic to decrypt a packet (128 packets per byte)
- However, this invalidates the Confidentiality property : I can read the content of the packet
- Moreover, injecting small packets is already enough to cause quite a mess (see, ARP Poisoning)

WPA-2

Another hole, another patch

- After the vulnerabilities discovered in the WPA protocol , IEEE moved on to design a solution, the WPA-2 (802.11i-2004)
- Finally, RC4 has been replaced with AES128-CCMP :)
- The message integrity is validated via a keyed MIC (an hash of the message,plus the secret key)
- The authentication is performed either via a pre-shared key or the common certificate architecture provided by RADIUS

WPA-2

see Page 196

- So, we finally ended up with a working solution? Not really...
- The IEEE 802.11i-2004 at page 196 specifies that the AP is allowed to use a group key (GTK) to talk with all the clients
- This means that everyone knows the group key in order to understand the messages...
- And since AES is a symmetric cipher, if I have the key, I can also encrypt with the same key...
- Which means I can embody the AP and noone will ever notice!

Summing up

DOs and DO NOTs

- Do NOT use WEP: it is breakable easily by anyone with a regular laptop and a common Linux distribution
- Do NOT use WPA: It takes longer to make a mess of your network, but it's still possible
- Do NOT use WPA-2 if you do not trust your network mates, they can hijack your traffic
- Do use WPA-2 since it anyways keeps the outsider as they are : out of the network
- DO USE strong pre-shared secrets (passphrases) : all the authentications can still be bruteforced

The Sledgehammer method

Secure VPNs

- Is there a way to achieve wired-equivalent privacy?
- Simply consider the whole 802.11 stack as insecure and put a VPN on top of it :)
- The most common method is to use OpenVPN on top of the Wi-Fi connection
- Pro: it's *secure*, Con: It is a bit of a performance hog for the AP