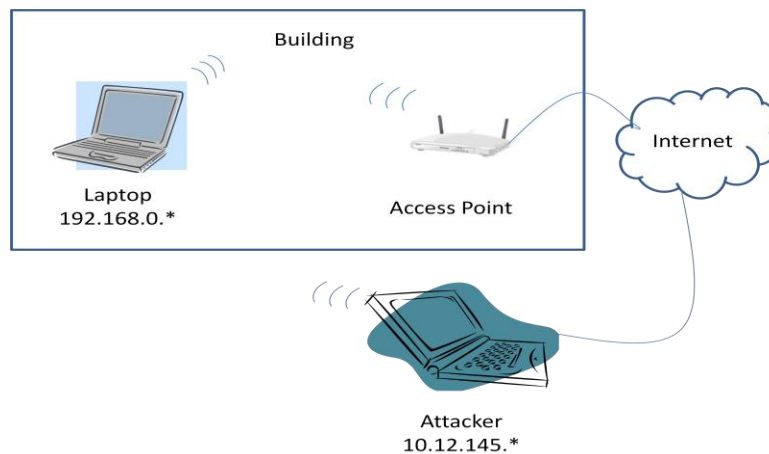


Network Security Goals:

Core Network Security Goals are

- Confidentiality
- Integrity
- Access Control

To make the terms clearer, let's look at an example.



Consider a building containing wireless devices that can legitimately connect to the Access Point over Wi-Fi to get ip-addresses in the subnet 192.168.0.*. The access point connects all these wireless devices to the Internet and acts as a wireless router. Attacker is outside the building and cannot legitimately connect to the wireless network within the building. The attacker has access to internet and has an ip-address of the range 10.12.145.*

Security goals of the system are that attacker should not be able to:

- a. Listen to our traffic - Confidentiality
- b. Inject Traffic - Access Control/Availability
- c. Modify Traffic - Integrity
- d. Block Traffic

The first three goals can be achieved using cryptography. But the last one cannot.

WEP (Wired Equivalency Protocol) Security:

WEP is used to encrypt and decrypt packets sent over an 802.11 wireless network. Here, every legitimate device has a secret key and all the devices have the same secret key (104 bits). To transmit a packet, CRC (Cyclic Redundancy Code) (32-bits) is calculated for the packet and appended to it. The

device then generates a keystream of the length of the packet from the secret key and an Initialization Vector (IV) (24 bits) using RC4 algorithm. A new IV is used for every packet. The keystream is XORed with the packet data to generate encrypted packet contents. It prefixes this new packet with the IV used for keystream generation and sends it over the network.

The receiver first calculates keystream from the IV sent in plain text and the secret key known to the receiver. It then XORs the keystream with the packet data. Due to the XOR property of $A \oplus A = 0$ and $A \oplus 0 = A$, this gives the original packet contents. It then calculates CRC over the relevant data and cross-checks the CRC in the packet. If the CRC is matched, it forwards the packet to application else discards the packet.

The attacker can listen to the packets transmitted but since he doesn't have access to the secret shared key, he will not be able to decrypt the contents of the packet and thus the first goal of confidentiality is achieved. Further, the attacker cannot modify or inject traffic as packets would be discarded if the decrypted data doesn't make sense achieving the other 2 security goals.

Importance of Initialization Vector:

A new IV is used for every new packet. If instead 0 is used as IV for all the packets, the keystream generated for all the packets would be the same as the shared secret key is the same. This will make the task of attacker to violate security goals easier. The attacker could host a web server and somehow persuade a legitimate wireless network user to access his webpage. Now, the attacker can send a known packet via internet and the Access Point would encrypt the packet using the keystream and forward it to the recipient. The attacker can listen to this encrypted packet and now, the attacker has the encrypted data as well as the plaintext data. Using the XOR property, if the two are XORed, the attacker will gain access to the keystream which will be the same for all the packets since IV is 0 for all of them. Now, the attacker can violate all the 3 security goals. Thus, it is important to use a different IV for every packet. The attack just discussed is known as chosen plaintext attack where, the attacker chooses the plaintext and gets the corresponding ciphertext.

Attacks on Initialization Vector:

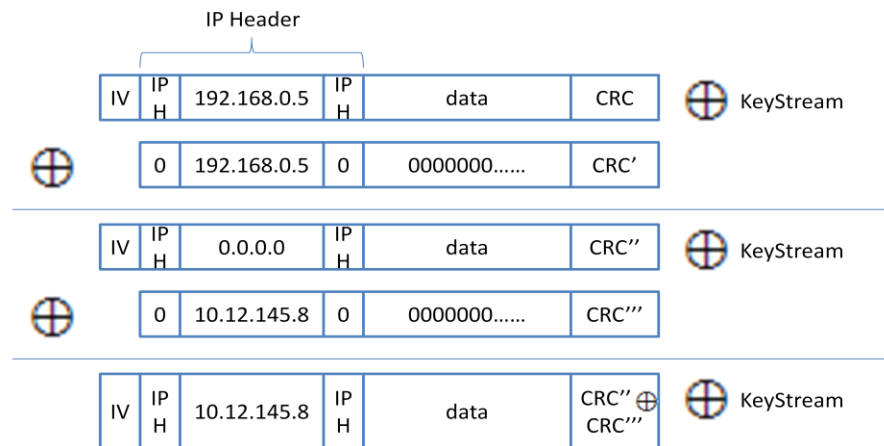
Even when IV is used, there still exist some security loopholes. The length of IV is 24-bits and IV is transmitted in plaintext. The attacker can use the above method to determine keystream for each IV used. He can create a dictionary to map each IV to its corresponding keystream. There are only 16.7 million different values of IV and once the attacker is able to create this dictionary, he can read the sent packets, modify them and even inject new ones. The problem here is that length of IV is too small. Due to that it is possible to do a brute force attack on IV. This can be solved by increasing the bit length of IV.

Another problem with IV is reuse. If the devices get restarted very often and if the restart resets the IV count, the same IV will be used again and again. Even without the above attack which takes much more time, a new faster attack can be devised for this scenario. The two packets with same IV can be XORed together to get an XOR of the plaintexts of both the packets. Then the attacker can speculate the contents of one of the packet and check to see if some valid data is extracted on XORing the speculative

data and XOR of the plaintexts. If the data is valid, he can continue speculation and figure out the contents of both the packets. The problem here is the reuse of IV. To solve this, the IV counter should not be reset on rebooting devices. However, since IV is too small, the IV value would wrap-around sooner or later. This again emphasizes on increasing the bit length of IV.

Chosen Ciphertext Attack:

Here, the attacker chooses a ciphertext and acquires the corresponding plaintext. Let's assume that the attacker knows ip-address of one of the system in the network and he wants to redirect all packets sent to that host to himself. The attacker can intercept the packet on its way to Access Point and manipulate the destination ip-address as shown in the following figure:



The packet XORed with keystream is intercepted and the attacker creates a new packet containing all 0s except for the destination address. The original plaintext destination address is placed in this packet and its CRC is appended to it. The new packet is XORed with the intercepted packet. The CRC follows the homomorphic property i.e., $CRC(P_1) \oplus CRC(P_2) = CRC(P_1 \oplus P_2)$. Thus the CRC'' for the resultant packet is a valid CRC for that packet. This effectively clears the destination address and places just the corresponding keystream in its place. The attacker then XORs the resultant packet with another manipulated packet containing all 0's except for the destination address which is set as the address of the attacker having CRC''' as its cyclic redundancy code. The attacker's ip-address gets XORed with the keystream and the final packet contains updated destination address along with updated CRC. There is an IP header checksum that needs to be manipulated as well. When this packet reaches the Access Point, it re-routes the packet over the internet to the attacker.

Keystream Extension Attack/ Reaction Attack:

Here, the attacker knows some initial part of the keystream - say first 44 bytes and depends on the reaction of the recipient for the success of the attack. As shown in the figure below, the attacker creates a packet of 41 bytes in length and appends CRC (4-bytes) to it making it 45 bytes long. He already knows the first 44 bytes of the keystream. He can generate the first 44 bytes of the Ciphertext packet and

needs just one more byte to generate a valid cipher packet. The attacker can now use brute force and try all the possible 256 values as the last byte of keystream and send those 256 packets over the network. If the recipient doesn't send any acknowledgement, the attacker tries the next value. On the other hand, if the recipient sends an acknowledgement, the speculated byte value is correct and now, the attacker knows the first 45 bytes of the keystream. The attacker can repeat the above process to keep extending the known keystream length. Hence this attack is called key extension attack. Further, the success of the attack depends on whether the recipient sends an acknowledgement or not. Therefore, this is also called reaction attack.

