Assignment 3

CS 458 Computer Security and Privacy – Shun Da Suo | sdsuo |20509411

Written Question 1: GnuPG (8 marks)

Part d) Importance of Fingerprints in GnuPG

Importance of Fingerprint:

- Fingerprints allow users to verify the authenticity of others' key without explicitly comparing the entire key (which is too long and cumbersome to compare manually).

Method to check fingerprints:

1. Alice need to obtain a copy of the fingerprint through a secure and trusted channel. This ideally should be either in person for visual verification, or through telephone for audio verification. Alice should be extremely confident about the authenticity and integrity of the "trusted" fingerprint.
2. Use `gpg --import key.asc & gpg --fingerprint` to compute fingerprint of Bob's key.
3. Carefully examine and compare the computed fingerprint with the trusted fingerprint, and only sign a key that has the correct fingerprint.

Types of possible attacks:

1. If Alice does not obtain the trusted copy of the fingerprint through a secure channel, Mallory can counterfeit a fingerprint that allows the malicious key to be verified.
2. This allows Mallory's malicious key to be signed by Alice, and in turn trusted by all users who trust Alice.
3. Then, with the private key, Mallory can decrypt all messages sent to Bob.

Written Question 2: Length Extension Attack (10 marks)

Part a) Explanation of Exploit

Vulnerability in Structure of MD5:

This exploit makes use of the iterative structure of Merkle-Damgård hashes. MD5 falls in this family of hash algorithms, and thus have the same vulnerability.

The key property of the iterative design is that one can compute the hash of "longer messages that start with the initial message and include the padding required for the initial message to reach a multiple of 512 bits", from "only the hash of a message and its length"

The Exploit:

This allows attacker to fabricate arbitrary valid requests without having access to the secret API key of the third-party application. To illustrate how this is done:

Consider a genuine request (message, mac), with mac calculated as:

$$mac = hash(secret \mid message) = hash(secret \mid message \mid padding)$$

We can forge a request (message', mac'), with:

$$message' = message \mid padding \mid extension$$

$$mac' = hash_{mac}(extension) = hash(secret \mid message') = hash(secret \mid message' \mid padding')$$

where $hash_{mac}$ refers to the hash function with its internal state initialized to the original mac. This new (message', mac') pair can be verified, due to the iterative structure of the vulnerable hashes.

Part b) SHA-256 Vulnerability

The "message extension" exploit should also work on SHA-256, since it is also an iterative hash algorithm of the Merkle-Damgård family.

SHA-256 Being a Random Oracle:

This implies that assuming SHA-256 as a random oracle in security proofs is seriously flawed and have real-life consequences. A random oracle should be a truly random function that does not take exponential time to evaluate. This, however, is impossible according to information theory.

Thus, security systems built on supposedly rigorous security proofs may not be secure in practice, since practical hashing algorithm (such as SHA-256) is far from being a real random oracle. If its output is indeed random, one should not be able to conduct length extension attacks on it.

Part c) Hash-Based Message Authentication Code

General Description:

In comparison with the simple and insecure MAC mechanism discussed before, the enhanced HMAC mechanism makes use of an additional layer of hashing. The mechanism is defined as follows:

$$i\_secret\_pad = XOR(secret, i\_pad)$$
$$o\_secret\_pad = XOR(secret, o\_pad)$$

$$mac = hash(i\_secret\_pad \mid message)$$
$$hmac = hash(o\_key\_pad \mid mac)$$

Prevention of Length Extension Attacks:

HMAC effectively prevents length extension attack with its double hashing structure. Since the attacker only controls variable-length input to the inner hash, the outer hash is not exposed to length-extension attacks.

In other word, the intermediate mac will have constant length, regardless of what message' the attacker picks in place of message.

Written Question 3: Relay Selection in Anonymity Networks (14 marks)

Part a) Attacker Control of Circuit

If the attacker controls guards, middle, and exit relays of a circuit. The user effectively loses all privacy guarantee when his or her request travels through the circuit. The attacker can piece together information about who is accessing what at which time by combining request histories from all relays of the circuit.

Illustration:
The guard relay g received a request r_1 from user u (identified by its IP address), and makes a request r_2 to a middle relay m_1 at time t_1.

Then the attacker can retrieve request logs from middle relay m_i to look for a request from g at time t_1, and determine the next relay m_{i+1} in the circuit.

By following this trail, the attacker will eventually reach the exit relay, and retrieve the destination that user u is visiting.

Effect:
The unlinkable anonymity guarantee provided by Tor is compromised and reduced to linkable anonymity. The attacker can connect all transactions to a user's IP address, which could be linked to the user himself.

Part b) Partial Control over Changing Circuit

1. Attacker can no longer link all of a user's actions to the user, since it only has control over the guard relays with a fraction of the times. This increases anonymity guarantee for users who would otherwise have the entirety of their digital history linked to their identities. Thus, users may be able to retain unlinkable anonymity since attackers may not be able to deduce identity with its limited access to each user.
2. Attacker can now link a fraction of all user's actions to the user, since every user may pick a malicious relay as its guard relay with a certain probability. This also has the effect of reducing aggregate anonymity, allowing the attacker to collect general information on users of Tor and sites visited through Tor.

Part c) Adversary Controls Destination

Consider the case where the attacker controls A relays out of a N relay network. We are interested in the case where the guard relay is controlled by the attacker, since it is possible to correlate requests at guard relay and destination via timing and volume.

Since there is a A/N chance that a malicious relay is chosen as the guard node, A/N of the connections the user make to the email account can be linked to the user.

Part d) Static Circuit

Disadvantage:
If a user initially chooses malicious relays as the guard and exit relays, the attacker will be able to connect all of the user's actions back to the user. As mentioned in previous parts, this means the anonymity guarantee is reduced to linkable anonymity for users of these infected circuits.

Advantage:
If a user does not choose a malicious relay as the guard relay, the attacker will not be able to track the user's identity at all. If the attacker only has control over a small percentage of the relays, he or she will only be able to track a small percentage of all users of Tor. Thus, an attacker would need full control over all relays to gain information on all users.

Part e) Small, Fixed Set of Guard Relays

This makes sense since the guard relay is the most important relay concerning the user's identity. If an attacker has control over both ends of the communication channel (the guard relay, and either the exit relay or the destination address), he or she can effectively determine who is visiting what.

Now, consider the case where the attacker controls the destination address, and A relays out of a N relay network. If the guard relay is chosen randomly from all available relays, the attacker is expected to be able to identify any user A/N of the times. On the other hand, if the guard relay is constrained to a small set of R relays, only a small fraction of the users' guard relays can be tracked.

Overall, it is better to leak larger fraction of *a few* users' traffic, rather than leaking a small fraction of *all* users' traffic. This is intuitive since it's only necessary to observe a small fraction of actions to profile a user. Knowing more is no better for the attacker.

Part f) Ordering and Delay

Attacker can match sequences of requests seen at the source relay and destination relay based on its pattern of volume and timing (i.e. quick succession of requests with the same target address). As a result, attacker can determine who (from guard relay) is visiting what (from exit relay).

Part g) Anonymous Publishing

Tor hidden services protocol is designed to allow users to offer services such as publishing while hiding their locations (i.e. IP address). The process relies on "rendezvous points" between users and services. A high-level overview of the entire process is as follows:

1. Server randomly picks relays as "introduction points", connects to them via Tor circuits, and tells them the public key of the service.
2. Service publishes signed descriptor of location of introduction points and public key to a distributed hash table.

3. Client retrieves the signed descriptor, and makes Tor circuit to randomly picked relay, and gives it a one-time secret for it to become a “rendezvous point”.
4. Client tells the hidden service the address of the “rendezvous point” and the one-time secret via encrypted message (to the hidden service’s public key) to the “introduction points”.
5. Server creates Tor circuit to the “rendezvous points”, and communicates with the client after verifying its one-time secret.

Bonus Question 1 (6 marks)

Pros of Government Surveillance Protocol

A surveillance program allows the government to detect potentially malicious communication. This may improve physical security of the citizens by deterring criminal activities that may be coordinated through the Internet.

This protocol is relatively simple to implement and maintain for the government, and provides strong cryptographic guarantee as long as the private key is not leaked.

Cons of Government Surveillance Protocol

This government surveillance protocol makes the government datacenter a single point of failure. If an attacker can compromise the private key, he or she can intercept and decrypt all messages sent with the protocol. This creates a significant privacy concern, since the security of the internet would depend on the security of the government datacenter.

This protocol does not enforce forward secrecy. This implies that those in charge of the private key will be able to decrypt and inspect all historical messages. With changing government officials and document classifications, it may be difficult to manage confidentiality and security clearance.

Furthermore, all previous discussion assumes that the government is a trust worthy entity. In reality, this is not necessarily true. Political parties may abuse the power to intercept all internet communication to sway elections, push policies, and etc.