Brendan Song

b7song

20466121

CS458 Assignment 3

1d.
Fingerprints are crucial in GnuPG since they are the verifying factor for people to ensure the keys being exchanged are actually from the person they want. Fingerprints should be checked in person if possible but something like over the phone also works if the participants know each other well enough. If fingerprints are not truly verified before being signed, there is the possibility for impersonation. The attacker can pretend to send a key as another person and ultimately decrypt anything sensitive the participants are trying to send to one another. This is troublesome since signing a false key makes the system less trust-worthy and damages the reputation of anyone's keys the person has signed.

2a.
The exploit used took advantage of known data structures and MD5's length extension exploit. Since the hash was calculated using a simple concat of the secret key, field names, and values, the attacker could add field names to the request and inject data. The length extension exploit allowed attackers to construct API requests that are seemingly from a third party application without actually knowing the secret key.

Since MD5 adds padding to blocks that are shorter than 512 bytes, attackers could simply provide padding to reach the 512 bytes then pass in their own values to construct requests. Since the first block contains the secret key, the attacker's request is already authenticated and the server trusts any hashed blocks that follow.

2b.
SHA-256 is also exploitable using length extension. The random oracle serves as an external source for hash calculations. It works on the bases that it returns consistent responses and it implements a random function. Since SHA-256 is not completely random, it does not necessarily work well according to the author of the article. This means attackers could construct requests to the oracle without knowing the secret key.

2c.
Hash-based message authentication codes are mechanisms for message authentication using cryptographic hash functions. HMACs require a hash function and a secret key to generate a stream of bytes of a specific length. The key is XOR'd with a fixed inner string and the intended data before being passed through the hash function and iteratively hashed again with a fixed outer string. The result is a message authentication code that can be used to check the integrity of information transferred over an unreliable medium.

HMACs mitigate length extension attacks by hashing twice. The inner hash serves as compression for the data while the outer hash serves as a message authentication code by hashing (key || inner). Since the attacker can only pass data to the inner hash, they are unable to control the length of the input to the outer hash and thus the length extension attack is prevented.

3a.
If an attacker controls guard, middle, and exit relays of a circuit then they control all traffic moving through the circuit and can determine who is attempting to reach where. They can also even redirect traffic to serve malicious exits to users.

3b.
Periodic changes of relays results in greater unlinkable anonymity since users are even harder to distinguish as they cannot be linked to guard nodes over time. However, users who end up in the malicious relays end up on the -nymity slider at linkable anonymity since the attacker can identify where the requests are coming from and where they were trying to reach. They will only have the IP of the user but are able to use the Tor network as an angle of attack.

3c.
Under the assumption the attacker controls the entire circuit, that is guard, middle, and exit relays, then they can link all incoming connections to the email account. Since the user must connect using the email account, the adversary can identify them at the guard relay and simply mock that to the user that they are hitting different relays when in reality the adversary is simply moving their traffic. However, if the relays are changed then the adversary may lose track of the user if the guard relay is no longer being used.

3d.
Advantage: Over time users will become unlinkable. Since circuits do not change, the middle relays will always get traffic from the same connections and will be unable to differentiate them as more requests are made. This differs from the periodic changes model since the circuits in that model are short-lived and likely does not result in a high volume of traffic to help diffuse and add noise to user requests.

Disadvantage: Users stuck with the malicious circuit cannot escape. The adversary would be able to monitor everything coming through their circuit and would not have to worry about losing track of users to other guard relays. This defeats the whole purpose of Tor and results in linkable anonymity for those unlucky users who use the malicious guard relay. This differs from the periodic changes model since users in such a network would eventually be switched to a different guard relay that is hopefully not malicious.

3e.
The guard relays are probably the most important since they are the direct link to identifying the user. By only trusting a handful of guard relays, Tor is able to guarantee the anonymity of users when they first connect. Since all further traffic is encrypted, even malicious middle/exit

relays cannot identify who the original user is, they only know it comes from one of the trusted web servers.

3f.
The guard node can simply limit requests to 1 at a time so it knows when the exit node has been hit and when the response is returned. By halting all other traffic coming through the guard node, there is guaranteed to only be one package returned at the end. This can be traced to the exit node since it will also only receive and forward a single package. The system is slowed down tremendously but the source and destination have been identified since there is a lack of volume and noise to increase anonymity.

3g.
Tor Hidden Services allows for anonymous publication by using a rendezvous point and one-time secret to ensure anonymity. The user enters the tor network like usual but also describes a rendezvous point which is the location for the picture to be published. This information is passed along as normal until an introduction point makes the connection to the rendezvous point and in this case would upload the picture. A success response would be returned and the client would be notified. This process ensures anonymity since the request to upload the picture comes directly from the Tor cloud where messages have already been encrypted and the receiving server only knows the picture came from an introduction point.