Nick Weaver
Fall 2018

CS 161
Computer Security

Homework 4

Due: Monday, 12 November 2018, at 11:59pm

Instructions. This homework is due **Monday, 12 November 2018, at 11:59pm**. No late homeworks will be accepted unless you have prior accomodations from us. This assignment must be done on your own.

Create an EECS instructional class account if you have not already. To do so, visit https://inst.eecs.berkeley.edu/webacct/, click "Login using your Berkeley CalNet ID," then find the cs161 row and click "Get a new account." Be sure to take note of the account login and password, and log in to your instructional account.

Make sure you have a Gradescope account and are joined in this course. The homework *must* be submitted electronically via Gradescope (not by any other method). Your answer for each question, when submitted on Gradescope, should be a single file with each question's answer on a separate page.

Problem 1 ***Wi-Fi Troubles*** **(15 points)**

Below is a description of the WPA Handshake Model to generate the PTK. Recall that the PTK (The ephemeral key encrypting a session between the client and AP) is determined from the ANonce, SNonce, MAC address of both the client and AP, and a pairwise master key (PMK), commonly known as the value sticky-noted on your refrigerator wall.

1. M_1: The client (also called the supplicant) receives an ANonce and handshake replay counter r from the access point (also called the authenticator).
2. M_2: The client now generates a SNonce. At this point, the client can derive the PTK as it has all the neccessary information. The client sends over the SNonce, along with the same replay counter r, and an tag to insure integrity (MIC).
3. M_3 : Once the authenticator has received the second message, it will verify integrity and send along the group temporal key (a separate key for encrypting broadcast and multicast messages), encrypted and MAC'd with the PTK with replay counter $r + 1$. At this point, state is initialized within the client and another counter t for transmitted data will be initialized to 0.
4. M_4: The client now confirms that the PTK generated in step 2 is to be used and responds back with an ACK with a replay counter of $r + 1$.

When the client wishes to send a message, the client will encrypt and MAC the data using a pair of keys (K_c and K_a) derived from the PTK. Furthermore, a counter mode is commonly used to encrypt the data and so we use a nonce (the client's MAC address), and a counter t (In this problem, consider CTR. In practice GCM is used, but this is changed for the purposes of simplicity).

It is not uncommon for messages to be lost in transmission. If the AP has not received M_4 within a certain period of time, it will resend M_3. Upon receiving another M_3, the client will reinstall the same PTK and in doing so, reset the data counter t to 0 again (ANonce and SNonce remain unchanged).

(a) Does this protocol provide forward secrecy, i.e. if the PMK becomes leaked are messages still secure? Why or why not?

(b) Suppose you wish to perform a man-in-the-middle attack against a client and AP. You have the ability to construct arbitrary packets, drop arbitrary packets, replay packets, etc. You do not have the PMK. Demonstrate an attack which allows you leak data encrypted with the PTK.

Hint: Recall Homework 3.

(c) How would you defend against such an attack?

(d) Assume your victim is accessing sensitive information over HTTPS. Can an attacker retrive that information without performing additional MitM attacks?

Problem 2 ***Abusing ARP and Routing Tables*** **(15 points)**

Mallory is an evil employee at GoodCorp, an organization that provides an online portal to answer questions lost and confused tourists to the Bay might have. Mallory wants all of the tourists to stay confused, so that her favorite restaurants won't have any waittime from the extra visitors.

One day, she walks into Brewed Awakening to get some coffee when she sees Albert sitting in the corner, happily sipping on coffee and answering all the tourist questions posted to GoodCorp. Mallory knows that Albert is the most proficient and helpful employee at GoodCorp, and if she doesn't do anything to stop him, there will be no hope of Mallory having lunch at her favorite seafood place on Embarcadero! She decides that she must find a way to impersonate Albert on GoodCorp and answer questions incorrectly in order to lead the tourists astray. Mallory knows a few things about network attacks that she could launch against Albert.

Note: Any answers involving physical harm to Albert, stealing his laptop, etc. will receive no credit.

Brewed Awakening is using a WiFi network with WPA-Enterprise encryption, which prevents her from eavesdropping on traffic not intended for her (i.e., she can only see packets sent to her own machine and broadcast packets). Therefore, she's going to need to exploit other network protocols. In particular, in parts (a)–(b), her attack must involve exploiting ARP.

ARP is a protocol to help systems discover the link-layer address (e.g., Ethernet address or WiFi address) of other machines, given we know the IP address of that machine. It works like this. Suppose my machine wants to send a packet to a system on the local WiFi network, but it only knows that that system's IP address is `1.2.3.4`; it doesn't know the WiFi address of that computer, which we need to send directly using the local network. To do so, my machine broadcasts across the local network an ARP request, which asks "What is the MAC address of the computer with IP address `1.2.3.4`?" When the computer with that IP address sees this broadcast packet, it responds with an ARP response, which contains the answer: e.g., "The computer with IP address `1.2.3.4` has MAC address `00:1A:AA:BB:CC:DD`."[1] The ARP response is sent directly (not broadcast) to the machine that sent the ARP request, i.e., my machine. When my machine receives this ARP response, it stores the answer (in the "ARP cache") for future use, and all future IP packets to `1.2.3.4` will be transmitted by encapsulating them in a (non-broadcast) WiFi packet with the WiFi destination address set to `00:1A:AA:BB:CC:DD`. Assume that if a machine receives multiple ARP responses, it uses the last one that it received; this is a typical implementation.

You may assume for this problem that GoodCorp associates an employee with a posting request by using cookies as authenticators. Also, GoodCorp uses `http` and sends everything in plaintext. You may also assume that Brewed Awakening's router and Albert's machine re-set their ARP cache every 4 hours.

[1] MAC addresses are 48 bits long, which by convention is expressed using pairs of hexadecimal digits separated by colons.

(a) If Mallory waited until the next time Albert showed up at Brewed Awakening (e.g., waited for Albert to connect to the network for the first time that day) how could she arrange to receive all of the traffic that Albert's browser sends to GoodCorp? How could she use this information to impersonate Albert on GoodCorp and post bogus answers?

(b) Fearing Albert's counter attack, Mallory needs to make sure that Albert doesn't figure out what she is up to. In particular, she wants to modify the data that GoodCorp sends to Albert, so Albert doesn't notice her bogus answers when he views the site.

How can Mallory extend the attack in part (a) to alter the responses he receives from GoodCorp's server so Albert won't figure out her diabolical plan?

(c) Now suppose Brewed Awakening's WiFi router wasn't using any encryption, so Mallory could eavesdrop on all packets sent on Brewed Awakening's WiFi network, broadcast or not. She fires up Wireshark. By the time she got to Brewed Awakening, Albert had already logged in to GoodCorp, so she didn't capture his password. Describe how she could use her ability to eavesdrop to enable her to later log in to Albert's account on GoodCorp, without sending any forged packets on the Brewed Awakening WiFi network.

(Guessing Albert's password on GoodCorp won't work; all GoodCorp employees are required to use 20-character passwords. Trying to get malware onto Albert's laptop won't work, either: Albert is too careful an employee for that. Mallory must find an attack that takes advantage of her ability to eavesdrop on the TCP connection between Albert's laptop and GoodCorp.)

Problem 3 ***DNSSEC*** **(20 points)**

DNSSEC (DNS Security Extensions) is designed to prevent network attacks such as DNS record spoofing and cache poisoning. When queried about a record that it possesses, such as when the DNSSEC server for `example.com` is queried about the IP address of `www.example.com`, the DNSSEC server returns with its answer an associated *signature*.

For the following, suppose that a user R (a resolver, in DNS parlance) sends a query Q to a DNSSEC server S, but all of the network traffic between R and S is visible to a network attacker N. The attacker N may send packets to R that appear to originate from S.

(a) Suppose that when queried for names that do not exist, DNSSEC servers such as S simply return "No Such Domain," the same as today's non-DNSSEC servers do. This reply is not signed.

Describe an attack that N can launch given this situation.

(b) Suppose now that when queried for a name Q that does not exist, S returns a signed statement "Q does not exist."

1. Describe a DoS attack that N can launch given this situation.
2. Can N still launch the attack you sketched in part (a)? If so, explain how this attack would work. If not, explain why the attack no longer works.

(c) One approach to address the above considerations is to use NSEC Records. When using NSEC S can return a signed statement to the effect of "when sorted alphabetically, between the names N_1 and N_2 there are no other names." Then if the name represented by the query Q lexicographically falls between N_1 and N_2, this statement serves to confirm to R that there's no information associated with the name in Q.

NSEC has a shortcoming, which is that an attacker can use it to *enumerate* all of the names in the given domain that do indeed exist. To counter this threat, ithe NSEC3 Record was designed to prevent DNS responses from revealing other names in the domain. NSEC3 uses the lexicographic order of *hashed* names, instead of their unhashed order. In response to a query without a matching record, NSEC3 returns a signed statement of the hashed names that come just before and just after the hash of the query.

Suppose that the server S has records for `a.cs161.com`, `b.cs161.com`, and `c.cs161.com`, but *not* for `abc.cs161.com`. In addition, assume that `a.cs161.com` hashes to `dee60f2e...`, `b.cs161.com` to `80a4cb36...`, `c.cs161.com` to `c218f96a...`, and `abc.cs161.com` to `99f3e2ba...`.

If R queries S for `abc.cs161.com`, what will S return in response? Describe how R uses this to validate that `abc.cs161.com` does not exist.

(d) The hashes in NSEC3 are computed as a function of the original name plus a *salt* and an *iteration parameter*, as follows:

```
Define H(x) to be the hash of x using the Hash Algorithm selected by
the NSEC3 RR, k to be the number of Iterations, and || to indicate
concatenation. Then define:

   IH(salt, x, 0) = H(x || salt), and

   IH(salt, x, k) = H(IH(salt, x, k-1) || salt), if k > 0

Then the calculated hash of a name is

   IH(salt, name, iterations)
```

The name of the hash function, the salt and the number of iterations are all included in an NSEC3 reply (that is, they are *visible* and assumed to be easily known). All replies from a given server use the same salt value and the same number of iterations.

Suppose an attacker has a list of "names of interest," i.e., names for which they want to know whether the given name is in a particular domain. If the attacker can get all of the NSEC3 responses for that domain, can they determine whether these names exist? If so, sketch how. If not, describe why not.

(e) Why would a domain change their *salt* value in NSEC3 replies?

(f) What is the purpose of the *iteration parameter* in NSEC3 replies?

(g) The specification of NSEC3 also sets an upper bound on the *iteration parameter*. What could happen if that upper bound did not exist?

Problem 4 ***RST Injections*** **(15 points)**

(a) In ONE SENTENCE, explain the purpose of a RST injection attack: that is, what goal does an attacker try to accomplish by launching such an attack?

(b) What information is needed for an attacker to carry out a successful RST injection attack?

(c) Explain under what circumstances an off-path attacker can conduct a successful RST injection attack. If it is impossible for them to do so, explain why they cannot.

(d) Assume Alice switches over to a WPA2-Enterprise connection. Does this protect her from RST attacks? Why or why not?

Problem 5 ***Feedback*** **(0 points)**

Optionally, feel free to include feedback. Whats the single thing we could do to make the class better? Or, what did you find most difficult or confusing from lectures or the rest of class, and what would you like to see explained better?