Raluca Ada Popa
Spring 2018

CS 161
Computer Security

Homework 3

Due: Friday, March 23rd, at 11:59pm

Instructions. This homework is due Friday, March 23rd, at 11:59pm. It *must* be submitted electronically via Gradescope (and not in person, in the drop box, by email, or any other method). This assignment must be done on your own.

Please put your answer to each problem on its own page, in the order that the problems appear. For instance, if your answer to every problem fits on a single page, your solution will be organized as follows:

page 1: your solution to problem 1
page 2: your solution to problem 2
page 3: your solution to problem 3
page 4: your solution to problem 4
page 5: your solution to problem 5

If your solution to problem 3 takes up two pages, your solution would be organized as follows:

page 1: your solution to problem 1
page 2: your solution to problem 2
page 3: first page of your solution to problem 3
page 4: second page of your solution to problem 3
page 5: your solution to problem 4
page 6: your solution to problem 5

Scan your solution to a PDF—or, write it electronically and save it as a PDF. Then, upload it to Gradescope.

Problem 1 ***TLS*** **(20 points)**

Alice wants to communicate with `smallfish.com` using TLS. Assume the browser and server use RSA-based key exchange (not Diffie-Hellman). Each part is independent. Assume that the cipher-suites chosen in the TLS connections **do not** make use of TLS sequence numbers. Assume each potential replay attack involves generating a new TLS connection.

(a) Suppose Alice downloads a buggy version of the FireFox browser that implements TLS incorrectly. The TLS specification says that, during the handshake, the browser should send a random 256-bit number R_B and the server should send a random 256-bit number R_S.[1] Instead of picking R_B randomly, it increments a counter and sends it instead. Which of the following is true?

1. If Alice visits a HTTPS URL, a man-in-the-middle can not replay that HTTP request to the server a second time.
2. If Alice visits the same HTTPS URL twice, when Alice visits that URL the second time a man-in-the-middle can not replay the HTML page that was returned by the server on Alice's first visit.
3. A man-in-the-middle can not compromise Alice's confidentiality (e.g., learn the data she sends over TLS).
4. A man-in-the-middle can not learn the symmetric MAC keys that protect data sent over TLS connections initiated by Alice's browser.
5. None of the above

(b) Alice downloads a patch for her buggy version of FireFox, which fixes the previous problem but introduces another bug. Now, her client generates the pre-master secret based on three items: the current absolute time, the total time Alice's computer has been powered on, and the process-ID of the current FireFox process. Which of the following is true?

1. If Alice visits a HTTPS URL, a man-in-the-middle can not replay that HTTP request to the server a second time.
2. If Alice visits the same HTTPS URL twice, when Alice visits that URL the second time a man-in-the-middle can not replay the HTML page that was returned by the server on Alice's first visit.
3. A man-in-the-middle can not compromise Alice's confidentiality (e.g., learn the data she sends over TLS).
4. A man-in-the-middle can not learn the symmetric MAC keys that protect data sent over TLS connections initiated by Alice's browser.
5. None of the above

[1] Also recall Discussion 6, Q2 here: http://inst.eecs.berkeley.edu/ cs161/sp18/handouts/dis6.pdf

(c) Alice downloads an updated version of FireFox that is finally correct. With her fixed browser, she visits `https://bluefish.com/`. That server's TLS implementation has a bug: instead of picking R_S randomly, it always sends all zeros. Which of the following is true?

1. If Alice visits the HTTPS URL, a man-in-the-middle can not replay that HTTP request to the server a second time.
2. If Alice visits the same HTTPS URL twice, when Alice visits that URL the second time a man-in-the-middle can not replay the HTML page that was returned by the server on Alice's first visit.
3. A man-in-the-middle can not compromise Alice's confidentiality (e.g., learn the data she sends over TLS).
4. A man-in-the-middle can not learn the symmetric MAC keys that protect data sent over TLS connections initiated by Alice's browser.
5. None of the above

Problem 2 ***TCP and LAN*** **(20 points)**

Paul has just opened his laptop and is attempting to connect to the internet to visit `www.moneydancemoves.com`.

(a) Eve sees the DHCP Discover message of the laptop and decides to interfere. What kind of DHCP message can she send at this point and what information does it include? Of the information included, **underline** the pieces of information that will allow Eve to exploit Paul's future internet connections. Assuming you do not know the timing of messages delivered in the local network, will this attack always work? Answer in no more than **two sentences**.

(b) Assume instead Eve does not interfere and Paul successfully runs DHCP to connect to the internet via unencrypted HTTP. Mallory is a man-in-the-middle of Paul's laptop and `www.moneydancemoves.com`. Mallory runs a stateless packet filter that checks each of Paul's TCP packets to see if any given packet contains the phrase "send the money". If it does, she injects a RST packet to sever Paul's TCP connection. Will Mallory always know whether Paul asks `www.moneydancemoves.com` to "send the money"?

(c) Eve and Mallory decide to escalate the situation to their superiors and the NSA gets involved. The NSA can leverage off-path, on-path, and man-in-the-middle attacks against Paul, but they would prefer to use the easiest attack (off-path is easier that on-path, and on-path is easier than man-in-the-middle). For each scenario, which attack will the NSA use (on-path, off-path, or man-in-the-middle)?

(i) The NSA seeks to create a UDP request to the website's server which appears to come from Paul's IP address. The NSA doesn't need to see the reply.

(ii) The NSA seeks to create a TCP connection to the website's server which appears to be from Paul's IP address. The server uses the current time to generate the initial sequence number. The NSA doesn't need to see the reply.

(iii) The NSA seeks to create a TCP connection to the website's server which appears to be from Paul's IP address. The server uses a secure RNG to generate the initial sequence number. The NSA doesn't need to see the reply.

(iv) The NSA seeks to inject content into an existing active TCP connection between Paul and the web server. The NSA knows Paul is paranoid and records his raw traffic, but the NSA does not want Paul to determine that the NSA has modified the traffic.

Problem 3 ***DNS*** **(20 points)**

Outis is a hacker attempting to sabotage CS161 from a cafe. He sees Raluca walk into the cafe and tell Won that she will upload her draft of midterm 2 to the secret TA website, `www.midterm.ta_secrets.com` after she has worked on it for an hour or two. Outis knows that the local DNS server lies on the cafe network and that it may be possible to interfere with its queries. He cleverly sends Raluca a malicious link titled "IMPORTANT PROJECT UPDATE FROM KEYHAN" that she will click immediately. Clicking the link will redirect her to `www.midterm.ta_secrets.com` and cause her computer to generate one DNS query for the secret TA site. Assume that the following subproblems do not build upon each other (no information is cached from a previous attack, etc).

(a) Suppose Outis owns his own website. He can customize its appearance and see all files uploaded to the site. Explain how Outis can eventually obtain Raluca's midterm 2 draft assuming he is able to successfully spoof a DNS response. Use no more than **two sentences**.

(b) Explain why Outis can spoof a DNS response now even though Raluca will likely upload the valuable midterm file in an hour or two. **Answer in a single sentence**.

(c) Outis is lazy and wants to use the easiest network attack possible to spoof the response. An off-path attack is easier than an on-path attack, and an on-path attack is easier than a man-in-the-middle attack. Suppose Outis can only generate a single fake response to the cafe's DNS server query (though his response will reliably reach the server first). What flavor of network attack will Outis use (off-path, on-path, or man-in-the-middle)? Justify your answer using no more than **two sentences**.

(d) Outis realizes that due to cafe security upgrades, he can only perform off-path attacks (he cannot see or intercept network traffic). However Outis' friend Kaminsky has told him about a new attack scheme: Outis upgrades his malware to generate k forged DNS responses that will all arrive before the legitimate response. Assume the DNS query randomizes only transaction ID. Give the probability p that Outis will succeed (Outis succeeds if Raluca accepts any of the spoofed responses as valid).

(e) Outis finishes reading Kaminsky's paper and realizes he can improve his off-path attack. He modifies the link in his message to Raluca ("IMPORTANT PROJECT UPDATE FROM KEYHAN") so that when she clicks it her laptop will generate m requests instead of one. The requests are for URLs: `1.ta_secrets.com`, `2.ta_secrets.com`, ..., `m.ta_secrets.com`. Despite the fact that none of these URLs are valid, Kaminsky has told Outis that if he can correctly spoof the response to a single one of these queries, he will be able to give a malicious IP address to the higher domain *ta_secrets.com* in the "Additional" field of the response and thus by extension control `midterm.ta_secrets.com`[2]! Assuming again that Outis can generate k spoofed responses *per request*, what is the probability, p, that he succeeds?

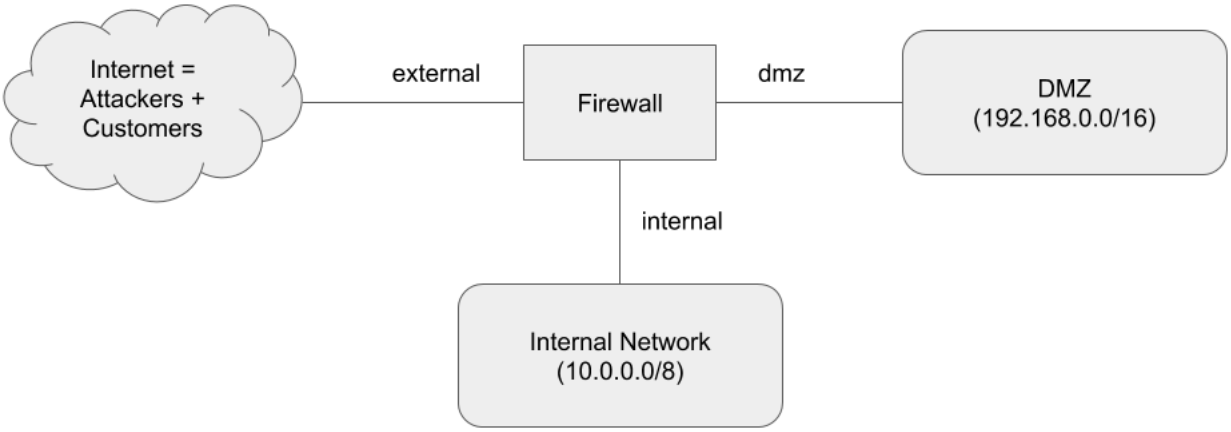
[2]For more details on the Kaminsky attack, see the "Dan's Shenanigans" section of this link

(f) Raluca sees Outis and suspects he is up to something. She switches her laptop to a network with a DNS server that uses DNSSEC for greater security. The resolver's cache is empty and when she types `www.midterm.ta_secrets.com` into her browser the DNSSEC resolver begins to recursively resolve the name. Suppose the resolver has just received the response from the name server at `ta_secrets.com`. Whose public key will it use to verify the response, and whose public key is contained `inside` the response? Assume `midterm.ta_secrets.com` is configured to be its own zone (with corresponding name server).
NOTE: There is no need to reference ZSK or KSK - they are out of scope of this class. Please answer with respect to what was taught in lecture.

(g) As discussed in lecture, DNSSEC name servers must sign responses to requests that do not exist with a cached record of the two closest domain names (this is a form of "off-line" signing called an NSEC record)[3]. Assume the only other subdomain of `ta_secrets.com` aside from `midterm.ta_secrets.com` is a secret subdomain, `thanks_won_park.ta_secrets.com`. When someone visits `thanks_won_park.ta_secrets.com` they are able to change the grades of the 161 students. How might Outis efficiently discover this secret domain? Explain in **two sentences** or less.

[3]Remember that if a server performs "online" signing (every time a request for a nonexistent name x comes in the server signs a message saying x does not exist) then the server is vulnerable to a DoS attack since signing is computationally expensive.

Problem 4 *Firewalls and DMZ* (20 points)



The figure above shows the network configuration of a company. The company cares very much about the information security, and has created a *demilitarized zone* (DMZ) to isolate the internal network from the public facing services hosted in the DMZ.[4]

You are tasked with writing the firewall rules for this scenario. You need to only worry about the TCP traffic. Here are the requirements:

- Deny all traffic, unless otherwise specified.
- Allow inbound web traffic from the Internet to TCP port `80` and TCP port `443` of a web server at `192.168.1.20`, which sits in the DMZ.
- Allow inbound SSH traffic from the Internet to TCP port `22` of an SSH server `192.168.3.40` in the DMZ.
- Allow all traffic from an SSH server `192.168.5.60` in DMZ to the internal network.
- Allow all outbound connection to the Internet from the internal network, except those of social networking sites with IP address blocks of `8.8.0.0/16`, `2.3.0.0/16`, and `56.78.90.0/24`.
- Allow all connections to the DMZ network from the internal network, except to IP addresses `192.168.250.0/24`.

You need to consider three separate sets of firewall rules, one for each of the interfaces (see the figure above). The interfaces are labeled as `external`, `internal` and `dmz`. The internal network is `10.0.0.0/8`; the DMZ network is at `192.168.0.0/16`; everything else is outside. Rules always apply to the packets *arriving* at an interface. For example, rules associated with the `dmz` interface apply to packets sent from the `192.168.0.0/16` network.

Use a format similar to the lecture, but annotated with the interface slightly differently. For example:

[4] Read a bit about DMZ here: https://en.wikipedia.org/wiki/DMZ_(computing)

`internal allow tcp 10.1.2.3:* -> 8.8.8.0/24:80 if ACK set`
specifies that on the firewall's `internal` interface, TCP packets arriving with a source address of `10.1.2.3` and any source port, destined for port `80` of any address in the `/24` block `8.8.8.0`, should be allowed providing they have the `ACK` bit set. Another example, that drops all the traffic from the internal network to the DMZ network, will be written as: `internal drop tcp *:* -> 192.168.0.0/16:*`

For your solution, list a separate ruleset for each interface. You should aim to keep the rules as simple (minimal) as possible. Keep in mind that the rules are applied in sequence, thus the order in which rules are processed matters. (*Hint:* You may want to put something like `<interface> drop tcp *:* -> *:*` as the very last rule of your list depending on the interface; this merely indicates that the default action is `deny`.)

Problem 5 ***DDoS*** **(20 points)**

Note that we talk about real world attacks in this question; we request you to revisit the 'Ethics' section in the online course policies before starting this question.

Let us talk about Distributed Denial of Service attacks.[5] Recently, `github.com` came under a massive DDoS attack at the rate of 1.3Tbps, which you should read about online.[6]. Just days ago, another unnamed service provider came under an even bigger attack, going as high as 1.7Tbps.[7] Using open `memcached` servers, the attackers were able to send bogus traffic to their victims and effectively *amplify* their bandwidth by a factor of as high as 51,000.[8]

The purpose of such DDoS attacks is to fill up the network bandwidth of a victim web-service with bogus traffic. When the victim has a finite network bandwidth, an attacker can prevent legitimate traffic from reaching the victim's web-service. A victim of a DDoS attack is quite helpless; using a firewall or IDS does not help free the upstream bandwidth.

The attack: DDoS attacks could be either at the network level (UDP floods, TCP SYNs), or at the application level (HTTP GET/POST); the `memcached` based amplification attack falls in the network level category. Let us calculate the potential size of a DDoS attack an adversary can launch using these `memcached` servers by *amplification. Before starting, make sure you have skimmed through the posts in the footnotes.*

(a) Imagine an attacker: a bored teenager with a modest Internet connection of 10Mbps (megabits/second). The attacker finds 10,000 open `memcached` servers on the Internet, all of which have very good bandwidth. He also discovers a standard query of size 25 bytes that all the servers respond to; the response is a fixed size of 750KB (kilobytes). Also note that after these high-profile attacks, all the `memcached` servers started using a rate limit of 10 requests/second per server for a given client IP address.

Given these constraints, what is the size of DDoS that this attacker can launch against `smallfish.com`, the website of a small local game-store that maps to an IP address `1.0.0.1`. Please provide the traffic-volume seen by the victim in units of Tbps (terabit/second), and provide a 2-3 sentence summary of how you achieved this number. Assume that 1 terabit = 10^{12} bits.

(b) Imagine the same situation for the attacker above, except that the bored teenager decided to use his school's Internet for his nefarious actions. The school Internet is

[5] Here is a fun little analogy for what it feels like to be under a DDoS attack (a scene from the movie Harry Potter and the Sorcerer's Stone): https://www.youtube.com/watch?v=yQIFkMlDF4M&t=150s

[6] Here is a good summary: https://www.wired.com/story/github-ddos-memcached/

[7] https://arstechnica.com/information-technology/2018/03/us-service-provider-survives-the-biggest-recorded-ddos-in-history/

[8] We do not require you to understand *all* the details, but please skim through the following post about `memcached`'s role in DDoS attacks; we expect you to have at least a high-level understanding of the actual attack: https://blog.cloudflare.com/memcrashed-major-amplification-attacks-from-port-11211/.

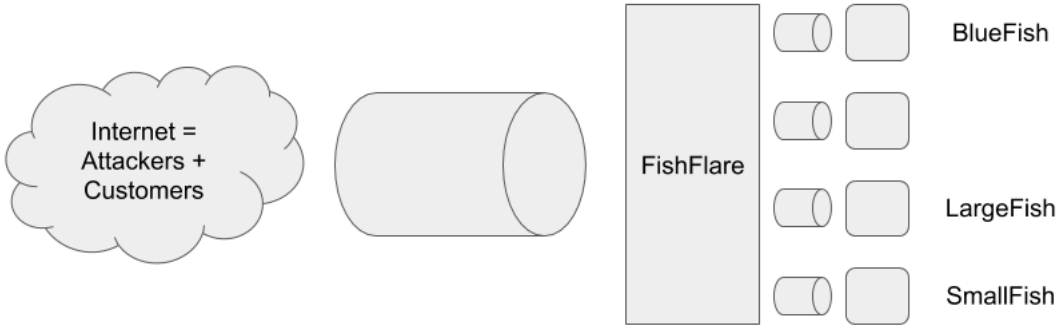
100Mbps (megabits/second).

Given the constraints, what is the size of DDoS that this attacker can launch against `smallfish.com`. Please provide the traffic-volume seen by the victim in units of Tbps (terabit/second), and provide a 2-3 sentence summary of how you achieved this number.

(c) The attacker is still at school, and enjoying his 100 Mbps Internet connectivity. Sufficiently content with destroying the `smallfish.com`'s business, he now wants to go after `largefish.com`. The major difference between `smallfish.com` and `largefish.com` is that the latter is a much larger and heavy traffic website. In order to process the traffic, `largefish.com` uses 10 web-servers with IP addresses in the range `2.0.0.1-2.0.0.10`, and uses a DNS-based load-balancing scheme.[9]

Given the constraints, what is the size of DDoS that this attacker can launch against `largefish.com`. Please provide the *total* traffic-volume seen by the victim `largefish.com` in units of Tbps (terabit/second), and provide a 2-3 sentence summary of how you achieved this number.

(d) Let us now look at some of the mitigations of these DDoS attacks. To protect against a potential DDoS of the magnitude we analyzed above, website owners have to acquire significant extra bandwidth (by orders of magnitude), which is just not practical. However, the risk of a *specific* website being under attack at a given time is relative small. This is a perfect fit for an insurance scheme, *i.e.* risk distribution across a number of participants.



In order to provide protection against such large scale attacks, a new company `FishFlare` provides a DoS mitigation service to its customers. This is how it works: a customer, `BlueFish`, points the DNS entry for `bluefish.com` to `FishFlare`'s servers, and tells `FishFlare` about the real IP address that host `bluefish.com`'s content. This way, `FishFlare` sits between the customers of `BlueFish` and actual

[9]This is a common load-balancing scheme used widely. In this example, the DNS lookup for `largefish.com` can return a randomly picked IP-address from the 10 choices. A particular client has a 10% chance of ending up with a given server, but it doesn't matter because all the servers host exactly same content. This allows `largefish.com` to scale linearly by just putting more servers.

servers hosting content for `bluefish.com` (see the figure above).[10]

The hope is once `FishFlare` acquires enough customers, it can also acquire large enough bandwidth to withstand very large DDoS attacks with the cost amortized over a large number of customers. Since all the traffic passes through `FishFlare`, it can *scrub*-away the bogus traffic and only send cleaned-up traffic to actual `BlueFish` infrastructure.

How can `FishFlare` cleanup the `memcached`-based attack on the website of one of its customers? In no more than 2 sentences, describe an efficient strategy that works at line rate and is effective (*i.e.* very few, if any, false positives or false negatives).

(e) Imagine that the idea of a DoS mitigation scheme became quite popular, and `FishFlare` is now quite successful. Now they have expanded their business in protection against not just network-level DoS attacks (such as the `memcached` attack), but also application-level attacks. More specifically, `FishFlare` acts as an application-level firewall and filters out malicious HTTP requests that attempt to exploit potential vulnerabilities in the web-applications of `BlueFish`.

`FishFlare` employs the most advanced analytics schemes to predict whether a particular HTTP request is a malicious request by an attacker, or a benign request by a real human. They are updating their analytics scheme and are deciding between the following:

1. Scheme 1 has a false positive rate of 5% and a false negative rate of 1%
2. Scheme 2 has a false positive rate of 1% and a false negative rate of 5%
3. Scheme 3:
   Run both given schemes in series, one after the other (run requests through scheme 1, and if they are marked benign pass them to scheme 2). The two schemes are not fully independent detectors: the false negative rate decreases to 0.1%, while the false positive rate increases to 5.5%

`FishFlare` performs a cost analysis and determines that each time an automated HTTP request by an attacker passes through undetected it costs $100. Each time a benign user request is incorrectly flagged as an attack it costs $3. The new schemes will not cost anything when a request is correctly categorized as malicious or benign. `FishFlare` estimates that 2 in every $10,000$ requests are malicious requests. Which detection scheme should `FishFlare` use? Which factor ends up mattering the most in this scenario, false positive or false negative rates? Show your calculations (Hint: which scheme would we *expect* to be more cost effective?) Please choose only from

[10] Note that this still leaves `BlueFish` vulnerable, since anyone who can figure out the real IP address can launch DDoS directly against `BlueFish` without going through `FishFlare`. This DNS based technique is merely one solution that fits small-scale businesses. The actual techniques used by large corporations are slightly more involved and require a bit nuanced knowledge of how the Internet routing works (specifically BGP), and context (whether a victim can seek help from its ISP in some filtering). You may learn about this more in a networking class; we ignore the details here to get the high-level idea across.

the 3 options given (do not, for instance, put "no option" or "option 1 in parallel with option 2" etc).