

Due: Friday, 22 March 2019, at 11:59pm

Instructions. This homework is due **Friday, 22 March 2019, at 11:59pm**.

You may use any of the following methods to complete your homework:

- Use the LaTeX template provided on Piazza to fill out your responses; or
- Use Adobe Acrobat to fill in this fillable PDF; or
- Print this paper out and handwrite your solutions, but **please make sure your responses do not overflow the box provided** before submitting to ensure that you get full credit for your response.

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 are true? **Select all that apply.**

- ☒ If Alice visits a HTTPS URL, a man-in-the-middle can not replay that HTTP request to the server a second time.
- ☒ 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.
- ☒ A man-in-the-middle can not compromise Alice's confidentiality (e.g., learn the data she sends over TLS).
- ☒ A man-in-the-middle can not learn the symmetric MAC keys that protect data sent over TLS connections initiated by Alice's browser.
- ☐ None of the above

¹Also recall Discussion 7, Q2 here: <https://inst.eecs.berkeley.edu/~cs161/sp19/sections/07.pdf>

- (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 are true? **Select all that apply.**

☒ If Alice visits a HTTPS URL, a man-in-the-middle can not replay that HTTP request to the server a second time.

☒ 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.

☐ A man-in-the-middle can not compromise Alice's confidentiality (e.g., learn the data she sends over TLS).

☐ A man-in-the-middle can not learn the symmetric MAC keys that protect data sent over TLS connections initiated by Alice's browser.

☐ None of the above

- (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 are true? **Select all that apply.**

☐ If Alice visits a HTTPS URL, a man-in-the-middle can not replay that HTTP request to the server a second time.

☒ 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.

☒ A man-in-the-middle can not compromise Alice's confidentiality (e.g., learn the data she sends over TLS).

☒ A man-in-the-middle can not learn the symmetric MAC keys that protect data sent over TLS connections initiated by Alice's browser.

☐ None of the above

Problem 2 TCP**(10 points)**

Aisling has just opened her laptop and is attempting to connect to the Internet to visit `www.randomkittengenerator.com`.

Aoife, who works for the local Three-Letter Agency (TLA), is determined to interfere with Aisling's connection. The TLA (and, thus, Aoife) can leverage off-path, on-path, and man-in-the-middle attacks against Aisling, but they would prefer to use the easiest attack. Recall that a man-in-the-middle can both observe as well as intercept traffic; an on-path attacker can observe traffic but not intercept it; and an off-path attacker can neither observe nor intercept traffic. Hence, an off-path attack is easier to launch than on-path, and on-path is easier to launch than man-in-the-middle.

For each scenario, which attack will Aoife use (on-path, off-path, or man-in-the-middle)?

- (a) Aoife seeks to create a UDP request to the website's server which appears to come from Aisling's IP address. Aoife doesn't need to see the reply.

☒ Off-path

☐ On-path

☐ Man-in-the-middle

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

☒ Off-path

☐ On-path

☐ Man-in-the-middle

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

☐ Off-path

☒ On-path

☐ Man-in-the-middle

- (d) Aoife seeks to inject content into an existing active TCP connection between Aisling and the web server. Aoife knows Aisling is paranoid and records her raw traffic, but Aoife does not want Aisling to determine that Aoife has modified the traffic.

☐ Off-path

☒ On-path

☐ Man-in-the-middle

Problem 3 *DNS*

(20 points)

Okoro is a hacker attempting to sabotage CS161 from a cafe. He sees Raluca walk into the cafe and tell a TA 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. Okoro 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 NICK” 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 Okoro owns his own website. He can customize its appearance and see all files uploaded to the site. Explain how Okoro can eventually obtain Raluca’s midterm 2 draft assuming he is able to successfully spoof a DNS response. Use no more than **two sentences**.

He can spoof the DNS response, so that `www.midterm.ta.secrets.com` will be linked to his faked website. After Raluca uploading her draft into this faked website, he can obtain it.

- (b) Explain why Okoro 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.**

Local DNS resolver will cache DNS response.

- (c) Okoro 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 Okoro 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 Okoro use (off-path, on-path, or man-in-the-middle)? Justify your answer using no more than **two sentences**.

He can use on-path attack.
Because he has only one shot, so he has to know the accurate transaction ID and UPD port to generate a valid fake DNS response.

- (d) Okoro realizes that due to cafe security upgrades, he can only perform off-path attacks (he cannot see or intercept network traffic). However Okoro's friend Kaminsky has told him about a new attack scheme: Okoro 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 Okoro will succeed (Okoro succeeds if Raluca accepts any of the spoofed responses as valid).

Transaction ID is 16-bits long, therefore, $p = \frac{k}{2^{16}}$.

- (e) Okoro 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 NICK") 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 Okoro 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 Okoro can generate k spoofed responses *per request*, what is the probability, p , that he succeeds?

He has $\frac{k}{2^{16}}$ chance to win in a single URL. Overall, he can success if he didn't mess up m times. Therefore, $p = 1 - (1 - \frac{k}{2^{16}})^m$.

- (f) 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_nick.ta_secrets.com`.

When someone visits `thanks_nick.ta_secrets.com` they are able to change the grades of the 161 students. How might Okoro efficiently discover this secret domain? Explain in **two sentences** or less.

Make a DNS request to ask NSEC record for `midterm.ta_secrets.com`. Since the only other subdomain is `thanks_nick.ta_secrets.com`, he can get info about it.

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

³Remember 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 *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.⁴ Recently, `github.com` came under a massive DDoS attack at the rate of 1.3Tbps, which you should read about online.⁵ Since then, another unnamed service provider came under an even bigger attack, going as high as 1.7Tbps.⁶ 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 51,000.⁷

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 an intrusion detection system 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 5Mbps (megabits/second). The attacker finds 5,000 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 600KB (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.

⁴Here 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=yQIFkMIDF4M&t=150s>

⁵Here is a good summary: <https://www.wired.com/story/github-ddos-memcached/>

⁶<https://arstechnica.com/information-technology/2018/03/us-service-provider-survives-the-biggest-recorded-ddos-in-history/>

⁷We 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/>.

$$\frac{600 \text{ KB}}{25 \text{ B}} \times 5 \text{ Mbps} = 0.12 \text{ Tbps}$$

Memcached server can amplify traffic by factor $\frac{600 \text{ KB}}{25 \text{ B}} = 24000$. Therefore the overall traffic victim will see is $24000 \times 5 \text{ Mbps}$.

- (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 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.

$$10 \times 5000 \times 600 \times 1000 \times 8 = 2.4 \times 10^{11} \text{ bps} = 0.24 \text{ Tbps}$$

There're 5000 open memcached servers and each of them can respond to a given IP address 10 times per second. And each response is 600KB, which is $600 \times 1000 \times 8$ bits.

- (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.⁸

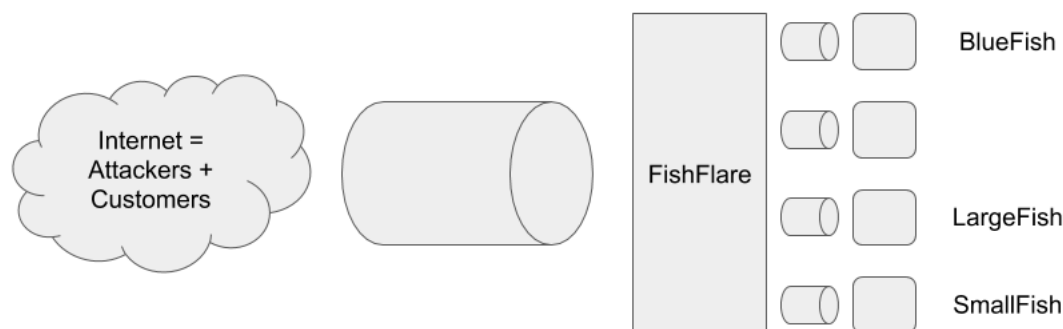
Given the constraints, what is the size of DDoS that this attacker can launch against **largefish.com**. Please provide the *total* traffic-volume seen by the victim **largefish.com** in units of Tbps (terabit/second), and provide a 2-3 sentence summary of how you achieved this number.

$$10 \times 0.24 \text{ Tbps} = 2.4 \text{ Tbps}$$

Since there're 10 different IP addresses belongs to **largefish.com**, the teenage can use the same DDoS attack against each of these addresses.

⁸This 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.

- (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 servers hosting content for `bluefish.com` (see the figure above).⁹

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).

⁹Note 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.

Block all inbound UDP packets from source port 11211.

Problem 5 *Reconnaissance Attacks***(15 points)**

The IP packet header contains a 16-bit ID field that is used for assembling packet fragments.¹⁰ The protocol standard states that the ID field should differ between different packets sent by a source to a given destination.¹¹ A common method that hosts use to implement the ID field is to maintain a single counter that the host increments by one for every packet it sends, regardless of to which destination it sends it. The host sets the ID field in each packet it sends to the current value of the counter. Since with this implementation the host uses a single counter for all of its connections, we say that the host implements a *global ID field*.

- (a) Suppose a host \mathcal{P} implements a global ID field. Suppose further that \mathcal{P} responds to ICMP ping requests. You control some other host \mathcal{A} . How can you test if \mathcal{P} sent a packet to anyone (other than \mathcal{A}) within a certain one minute window? Can you use that same method to test if \mathcal{P} received a packet from anyone within a certain one minute window? You are allowed to send your own packets to \mathcal{P} .

Let \mathcal{A} send a ICMP ping request to host \mathcal{P} before and after this one minute window. Check the ID field in response packets. If they're differed by 1, \mathcal{P} didn't send packet to anyone else. Otherwise \mathcal{P} did sent packets to someone. This method cannot test whether \mathcal{P} receive a packet from anyone.

- (b) Your goal now is to test whether a victim host \mathcal{V} is running a server that accepts connections to TCP port n (that is, test if \mathcal{V} is listening on TCP port n). You wish to hide the identity of your machine \mathcal{A} . Hence, \mathcal{A} cannot directly send a packet to \mathcal{V} , unless that packet contains a spoofed source IP address. Explain how to use \mathcal{P} to do this, assuming that \mathcal{P} is quiescent during this test. If \mathcal{P} is not quiescent during the test, will this method still work? Why or why not?

HINT: recall the following facts about TCP.

- A host that receives a SYN packet to an open port n sends back a SYN/ACK response to the source address.
- A host that receives a SYN packet to a closed port n sends back a RST packet to the source address.
- A host that receives a SYN/ACK packet that it is not expecting sends back a RST packet to the source address.
- A host that receives a RST packet sends back no response.

¹⁰ For now, don't worry about how such fragments work. For this problem, we only care that this field exists and how hosts set it in the packets they send.

¹¹ Clearly, if a host sends more than 2^{16} packets, the field will necessarily repeat. That consideration doesn't matter for this problem, either!

Send a SYN packet to \mathcal{V} with a spoofed source IP address (source address is \mathcal{P} 's). Then check whether \mathcal{P} send a packet using the method mentioned previously. If \mathcal{P} did send, \mathcal{V} is listening on TCP port n . Otherwise, it didn't. This method works because \mathcal{V} will send a SYN/ACK packet to \mathcal{P} , and \mathcal{P} will respond with a RST packet.

If \mathcal{P} is not quiescent during the test, this method doesn't work anymore. Because we cannot determine whether \mathcal{P} send a packet to \mathcal{V} or someone else.

- (c) How would you change \mathcal{P} to stop these attacks? You are not allowed to modify the TCP/IP protocol or the services running on \mathcal{P} . You may only modify the *implementation* of TCP/IP on \mathcal{P} .

Let \mathcal{P} maintain a counter for each pair of source address and destination address.