Popa & Wagner
Spring 2016

CS 161
Computer Security

Homework 4

Due: Monday, May 2nd, at 11:59pm

Instructions. This homework is due Monday, May 2nd, 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 5: your optional feedback ("problem 4")

If your solution to problems 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 optional feedback ("problem 4")

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

Problem 1 ***Attacking home routers*** **(40 points)**

Cheaporouter builds a wireless DSL router that ISPs often ship to their customers. It has an administrative interface that lets you change lots of configuration options by accessing its web server (which is open to the world):

URL	**Purpose**
`http://yourrouter/login?u=daw&p=mypass`	to login
`http://yourrouter/set?ssid=SkyNet`	set the name of the wireless network
`http://yourrouter/set?wifichannel=3`	to set the WiFi channel
`http://yourrouter/set?time=11:36AM`	set the date/time
`http://yourrouter/set?dns=1.2.3.4`	set the primary DNS server
`http://yourrouter/set?speed=1.5Mbps`	set the link speed
`http://yourrouter/set?dhcp=on`	enable DHCP
`http://yourrouter/set?logging=on`	to enable logging
`http://yourrouter/set?report=24hr`	set how often the router reports status

You have to log in using the correct username and password for that router before setting any configuration option; logging in sets a session cookie on your browser, and then subsequent requests to the router are allowed to set config options. Unfortunately, the default username and password is `admin`/`password`, and many users do not change the default.

(a) Explain how an attacker anywhere on the Internet can attack Cheaporouter users who haven't changed their default password, to steal all their subsequent search queries to Google and redirect them to the `HackrzSrch.com` search engine (thus getting the ad revenue for themselves). Your method should require only a one-time attack on the router, and should not assume the existence of any implementation bugs in the router's software. Assume the users' Google search queries are sent via http, not https.

(b) Cheaporouter hears about this flaw, and they decide to modify their routers to prevent this attack. On the new routers, the web server providing the administrative interface will now respond only to connections from the internal home network (e.g., from machines on its local wireless network or local machines connected via Ethernet to the router), at the IP address 192.168.0.1. The router will not respond to connections coming in over the Internet connection (coming in over DSL/cable) to its administrative interface. By default, the router ships with its wireless connection enabled and configured for open wireless, with no password or access control. Explain how an attacker who drives by the house of someone who has bought one of these new Cheaporouter's and is using it without changing any default setting, can mount the attack you described in part (a).

(c) Cheaporouter decides that the new default will be to leave wireless disabled. Imagine that Joe is using their newest router, with all the defaults left intact, and he has several home computers hooked up to his Cheaporouter. He allows a friend of his to connect her laptop to his home network; unfortunately, it's infected with some malware. Explain how that malware could exploit features in the Cheaporouter to steal all search engine traffic coming from all of Joe's home computers.

(d) Sam is using Cheaporouter's newest router, with all the defaults. Sam often visits random third-party websites. Suppose the attacker controls a website (dancing-bears.com) that Sam happens to visit. Explain how the attacker can exploit features on Sam's Cheaporouter to steal all of Sam's subsequent search engine traffic subsequently coming from Sam's computer. Assume that Sam uses a fully-patched web browser, and the attacker doesn't know any exploits for Sam's browser, so the attacker can't get malware onto Sam's machine.

Problem 2 ***Denial-of-service via email*** **(20 points)**

One day you try to email five of your friends at Stanford your giant, high-definition torrented file of the latest Game of Thrones episode to their `stanford.edu` addresses. Unfortunately you typed two of their email addresses incorrectly so you received two Non-delivery Notification (NDN) messages that included both the original text from the email you sent and copies of the attachment.

(a) Knowing this, how could you launch an amplified denial of service attack against some poor unsuspecting soul via email?

(b) How could the developer of the mail server mitigate this issue?

Problem 3 ***Batch signatures*** **(40 points)**

Jean works for a startup that sends real-time stock data, signed, to its clients. They need to sign millions of messages per second. Jean has benchmarked the signing algorithm and found that their server can sign 10,000 messages/second (i.e., each signature takes 0.1ms to compute). This has her manager pretty worried: it sounds like they're going to have to buy hundreds of servers just to sign all of these messages, which sounds expensive.

Jean comes up with a clever idea: the server will collect a batch of ten messages $m_1, \ldots, m_{10}$, then compute a single signature on the value

$$x = H(H(m_1) \mid\mid H(m_2) \mid\mid \cdots \mid\mid H(m_{10}))$$

(this is intended to essentially prove that all of $m_1, \ldots, m_{10}$ were signed), then do the same on the next ten messages, and so on. Jean has the intuition that the one signature $\text{Sign}(x)$ should be essentially as good as ten signatures on the ten messages $m_1, \ldots, m_{10}$. Help her work out the details.

(a) Suppose the startup needs to send m_i to a client (for some i with $1 \leq i \leq 10$), so the client can verify m_i was included in the batch of signed messages. Jean initially suggests they should send $m_i, x, \text{Sign}(x)$ to the client, where x is as defined above. Jean's initial suggestion is on the right track, but something is missing: what else does the startup need to send to the client, so the client can verify that m_i was signed?

(b) With your scheme, how can the client verify that m_i was signed (included in the batch of signed messages)? You can assume that the client knows the startup's public key.

(c) Suppose H is a collision-resistant hash function and $\text{Sign}(\cdot)$ is an existentially-unforgeable digital signature scheme. Explain why your scheme is secure, i.e., why an adversary can't fool a client into thinking m' was signed when it actually wasn't.

(d) How many messages can be signed per second, using a single server, with Jean's scheme? You can assume that hashing is so incredibly fast that it's essentially free.

(e) How many bytes of additional data need to be sent to the client with each message, in total? Assume that each signature is 32 bytes long and H outputs 32-byte hash digests.

(f) Describe how to improve Jean's scheme, so a single server can sign 10,000,000 messages/second—yet when we send a message m_i to a client, we only need to send ≤ 1000 bytes of additional data to enable the client to verify m_i was signed. Make the same assumptions as before (signatures and hash digests are 32 bytes long; hashing is so fast that you don't need to count the time it takes to compute the

hash of a value). How does your scheme work? How many bytes of additional data need to be sent with each message?

(The obvious solution is to generalize Jean's technique, changing the 10 to 1000. However, then we'll need about 32,000 bytes of additional data to be sent with each message—too much. So you'll need something cleverer.)

Problem 4 ***Feedback*** **(0 points)**

Optionally, feel free to include feedback. What's 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?