Wagner Spring 2014 — CS 161 Computer Security — Homework 2

Due: Tuesday April 1, at 11:59pm

Instructions. This homework is due Tuesday April 1, at 11:59pm. It *must* be submitted electronically via Pandagrader (and not in person, in the drop box, by email, or any other method). Realistically, you should treat the deadline as 10:59pm; the 11:59pm deadline is strict, and Pandagrader may be overloaded immediately before it. Therefore, we *strongly* recommend you upload your solution by 10:59pm.

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
page 6: your solution to problem 6
page 7: your optional feedback, or a blank page ("problem 7")

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
page 7: your solution to problem 6
page 8: your optional feedback, or a blank page ("problem 7")

Scan your solution to a PDF—or, write it electronically and save it as a PDF. Then, upload it to Pandagrader. You must submit it as a PDF, not a series of images; you will receive no credit if you submit it as images.

Make sure to select pages for each question after uploading in Pandagrader. If you fail to complete the process (of page selection) so that your PDF does not render properly, your homework will be marked as late and therefore not accepted. If you have no response to the optional feedback, please attach a blank page at the end of the PDF and select that page for problem 7.

Problem 1 ***Attacking home routers*** **(12 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.

(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 ***Kaminsky attacks*** **(14 points)**

This question considers attacks against a victim, where the attacker's goal is get the victim to accept a spoofed DNS response. In one iteration of the Kaminsky attack, the attacker forces the victim to make a single DNS query and then sends the victim k forged packets containing spoofed responses to that query, each with a different guess at the transaction ID of that query. Assume that all k of those packets will arrive before the legitimate response from the legitimate DNS server. Also assume that the victim's DNS software randomizes the transaction ID in all DNS queries it sends, but it does not randomize the UDP source port or use any other defenses against the Kaminsky attack.

In each part, show your calculation for how you got the formula/number that the question asks for, and circle your final formula/number.

(a) Give a formula for the probability p that the attacker succeeds in the first iteration, as a function of k. (The attacker succeeds if the victim accepts any of the spoofed responses as valid.)

(b) Suppose the attacker performs m iterations of the attack. Give a formula for the probability q that the attacker succeeds in at least one of these iterations, as a function of k and m.

(c) Suppose $k = 64$. How large does m need to be, to have a 70% probability of success?

Problem 3 ***DNS spoofing and the birthday paradox*** **(18 points)**

Now let's consider an improvement of the Kaminsky attack, based upon the birthday paradox. Consider an attack where the attacker causes the victim to make n DNS queries, all for the same hostname. (Assume that the attacker can trigger the victim to make multiple concurrent DNS queries, all for the same hostname.) Next, the attacker sends k forged packets containing spoofed responses to that query, each with a different transaction ID. Assume that all k of these forged packets will arrive before the first legitimate response from the legitimate DNS server.

In parts (a)–(c), assume that the victim's DNS software randomizes the transaction ID in all DNS queries it sends, but it does not randomize the UDP source port or use any other defenses against the Kaminsky attack. In each part, show your calculation for how you got the formula/number that the question asks for, and circle your final formula/number.

(a) Give a formula for the probability p that the attacker succeeds, as a function of k, n. Note that the attacker succeeds if any of the spoofed responses is accepted by

the victim, i.e., if any of the k spoofed responses has a transaction ID that matches any of the transaction IDs of the n outstanding DNS queries.

(b) Suppose we insist that $k = n$. How large does n need to be, for the attacker to have a 70% probability of success?

(c) Suppose that the attacker can only get 64 spoofed responses to arrive at the victim before the legitimate DNS server's first response arrives, so the attacker is forced to set $k = 64$. What is the smallest value of n that gives the attacker a 70% probability of success?

(d) Now suppose the victim upgrades her DNS software to include UDP port randomization, so each query uses a random transaction ID and a random 16-bit UDP source port. Also assume that in a single iteration of the attack, the attacker can trigger the victim to send at most $n = 256$ outstanding queries at once, and the attacker can send $k = 256$ spoofed responses to the victim (all of which will arrive before the legitimate DNS server's first response). What is the attacker's success probability in a single iteration? Assuming that each iteration takes one second and that the attacker performs enough iterations to ensure a 70% probability of success, how long will the whole attack take in total?

Problem 4 ***Denial-of-service on the web*** **(12 points)**

Your friend has just launched `SiteTester.com`, a cool web service that helps website developers see how their web site will look when rendered with Chrome vs. how it will look in Internet Explorer.

The service is pretty simple. If you visit a URL like `http://sitetester.com/?u=http://berkeley.edu/`, the SiteTester server launches a process running the Chrome browser, loads `http://berkeley.edu/` in Chrome, and takes a screenshot of the Chrome window after the site finishes loading. In parallel, SiteTester starts up Internet Explorer, loads `http://berkeley.edu/` in Internet Explorer, and takes a screenshot of Internet Explorer after the page loads. After both screenshots are available, the SiteTester server serves you a dynamically-generated HTML document that shows both screenshots side-by-side. (Note that, while the SiteTester is handling this HTTP request from the user, it will to issue two separate HTTP requests to `berkeley.edu`, one for each browser.) The SiteTester service can be used on any web page you specify; everything after the `?u=` is treated as a URL and loaded into both browsers. This makes SiteTester very useful to web developers for testing how portable their website is.

How could an attacker mount a denial-of-service attack against the SiteTester server, simply by visiting a single URL? Show in your answer the malicious URL that causes so much trouble. You can assume the SiteTester developers haven't taken any special precautions against denial of service.

Problem 5 ***Denial-of-service via email*** **(14 points)**

One day you try to email five of your friends at Stanford your giant, high-definition torrented file of the latest Glee 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 6 ***Abusing ARP*** **(30 points)**

You are an evil cs161 student who wants everyone else in the class to fail so you can get an A+. One day you walk into La Burrita to get some lunch when you see Rohin sitting in the corner, happily munching on nachos and answering everyone's Piazza questions. You soon realize that if you don't do anything to stop Rohin, there will be no hope of the rest of the class failing! You decide that you must find a way to impersonate Rohin on Piazza and answer questions incorrectly in order to lead other students astray. Luckily you've been paying attention in lecture and know a few things about network attacks that you could launch against Rohin.

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

La Burrita is using a Wifi network with encryption that prevents you from eavesdropping on traffic not intended for you (i.e., you can only see packets sent to your machine and broadcast packets). Therefore, you're going to need to exploit other network protocols. In particular, in parts (a)–(b), your attack must involve exploiting ARP; in parts (c)–(d), your attack must involve exploiting the router's routing table.

ARP is a protocol to help us discover the link-layer address (e.g., Ethernet address or Wifi address) of another machine, when we know the IP address of that machine. It works like this. Suppose my machine wants to send a packet to IP address `1.2.3.4` on the local Wifi network, but it doesn't know the Wifi address of that computer. My machine broadcasts an ARP request, which asks "What is the Wifi address of the computer with IP address `1.2.3.4`?" When the Wifi router 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 Wifi address `00:1A:AA:BB:CC:DD`." The ARP response is sent 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.

You may assume for this problem that Piazza figures out which user is posting by using cookies, but does nothing else. Also, Piazza uses `http` and sends everything in plaintext.

(a) If you could force a reset of La Burrita's network, how could you arrange to receive all of the traffic that Rohin's browser sends to `piazza.com`? How could you use this information to impersonate Rohin on Piazza and post bogus answers?

(b) Fearing Rohin's counter attack, you need to make sure that Rohin doesn't figure out what you are up to. In particular, you want to modify the data that Piazza sends to Rohin, so Rohin doesn't notice your bogus answers when he views the site.

How can you extend the attack in part (a) to spoof Piazza's responses so Rohin won't figure out your diabolical plan?

(c) In order to launch the attack you planned in parts (a) and (b), you go in search of La Burrita's router. Just as you are about to reboot it, the manager catches you! You manage to escape this awkward situation by claiming you were just looking for the bathroom, but now your plan is foiled. Luckily, you were able to catch a glimpse of the router's brand and you happen to know that this particular brand has a vulnerability that allows you to modify the routing table. (The routing table is a data structure that essentially indicates, for each possible IP address, where to send packets destined for that IP address: e.g., they can be sent out over the router's Internet link, or they can be sent to a particular host on the local Wifi network with a specific link-layer Wifi address.)

You also run the `dig` command in your terminal and find out that Piazza has many different IP addresses for its many servers. You know that Rohin's browser will be using the first IP address that dig returned. How can you impersonate Rohin on Piazza now? (For this part, you do not need to worry about tricking Rohin into thinking everything is fine.)

Dig output:

```
www.piazza.com: 23.23.249.136
www.piazza.com: 23.23.249.137
www.piazza.com: 23.23.249.138
www.piazza.com: 23.23.249.139
```

(d) Again, you want to prevent Rohin from figuring out your brilliant scheme. To do this, we must stop any packets from Piazza from getting to Rohin, as they will reveal your bogus answers to him. If you succeed, Rohin will assume that Piazza is down or something, and you will be safe. What should you do?

(e) Now suppose La Burrita's Wifi router wasn't using any encryption, so you could eavesdrop on all packets sent on La Burrita's Wifi network, broadcast or not. You fire up Wireshark. Unfortunately, by the time you got to La Burrita, Rohin had already logged into Piazza, so you didn't capture his password. Fortunately, you attended lecture so you remember some other attacks. Describe how you could use your ability to eavesdrop to enable you to log into Rohin's account on Piazza, without sending any forged packets on the La Burrita Wifi network.

(Guessing Rohin's password on Piazza isn't gonna work; knowing him, he has probably chosen some 20-character password. Trying to get malware onto Rohin's laptop isn't gonna work, either: he is too wily for that. You must find an attack that takes advantage of your ability to eavesdrop on the TCP connection between Rohin's laptop and Piazza.)

Problem 7 ***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?