Paxson Spring 2017 — CS 161 Computer Security — Homework 4

Due: Monday, March 20th, at 11:59pm

(Version 1.0: released Monday March 13th)

Instructions. This homework is due Monday, March 20th, at 11:59pm. You *must* submit this homework electronically via Gradescope (not by any other method). When submitting to Gradescope, *for each question* your answer should either be a separate file per question, or a single file with each question's answer on a separate page. This assignment must be done on your own.

Problem 1 ***Attacking home routers*** **(15 points)**

CheepO'Router 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=vp&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 CheepO'Router 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.

Solution: The attacker can log in using the default password, and then change the primary DNS server to make all DNS requests go through a DNS server controlled by the attacker. The attacker's DNS server could then respond to DNS requests for `google.com` with the IP address of the server that hosts `HackrzSrch.com`.

Put another way, the attacker visits these two URLs, in order:

```
http://yourrouter/login?u=admin&p=password
http://yourrouter/set?dns=6.6.6.6
```

where `yourrouter` should be replaced with the IP address of the CheepO'Router, and `6.6.6.6` should be replaced with the IP address of a DNS server that the attacker controls.

(b) CheepO'Router 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) to the router's permanent IP address of 192.168.250.1. 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 CheepO'Router's and is using it without changing any default setting, can mount the attack you described in part (a).

Solution: The attacker can connect to the wireless network, which does not require a password, and then visit the router's web interface at 192.168.250.1 in their web browser. They can then perform the above attack: log in with the default credentials and change the primary DNS server to point to a malicious one.

(c) CheepO'Router 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 via wired Ethernet to his CheepO'Router. 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 CheepO'Router to steal all search engine traffic coming from all of Joe's home computers.

Solution: The malware can force the laptop to connect to the CheepO'Router administrative interface and perform the same attack as above.

(d) Sam is using CheepO'Router's newest router, with all the defaults. Sam often visits random third-party websites. Suppose the attacker controls a website (dancingbears.com) that Sam happens to visit. Explain how the attacker can exploit features on Sam's CheepO'Router to steal all of Sam's subsequent search engine traffic. 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.

Solution: The third-party website can mount a CSRF attack against the vulnerable administrative interface in order to force Sam's computer to perform actions provided by the interface. In particular, consider what happens if the web site contains the following HTML fragment:

```
<img src="http://192.168.250.1/login?u=admin&p=password" />
<img src="http://192.168.250.1/set?dns=6.6.6.6" />
```

This will cause Sam's browser to log in and change the primary DNS server for his router to a malicious DNS server run by the attacker.

Comment: Congratulations, you've just re-discovered an attack that was apparently good enough to get coverage in the newspapers.

In 2007, researchers at Symantec first warned about this attack and discovered that many home routers were vulnerable to it, including routers from Linksys, D-Link, Belkin, Netgear, and Cisco. (After the attack was discovered, Cisco listed 77 of their router models as vulnerable.) The Symantec folks wrote a paper on the subject: http://www.symantec.com/avcenter/reference/Driveby_Pharming.pdf, and their work got featured in the press. Incidentally, the Symantec folks estimated that about 50% of home users were vulnerable to this attack, so it was quite a significant vulnerability.

Comment: Problem 5 was inspired by vulnerabilities in Dave Wagner's home DSL router, a run-of-the-mill Netgear DSL router that happens to be vulnerable to the attacks from parts (b) and (c), and (with slight modifications) to the attack from part (d) as well. These vulnerabilities are widespread in many home routers. It's not just CheepO'Router—these issues have affected routers built by essentially every major vendor!

Problem 2 *Reconnaissance Attacks* (15 points)

The IP packet header contains a 16-bit ID field that is used for assembling packet fragments.[1] The protocol standard states that the ID field should differ between different packets sent by a source to a given destination.[2] A common method that hosts use to

[1] 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.

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

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 $\mathcal{P}$.

(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.

(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 $\mathcal{P}$.

Solution:

(a) Send a ping to $\mathcal{P}$ at the beginning of the window and another ping at the end of the window. Check if the difference in identification values is greater than 1. You cannot use this same method to test if $\mathcal{P}$ received a packet, because the counter only advances on sending.

(b) Send an initial SYN to $\mathcal{V}$ with source address set (spoofed) to that of $\mathcal{P}$ and destination port set to n. Using the technique in part (a), test whether $\mathcal{P}$ then sends out a packet.

This works as follows. Suppose that $\mathcal{V}$ does *not* have a service running on port n. In that case, when $\mathcal{V}$ receives the initial SYN (purportedly from $\mathcal{P}$), it will

respond by sending a RST packet to $\mathcal{P}$ (per the second fact above). When $\mathcal{P}$ receives this packet, it ignores it, per the last fact listed above. Thus, in this case, the measurement will show that $\mathcal{P}$ did not send a packet.

Suppose that instead $\mathcal{V}$ *does* have a service running on port n. In this case, when it receives the initial SYN, it will respond by sending a SYN/ACK to $\mathcal{P}$ (per the first fact). $\mathcal{P}$ however will find this SYN/ACK unexpected, and respond with a RST packet (per the third fact). Thus, in this case the measurement will show that $\mathcal{P}$ *did* send a packet, enabling us to tell the two cases apart.

If $\mathcal{P}$ is not quiescent, this method won't work. This method requires that $\mathcal{P}$ can only be sending packets in response to $\mathcal{V}$; if $\mathcal{P}$ is sending packets to other hosts then we can't know for sure that the increment we see is in response to $\mathcal{V}$ or just a packet to some other host.

(c) Two possible approaches:

- Manage a separate counter per-destination. This suffices to block the attack, because the attacker can now only observe a counter associated with sending to their own system, which does not allow them to infer about traffic $\mathcal{P}$ sends to other systems. This is the simplest approach, though it costs a modest amount of state.
- Use a pseudo-random number generator to generate 16-bit values for the counter, rather than incrementing it. This approach renders the relationship between successive ID fields difficult for the attacker to guess.

Problem 3 ***Abusing ARP and Routing Tables*** **(20 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; in part (c), her attack must involve exploiting the router's routing table.

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 WiFi 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 WiFi address `00:1A:AA:BB:CC:DD`."[3] 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.

(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?

> **Solution:** Spoof ARP replies from Brewed Awakening's router, and mount a man-in-the-middle attack. Upon its initial connection to Brewed Awakening's network, Albert's laptop will send out an ARP request for the WiFi address of the gateway router. The attacker can then send Albert a spoofed ARP reply that causes Albert's laptop to associate the attacker's WiFi address with the gateway router's IP address. Now whenever Albert sends a packet to GoodCorp, it will be sent to the attacker. When the attacker intercepts the packet, they can modify the data part of the packet to contain a bogus tourism answer and then forward the packet on to GoodCorp.

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

Or, rather than modifying the data portion of the packet, alternatively the attacker could use the same sort of ARP spoofing to simply observe the packets that Albert sends, learn the session cookie that his browser has for GoodCorp, load the cookie into their own browser, and then access the GoodCorp web site posing as Albert.

(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?

Solution: Spoof ARP replies from both Brewed Awakening's router and Albert's laptop, and mount a bidirectional man-in-the-middle attack. In addition to ARP spoofing as in part (a), the attacker can send an ARP reply message to the gateway router that will cause the router to associate the attacker's WiFi address with Albert's IP address. The gateway router will send an ARP request for Albert's IP address because the gateway router does not yet have the WiFi address mapping for Albert's machine, as Mallory makes sure to time her attack to come right after the router and Albert's ARP cache/table expire. Now when GoodCorp sends packets to Albert, the router will forward them to the attacker instead of to Albert's laptop. The attacker can then modify the data part of the packet to contain Albert's original answers, and forward the modified packets to Albert's laptop.

Note that if in part (a) you used the cookie-stealing attack, then Mallory still needs the additional step framed in the previous paragraph to mislead Albert into not seeing the bogus tourism answers. Mallory cannot mask the answers solely by logging into GoodCorp using Albert's cookie.

(c) Mallory doesn't want to wait until the next time Albert comes to Brewed Awakening in order to enact her malicious plan. While waiting in line for the bathroom, she catches a glimpse of the router's brand. She happens to know that this particular brand has a vulnerability that allows her to modify the routing table. (The routing table is a data structure that essentially indicates, for each possible IP address, where to forward packets destined for that IP address: e.g., they can be sent out over the router's Internet link, or they can be sent out over the local WiFi network. It is also possible to specify that packets should be forwarded to a specific IP address for their next hop.)

Mallory runs the `dig` command and finds out that GoodCorp has many different IP addresses it uses to load-balance requests across multiple servers. She knows

that Albert's browser will be using the first IP address that dig returned. How can Mallory impersonate Albert on GoodCorp now? (For this part, you do not need to worry about tricking Albert into thinking everything is fine.)

Dig output:

```
www.goodCorp.com: 23.22.249.134
www.goodCorp.com: 23.22.249.135
www.goodCorp.com: 23.22.249.136
www.goodCorp.com: 23.22.249.137
```

Solution: Mallory can modify the routing table to mount a man-in-the-middle attack. First, she can modify the routing table so that any packets sent to 23.22.249.134 (the GoodCorp IP address Albert will be using) are forwarded to her. Then, when she receives the packets, she can change them to have bogus tourism answers. Next, she can send the modified packets on to GoodCorp using any of the other 3 IP addresses for GoodCorp. (She can't use 23.22.249.134 because that address would come right back to her!)

(d) 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.)

Solution: Using Wireshark, Mallory can observe the packets that Albert sends, learn the session cookie his browser has for GoodCorp, load it into her own browser, and then access the GoodCorp web site; it will think she is Albert.

Comment: This attack works on any open WiFi network, as open WiFi networks don't use encryption. It also works on WiFi networks that use today's common form of encryption, WPA-Personal, if the attacker can connect to the network: when you connect to the network, you are given the crypto keys needed to encrypt and decrypt all of the WiFi traffic. WPA-Personal only attempts to prevent eavesdropping by outsiders who cannot connect to the network; not eavesdropping by others on the same network. Only WPA-Enterprise would protect against the attack, and it is not nearly as widely used because it requires

pre-configuration. So, this attack remains possible today against common WiFi networks, if the victim is using HTTP. Stick to HTTPS if you are doing anything sensitive over a public WiFi network!