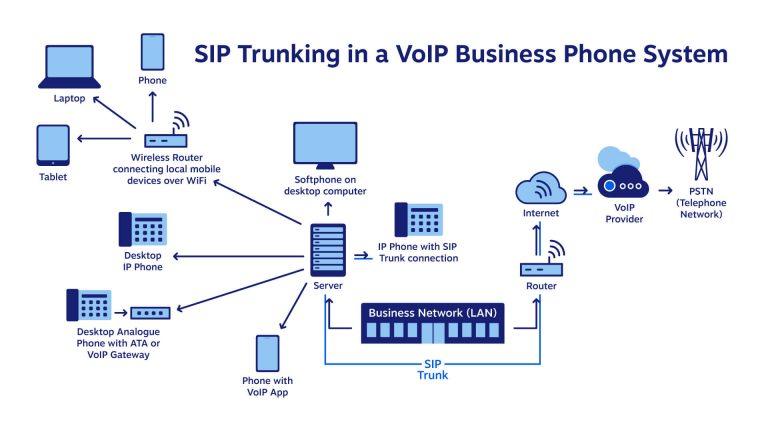
Notes:

* As I said, we don’t expect dedicated hardware PBX setup, while hacking a software PBX (“IP PBX”) is not much different from hacking a regular computer system. I guess it would be more interesting to demonstrate how an IP PBX system works to students including how it connects to the PSTN (I tried it just now and it was also new to me), and for EECS students, let them practice hacking a known-vulnerable system using one of the techniques of their choice learnt previously / elsewhere.

Preparation:

* Freshly installed FreePBX systems inside VirtualBox. If possible, provide .ova files with installation done; if not, provide installer .iso files, and instruct students to create a virtual computer and install it as the first step when the lecture is given. (act as “server” in the diagram) Likely “Bridged Adapter” is needed in VirtualBox network settings.
* On the host machine, install a SIP client (act as “softphone on desktop computer” in the diagram), connect headsets
* (Optional) Instruct students to install a SIP client on their mobile devices (act as “phone with VoIP app” in the diagram, devA)
* Make students have telephony service on their mobile devices (multiple roles, preferably on a second/third device - maybe two students in a group? devB, devC, etc.)

Lecture:

* Explain the PBX system, IP PBX, SIP protocol, SIP trunking, IVR, PSTN stuff.
* We’re using this setup. Not all components exist. We use the builtin SIPstation trunking (“VoIP provider” in the diagram). Explain that all call processings happen on our FreePBX instance and SIPstation only provides connectivity to the PSTN. They would also bill us by the amount / minutes of calls (after the trial period).  
  

Steps:

1. Activate the FreePBX instance and set up an account with SIPstation / Asterisk. Make up some information (they ask something like “what does your business do and what’s your business address”). As of writing, they provide a 20-day free trial which should be enough for the class. After initial set up, they demand me to do a “verification” by calling them and talking with a person (not necessarily from the FreePBX setup). I suspect that rather than verification, it would be for a sales pitch and I didn’t bother to call.
2. By setting up the account, you would get a phone number on your PBX (you could choose it during the setup). Technically the phone number is on SIPstation and SIPstation connects to your PBX when the phone number is called.
3. Add two telephone extensions (assume they’re “Sales” and “Support”), connect to them using SIP clients on the host computer and devA. (“Applications -> Extensions”)
4. Set up an IVR so that when the phone number on your PBX is called, the user hears a message “Welcome to XX company. To place an order, press 1; to inquire about a previous purchase, press 2” and the corresponding extension is connected according to user selection. (“Applications -> IVR” and “Connectivity -> Inbound Routes”. You’ll also need things like “Applications -> Announcements” to make the recordings)
5. Use the devB to call your PBX and test / verify your setup.
6. Call the devB by its phone number using one of the SIP clients through the PBX. This is verifying the setup in the other direction. Now a proof-of-concept PBX for a small company is done.
7. Let’s see some way sneaky to make outbound calls. Edit the IVR, add a third entry with digit 3 pointing to a new “Misc Destination” (“Application -> Misc Destinations”) of the number on devC. Note that there’s no need to include the entry in the audio announcement message. We take this approach since students already worked with IVR configurations.
8. Use the devB to call the PBX. Press 3 (despite not being announced in the message) and you get connected to devC. devC would receive the call from the PBX (not devB). Imagine the scenario when PBX is running on a toll-free number and devC is a premium-rate phone number: PBX owner pays all costs involved in the call.
9. Business students stop here.

Hacking the PBX:

I haven’t tried them myself yet. Students can make choices based on their skill level.

* Weak password: Imagine the PBX configuration interface is open and only protected by a password. If the administrator selected a weak password and it got known by the hacker, the hacker can log in to the system and mess up the configuration. Construct another approach to drain PBX owner’s money.
* Firewall bypass: FreePBX has a builtin firewall system. As in the design, the configuration interface can only be accessed from specific addresses and internal networks. Assume the PBX instance is well-protected but another host inside the internal network is not, and it got broken into; access the PBX through the faulty machine and do something similar to the above.
* Vulnerabilities in FreePBX: there can be security bugs in the FreePBX software. CVE lists some of them. Refer to the links in the references to identify a faulty FreePBX version, install it, and try to attack it according to published vulnerability details.
  + One example linked below is an XSS vulnerability: “The vulnerability results in arbitrary javascript execution if a user clicks an external malicious link while being logged in as an administrator The user has to be previously authenticated as a FreePBX administrator and be tricked into clicking an external link that would generate the javascript XSS”. Try to construct such a malicious link so the “another approach to drain PBX owner’s money” happens when a PBX administrator clicks on such a link.

Links:

* FreePBX:  
  <https://www.freepbx.org/downloads/>  
  <https://downloads.freepbxdistro.org/ISO/SNG7-FPBX-64bit-1805-2.iso>
* Image source: <https://www.nextiva.com/blog/hosted-pbx-vs-sip-trunking.html>
* FreePBX CVE:  
  <https://www.cvedetails.com/product/10928/Freepbx-Freepbx.html?vendor_id=6470>  
  <https://wiki.freepbx.org/display/FOP/2018-09-11+Core+Stored+XSS>   
  <https://issues.freepbx.org/browse/FREEPBX-18161>