CS530 Professor Wang
Group Project
Team Members: Jang, Wonhyuk: Kanai, Kenichiro: Xing, Zemin: Jamil, Ammar
Title: Stop Terrorism and Hack a Bank for about $5: Proof of Concept: Dedicated Server Hardening Devices

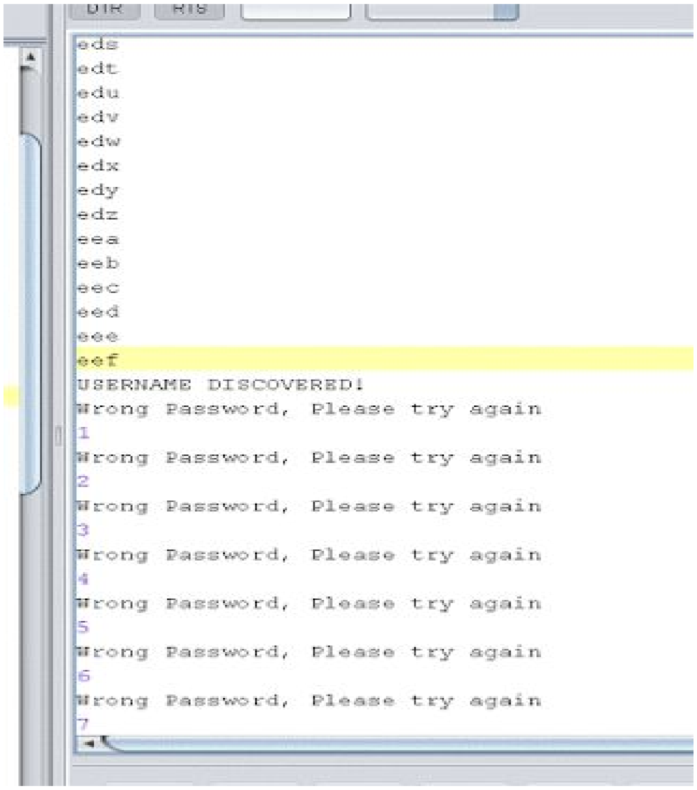
Our motivation was to learn how hacking works by investigating various aspects of the deed. 1st we learned how servers work along with basic networking, and then we wrote an application using a cheap microcontroller to brute force discover a valid username and password. We built the server in C++ using Visual Studio and wrote microcontroller code in both C (project version 1) and LUA (project version 2) languages. We achieved all this with the limited resources of a microcontroller and limited libraries that did not even provide layer 3 networking protocols (such as ping).

If this were a real hacking attempt is would go as such; we find a target server (like the one we wrote) and we send it a message then listen for a response. Most servers would introduce themselves by sending back some sort of hello message to identify themselves. From here it is a matter of looking for documentation about the server and then figuring out how to get in. In our case, our server simply replies by asking for a username followed by a password once a correct username is detected. We customized our microchip to respond specifically to our server by looking for specific replies. Then it was just a matter of implementing a brute force algorithm to gain entry.

Although you can do a lot of this using a computer, a computer is not very user friendly (especially for non-programmers such as most administrators) and it is much more bulky than a small chip the size of a quarter. By building this functionality on such a small and cheap piece of hardware, we can now create dedicated devices that are much easier for systems and network administrators to use for their own security scans and server hardening exercises. Further, this knowledge is directly transferable to cell phone technology since most use similar type RISC chips and 802.11 radios all basically function the same way.

Other advantages to using the ESP8266 chip is that it is very cheap, easily expandable, and uses very little power averaging about 90mA/hr while in 802.11 mode. This chip is also production ready, in that, it can be used for mass production of devices unlike more popular hobbyist hardware like the Raspberry Pi or Arduino that are too bulky and/or expensive for production purposes. Even the Raspberry Pi Zero would not be able to compete on the price of the ESP8266 and it does not come with Wifi capability.

As for expanding this project, it would be almost trivial to add more software functionality to this device such as network scanning and penetration, server discovery (will need some layer 3 libraries built into the firmware), port scanning, server fuzzing, and even protocol discovery (finding all the commands available on a server you already logged into even if it is an undocumented server).



```
eds
edt
edu
edv
edw
edx
edy
edz
eea
eeb
eec
eed
eee
eef
USERNAME DISCOVERED!
Wrong Password, Please try again
1
Wrong Password, Please try again
2
Wrong Password, Please try again
3
Wrong Password, Please try again
4
Wrong Password, Please try again
5
Wrong Password, Please try again
6
Wrong Password, Please try again
7
```

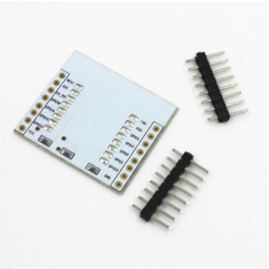
In this picture, we sent all combinations of username starting from 1 character to 3 character. Right after we send correct username, server send back to the chip and we start sending all possibilities of password character range from 1 to 3.

Video Link:https://www.youtube.com/watch?v=kXTNbqOPfc8

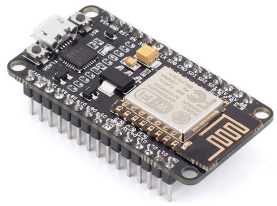
Our Hardware

ESP8266 -07 or ESP-07
(both version 1 and 2)

ESP8266-07 Breakout PCB
(for breadboard pinouts)



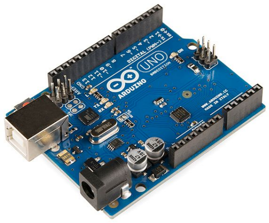
Nodemcu
ESP8266 with bells and whistles



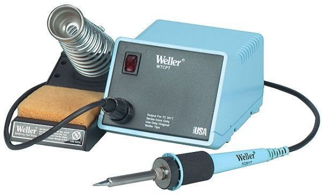
Generic FT232 Adapter
(Serial via USB Adapter)

Generic Breadboard PS Unit
(3.3V/5V up to 5A)

Arduino Uno
(not used in version 2 of project)



Very Crappy Soldering Iron



...Some Wires