CSG 1105 / 5130 - Applied Communications

Module 11 Tutorial

Objectives

- To learn about 'sniffing'
- To learn about password scoring
- To learn good personal security measures

By the end of this workshop you should be able to

- Use tools to sniff passwords and cookies
- Use tools to determine if your password is adequate
- Be able to secure your own devices or usage to prevent security flaws / hacks affecting you

Required Downloads

- Demonstration Video - Sniffing & Man in the Middle Attacks
- Wireshark (Available in Unit Documents on Blackboard)
- L0phtCrack - Link available in Unit Schedule - Module 11 on Blackboard

Required Software

- A terminal or command prompt capable of Telnet - I would advise either of the following (links available in Unit Schedule - Module 11 on Blackboard):
  - Mac OS X - iTerm 2
  - Windows - PuTTY

Optional Downloads

- A recording of this Tutorial from Blackboard; discussions are held in class and clarifications are made too.

Demonstration Video - Sniffing & Man in the Middle Attacks

Watch the video listed in the required downloads showing how easy it is to sniff session cookies using a simple Man in the Middle Attack when a website doesn't use HTTPS (SSL) on all of their pages.

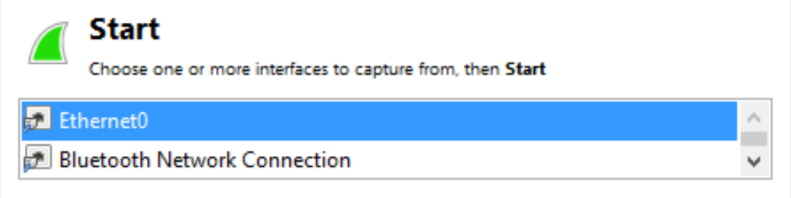
Note: The video will not show you how to set this kind of attack up due to ethical reasons.

Activity Module 1 - Sniffing Credentials

Part 1 - Telnet Credentials

Open both **Wireshark** and also your preferred **terminal app**. We will be going through how to take the credentials of a simple telnet session by sniffing the packets in Wireshark. Keep in mind - this is currently only going to work on your local workstation or personal laptop due to the fact that the machines are not connected to a Hub (which broadcasts all traffic), and that the network interface is not running in *Promiscuous Mode* or *Monitor Mode*. In order for Wireshark to be able to sniff **all** traffic on a network your network interface, drivers and operating system have to fit certain configurations otherwise it won't work correctly.

Once you have Wireshark open start a capture of the packets by selecting all the interfaces and clicking the **Start Capture** button. See below for an example screenshot.



Now add a **filter** to limit our scope to **telnet** (press enter to apply it) and we'll then switch to our terminal app that we have decided to use. Once you are in your terminal enter the following command:

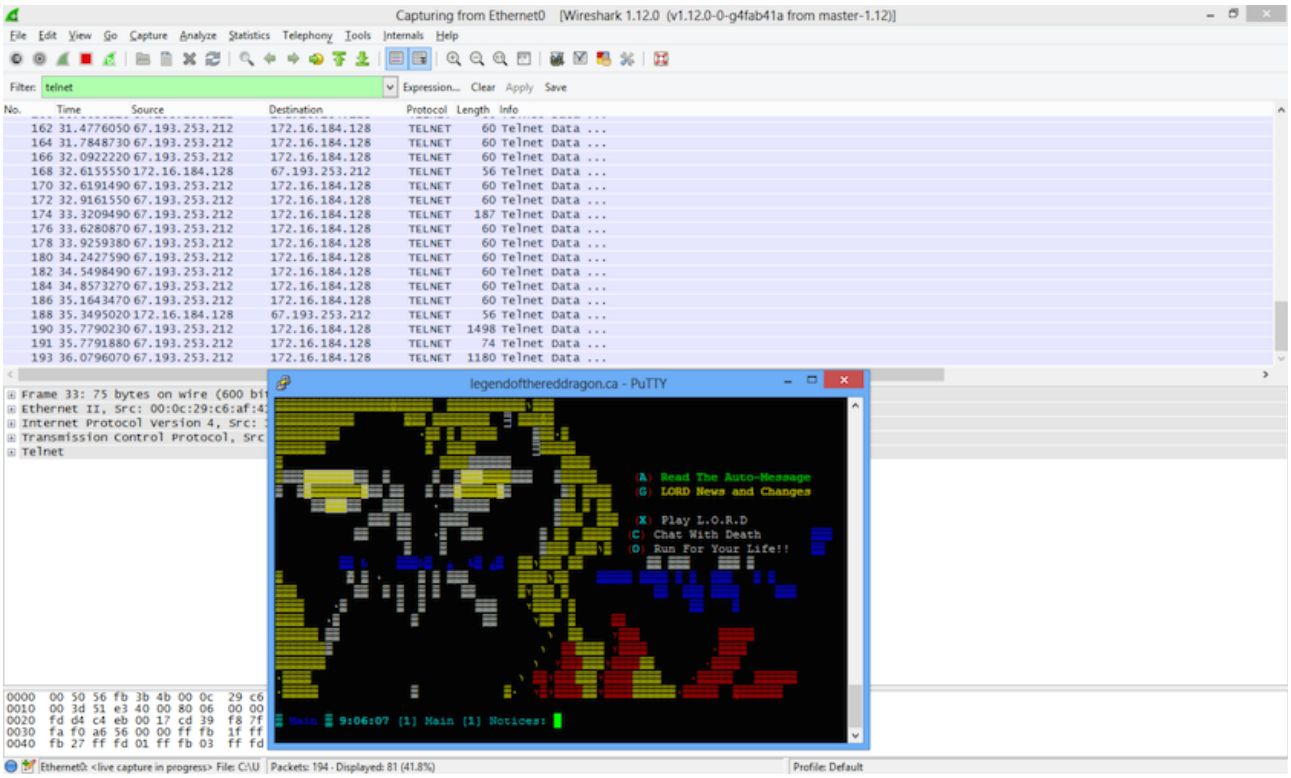
```
telnet legendofthereddragon.ca
```

This is a text based online game which can be played through any terminal app that has telnet capabilities. For ease of use and quick access I have created a test account for you to sniff the credentials of. When it prompts you, log in with the following credentials, but be sure to pay attention to your Wireshark window as well:

Username: `ecutest`
Password: `ecu1test`

It may automatically capitalise the first character of the username - do not worry about this.

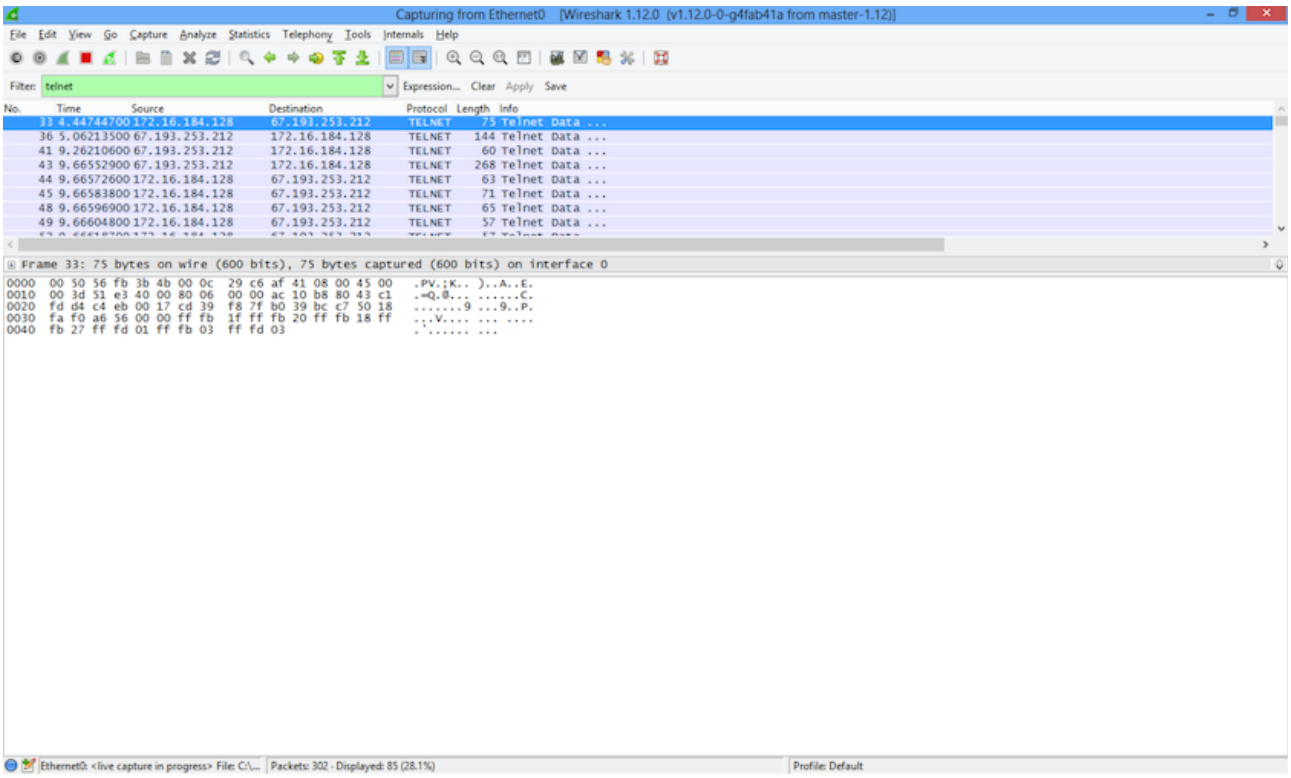
After you have connected to the server your screen for both Wireshark and your terminal app should look similar to mine below.



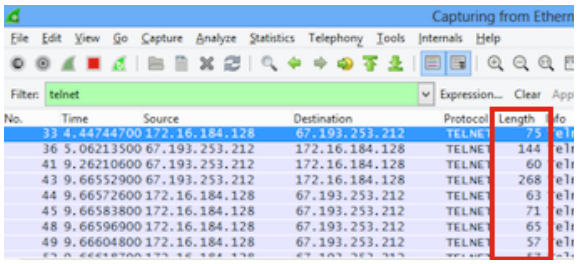
If you paid attention to the Wireshark window you will have noticed that every key you entered there was a new communication of telnet data captured in Wireshark.

Why? This is because as you type in on your terminal app it sends that key-press to the telnet server which then sends back a newly updated telnet display to your client. This makes looking through individual packets for user credentials quite a pain as you'd be looking character by character. Let's have a look at that in action.

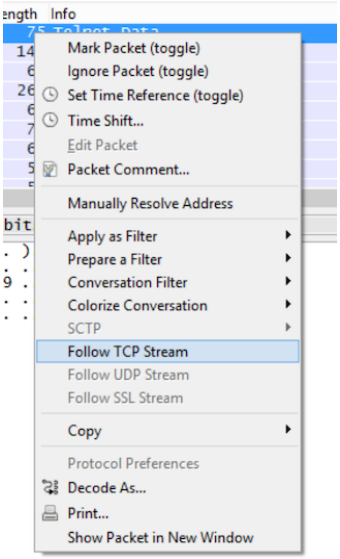
Go back to **Wireshark** and go to the top of the captured packet list. Let's set up our Wireshark window to be able to look at this captured information more easily - go ahead and resize the different windows so the frame information pane (bottom one) is the largest and so the layer pane (middle one) is very small, like mine on the next page.



Now, progress through the list with the down arrow until you get to some of the larger packets. We can tell that they are larger by looking at the **Length** column in the capture pane (top one), I've circled them in the picture to the right.



Once you get to the larger packets keep progressing keeping an eye on the data in the frame information pane. Eventually, you should see a very large frame that then prompts for '**Login:**'. When you have seen this frame keep progressing through the packets, you'll notice that the packets now are very small and if you look at the last character of these small packets you'll begin to see the individual letters of our username, but they begin to get jumbled up with the incoming packets from the server to update our view on the client side. This can also be done to look at the password.

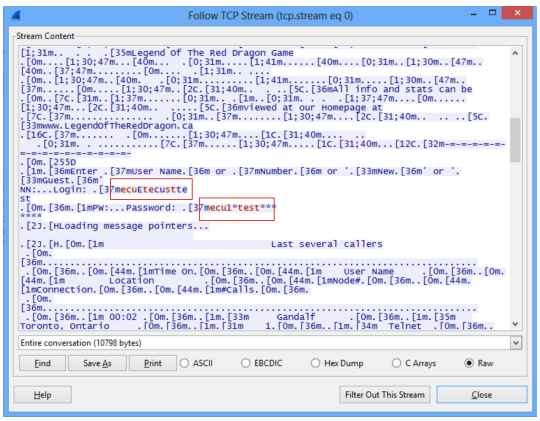


Frustrating?

Now let's look at the easier way to do this. Go back to the first packet then **Right Click** on it and select the option **Follow TCP Stream**. A new window will open showing you all of the telnet data that has been sent back and forth between the client and the server.

Wireshark has an easy way to identify where the data has originated from (client or server). The red information is user input information and the blue is the information sent from the server.

If you scroll down and look for some red text you should eventually see the username and password amongst the wall of text. I have identified it in my TCP stream on the next page.



Ignoring the returned information from the server we can make out the username of ecutest and the password of ecu1test.

Congratulations! You have sniffed someone's credentials from a telnet session.

Note: Telnet for private information and what not is incredibly uncommon in todays world as SSH has by-and-large replaced it. When SSH packets are sent they are always encrypted - as such, all information contained within them is sent as ciphertext, making this method of sniffing credentials useless.

Part 2 - HTTP Credentials

Now, let's look at sniffing a username and password from a website. This is basically the same process that was involved in the video from the demonstration except we do not need to set up a Man in the Middle attack. Once again, this will only work from the local machine you are on unless you are able to put your network interface into *promiscuous mode*.

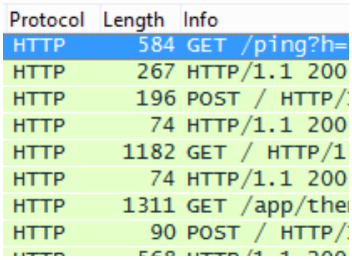
Keep in mind, that this process also relies on unencrypted communications - this means that for HTTPS enabled websites this will not work (the cookies method may if the website has some unsecured pages).

Once again, we will be using a test account to sniff. Open your web browser and navigate to the website (type it in full) *http://www.lifehacker.com.au* and then click on **login** towards the top right. Now, start a new capture in **Wireshark** (we have done it backwards for ease of use).

Once you have started capturing the packets login with the following credentials:

Username: `ecutest`
Password: `ecu1test`

Once you have successfully logged in, go back to Wireshark and stop capturing. Apply a filter for **http** and press enter to apply it.



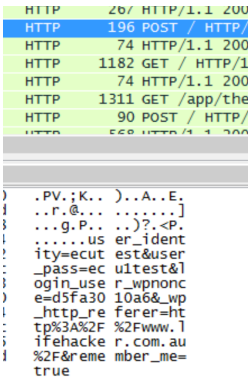
We won't bother trying to look through every individual packet we captured, instead we'll keep an eye out for key terms in the **Info** section (the far right column). You should be able to see some different terms there such as **GET**, **HTTP/1.1** and **POST**. These are different methods of retrieving or sending information between webpages.

If a website is not using any form of encryption it will normally send the credentials through **POST** (or in rare, incredibly unsecured websites **GET**). These two methods basically do the same thing although **POST** sends through the information as a variable, while **GET** places the information in the URL of the new page.

There are some key terms when looking through the packets to identify the login credentials of the user account, you can probably guess them but they are normally *username, identity, email, user, pass, password, secret, key,* etc. All of these terms will be common names for the variables that the user information will be stored in.

At this point start looking at the information sent to and from the sever. Focusing on the GET and POST packets look for the term *pass*. There are two reasons for this, firstly I have already identified that as the term used in the LifeHacker website, and, secondly the password field has

fewer different terms for the variable name than the username field. You should look through the plaintext sent located in the bottom pane, see right for a screenshot.



You can see in my screenshot that I have found the POST packet containing the users credentials. We can see the information as `user_identity=ecutest&user_pass=ecu1test`.

That's how you sniff website credentials for unsecured pages!

Activity Module 2 - Port Scanning

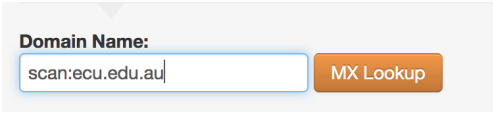
A key tool when looking for holes in someones security - or for bolstering your own security is to make use of port scanning. Ports are the holes in the firewall that allow traffic in and out of your network. You may know some common ports already, such as:

80 - HTTP
21 - FTP
22 - SSH
23 - Telnet
25 - SMTP (Outgoing Email)
110 - POP (Incoming Email)

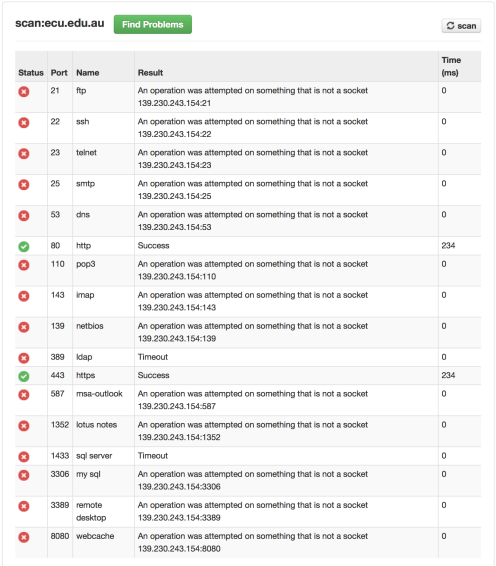
There are thousands upon thousands of ports for every protocol, every application and some are shared as well. These are the holes that hackers want to exploit and security personnel want to close up or protect.

There are many different port scanning tools out there but there are also some online websites available to perform these sorts of checks for you as well. A quick and easy to use website which scans the most common ports is *http://mxtoolbox.com*. Let's browse to it now and check what ports ECU appears to have open.

Once you have gone to the website, type in the following into the **Domain Name** field and press the orange **MX Lookup** button, like below.



After you have done so it will show you the results for it's quick scan and show you what ports are open. ECU appears to have their ports locked down quite tough!



MXToolbox is great for a quick scan to see if the most common ports are open, however, to check if there are other potential openings in the firewall a proper portscan tool is recommended for use.

Nmap is a great tool available from *http://nmap.org*, it has the capability to search for every port possible, however, it will take a very long time to do due the number of ports. There are 65535 ports in 3 categories:

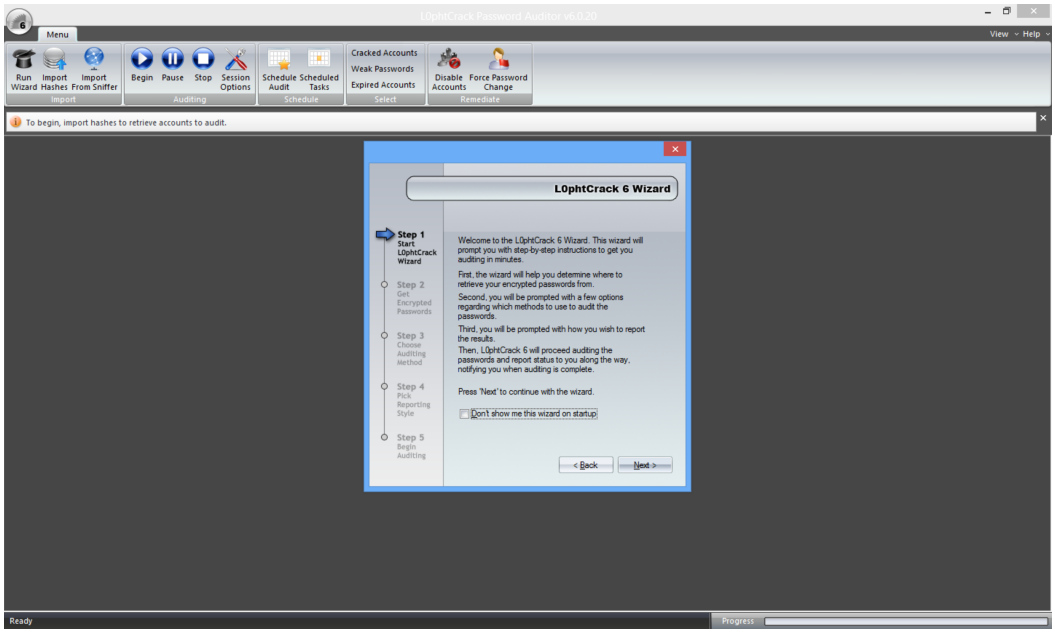
0 - 1023: Well-known
1024 - 49151: Registered
49152 - 65535: Private

Activity Module 3- L0phtCrack

L0phtCrack is a password auditing tool designed for system and network administrators to rate their users passwords and help manage a high level of security in the passwords chosen by the users.

Note: Students on-campus will need to download this on their personal machine or watch the tutorial.

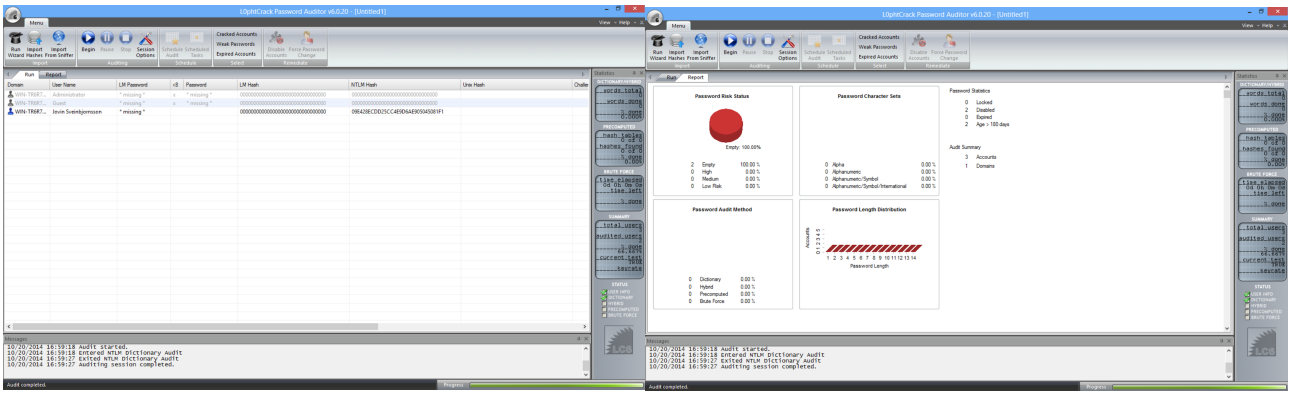
Once you have installed the trial for L0phtCrack open it and you will be welcomed by it's initial window and a Wizard. For ease of use, we will progress through the wizard (pictured below). Follow the instructions below to continue with the Wizard:



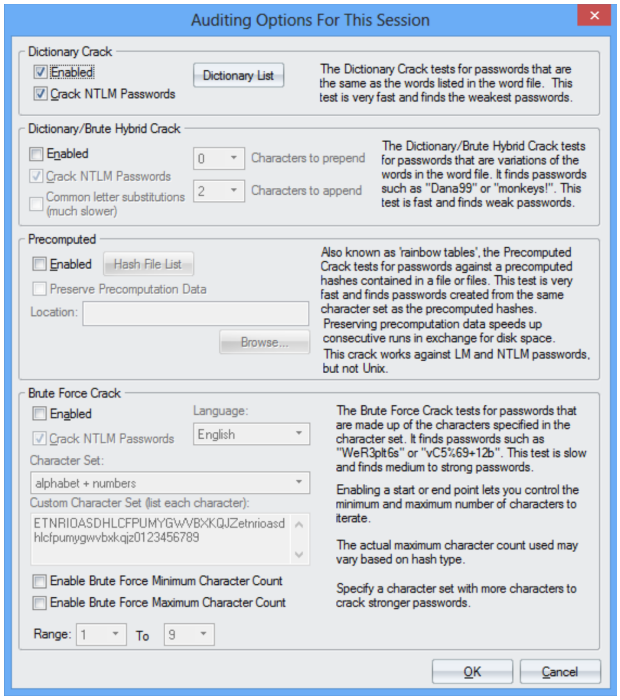
Step 1: Click Next
Step 2: Select *Retrieve from the local machine*, then click Next
Step 3: Select *Quick Password Audit*, then click Next
Step 4: Ensure all check boxes are ticked, then click Next
Step 5: Uncheck *Save these settings as session defaults*, then click Finish

The first time you run the service it will notify you that it is sending a tool across to the remote server, click OK, this is how it is able to retrieve the password hashes for the user accounts.

If you look to the right hand side you can see the currently testing word against the hash, after a short while this will complete and you can check the results and report on the system security.



Let's have a look at a more in-depth scan of our users now. Click on the **Session Options** button at the top in the Auditing section, you should then be able to see some new options for our password audit session.



Go through and check the box for the **Dictionary/Brute Hybrid Crack** and then click OK. As it says on the side, a Dictionary/Brute Hybrid is a password cracking algorithm which chooses words from a dictionary as well as modifications to it (added numbers, symbols etc.)

If you enable the **Brute Force Crack** it will try every possible combination of every character available in the character set. The more characters available in the set and the larger the range you use will increase the time to crack the password by a *very* long time.

The **Precomputed** crack is one where you can provide a Hash File which is a file providing some predetermined tests and combinations of characters. These files can be very large (several hundred gigabytes) however they have the benefit of greatly improving the speed in which a password can be cracked.

Now that we have selected some new options for our audit, lets **Begin** again (the big blue play button). On the right hand side you can see the progress of the check. Here it will provide you with the information on how many different passwords it has tried and how far through the check it is. You can see even without a traditional **Brute Force** attack it will still take a very long time.

As a network or system administrator you can use this tool to regularly check the quality of the passwords of your users and **force** password changes for those that are too weak.

Securing Yourself
Now that you know about these security flaws and improvements, what will you do to help improve your personal security?

Use HTTPS whenever possible? Instead of telnet try switching to SSH? Maybe investigate the use of Proxies or VPNs to aid in securing your information. Encrypt drives, ensure your passwords are strong (long doesn't mean strong).

The best method is to stay alert with the networks you are on and make sure you are not logging onto any and every service on a public network.