SEC301 | INTRODUCTION TO CYBER SECURITY
GIAC Information Security Fundamentals (GISF)

Lab On Demand Workbook

SANS | GIAC CERTIFICATIONS

THE MOST TRUSTED SOURCE FOR INFORMATION SECURITY TRAINING, CERTIFICATION, AND RESEARCH | sans.org

SEC301-1.1: Introduction to LODS & Quizzes

Objective

This lab has three objectives:

1. Our first objective is to ensure everyone can access LODS

2. Our second objective is to get students used to working in the LODS interface.

3. The final objective is to make students aware of the quizzes that are available to them as part of the SEC301 experience. You will conduct the quiz for Module 1 here.

Scenario

Virtual Machines

1. SEC301-Win10
2. SEC301-LabServer

Introduction to LODS

In this portion of the lab, we make sure that you know how to utilize the LODS platform.

- *LODS stands for Lab On Demand System*
- *It provides each student with their own, independent virtual lab network environment*
    - *Each lab environment is running one or more virtual machines*
- *Activities of one student do not impact other students in the class*

1. **Look at the LODS Screen**

   - When you look at your screen, you see a blue border along the right-hand side of the screen. The primary steps for your lab are listed here.
   - At the bottom, you see the box (the one you are reading this text in) that contains the detailed steps of the lab.
   - These are the two primary areas for finding lab instructions, but they are not all the areas we highlight in future stages of this lab.
   - When you have identified these areas, click the "Done" button to the right of this instruction.

2. **Understand the Done button**

At the end of each task, it is essential that you click on the "Done" button on the bottom right of this box. Each time you click on "Done", you advance to the next task, and these instructions change to match that task.
Sometimes, you may see the "Done" button disappear. This happens when you click on another task without clicking on Done. To get the "Done" button back again, notice that the current task has a small arrow next to it. Click on that task, and the "Done" button reappears.
You can also accomplish this by looking at the bottom right of the screen. There is a black area that, when you do not have the current task selected, says "Return to current task". Simply click that and you will automatically return to the task you are currently working on.

-- when you are finished reading this task - click on "Done" to the right of these instructions --

3. **Understand Tips**

   On specific tasks, you might notice the head toward the bottom left of the screen highlight. When this is the case, there is a "Tip" about that task. Just click on the head to view the tip. When done reading the tip, click on the head again to close the tip.

   If you have not already done so, click on the head now.

   Remember to click "Done" when you are finished looking at the Tip.

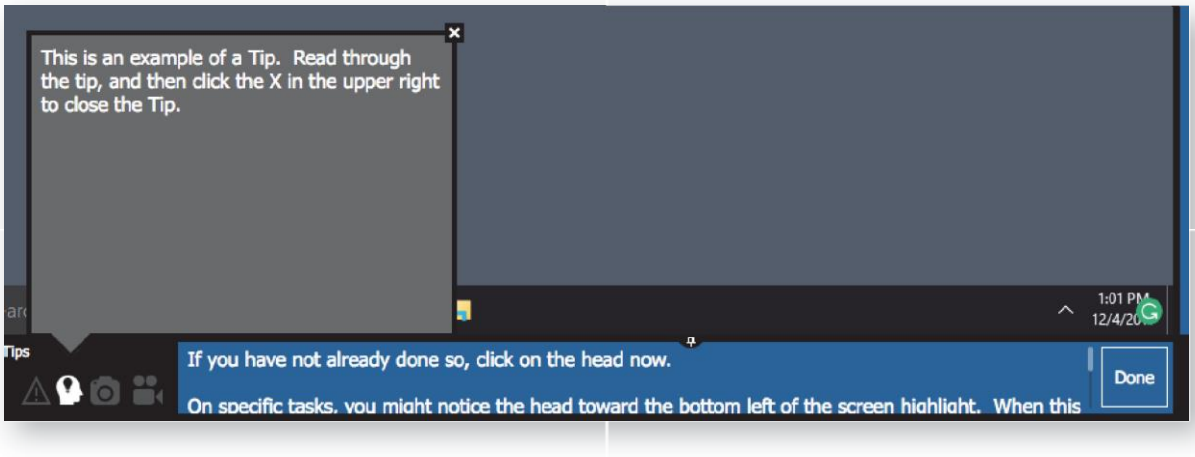
   

   Inside this box is an example of a Tip. Read through the tip, and then click the X in the upper right to close the Tip. If you prefer, you can just click on the head icon again to close the tip.

4. **Understand Screenshots**

   Sometimes, we make screenshots available to illustrate what you should see on your screen at that time. You know there is a screenshot when the camera icon highlights. Click the camera to view the screenshot and then click the camera again to close it the screenshot.

   Don't forget to click "Done" when you finish this step.



5. **Understand Warnings**

   On a small number of tasks, as soon as you move to that task, a red warning will pop open on your screen. Warnings indicate something we believe to be especially crucial for you to read. If you need to close the warning, just click on the warning sign icon to close the warning. Note, you can click the warning sign icon again if you would like to reopen it. This also works for tips and screenshots.

   Click "Done" when you are finished.

   ⚠ Just as with the Tip, read the text of the Warning and then the X in the upper-right corner to close the Warning or click the warning sign below to close the warning.

6. **Understand Exercises**

   Each lab is made up of some number of exercises, and each exercise contains some number of tasks. Some labs may only have one exercise; others may have two or more.

   This particular lab contains two exercises. When you click Done, you have completed the first exercise (consisting of 5 tasks) and automatically move on to the second exercise. The LODS interface expands the tasks of the second exercise as soon as you click on Done.

   So click on "Done" now.

   ⚠ Future labs will not always tell you when to click on the Done button. You will have to remember to do so.

At this point, you should know how to:

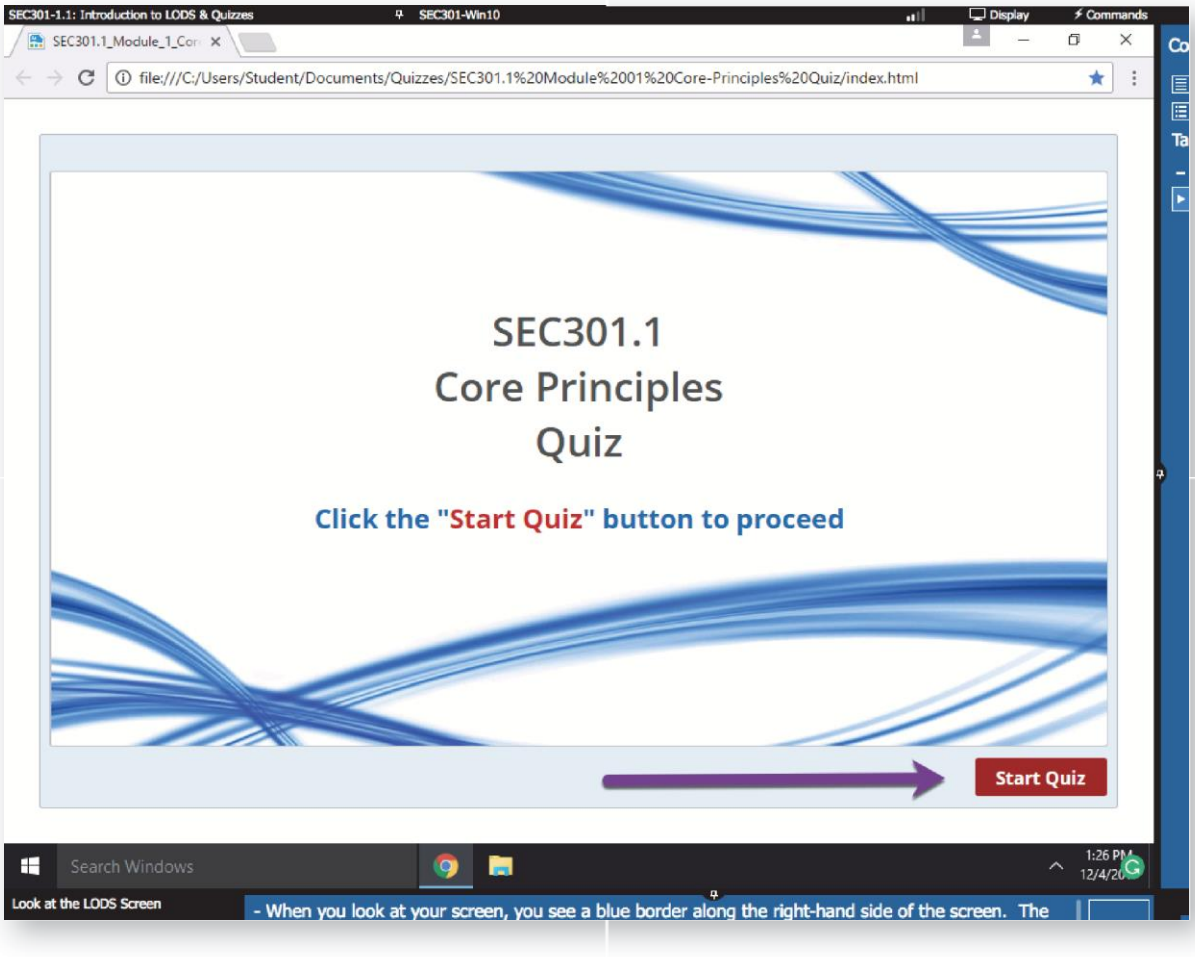
- *Connect to the LODS platform*
- *Maneuver around the environment and find the tips, screenshots, and warnings*
- *Understand how the different quiz question types work*

1. **Open the Quiz**

   In the upper left corner of the LODS Windows 10 screen, you see a question mark icon labeled "Module 1 Quiz". Double-Click that shortcut.

2. **Start Quiz**

   A screen like that shown in the screenshot opens. Click the red Start Quiz button in the lower-right of that screen.



3. Answer Questions

For each quiz question, you answer the question and then click on Submit in the lower-right of the screen. There are several different types of questions used. The quiz engine randomizes the questions each time you take the test, so it is impossible to show you screenshots in the same order you will see the items. NOTE: If you get a question incorrect, the quiz engine will pop up the correct answer.

4. Finish Quiz

When you finish the quiz, review your score. Note the score is for your information only.

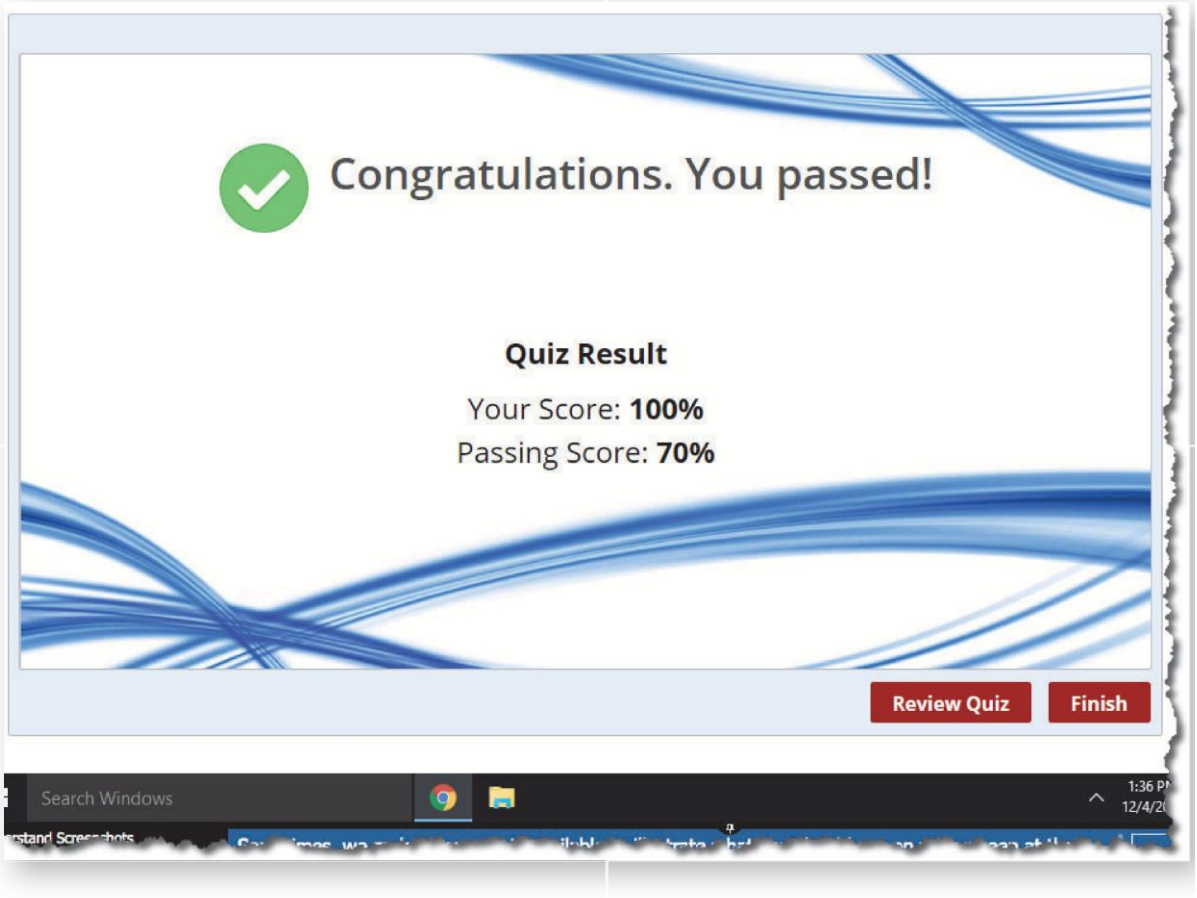
We do not track score information in any way.

When you finish the lab, click "Done" and then click to close the window.

⚠ **REMEMBER:**

You must exit each lab correctly. That means that you click Done at the end of the last step.

You can also click Exit in the far upper-right corner and cancel (or save) any lab. But LODS only allows you to use the resources for one lab at a time.



Congratulations. You have completed Lab 1.1 of SEC301. You now have a good understanding of how to use LODS for the remainder of the labs.

***BE SURE** to click Done one last time to properly close this window. You will need to ensure you end each lab this way.*

SEC301-1.2: Exercise - Building Better Passwords: The Haystack

Objective

In this exercise, you will work with a tool to better understand which passwords are better to use, and which are just not good enough.

Scenario

Virtual Machines

1. SEC301-Win10
2. SEC301-LabServer

Building Better Passwords: The Haystack

In this lab, we examine the password more closely. To do so, we utilize an online tool made available by Steve Gibson of GRC Labs. (Mr. Gibson has been kind enough to allow us to replicate the functionality of his website.) The tool will enable us to type in various passwords. It then reports how long it would take to brute force that password given three possible cracking speeds.

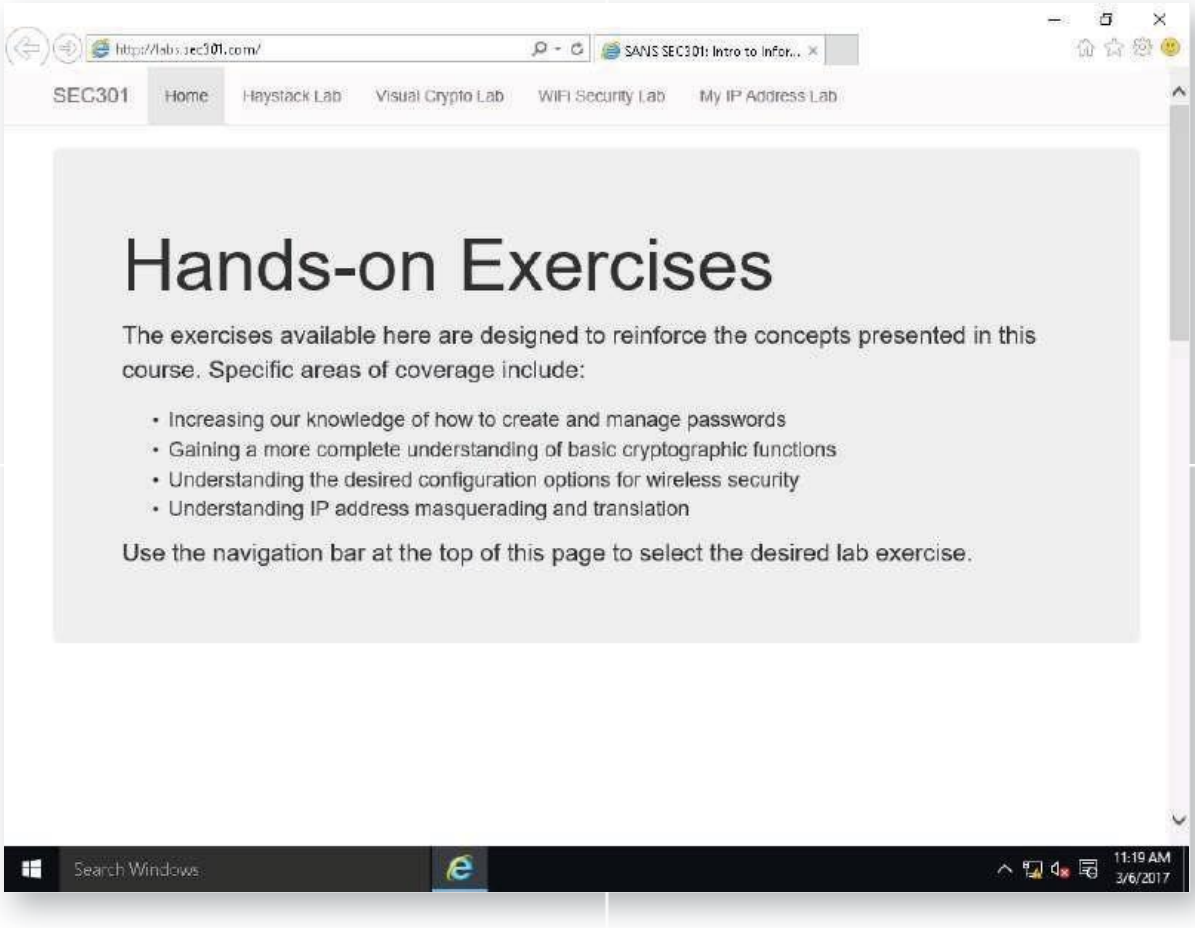
In this way, we can determine which passwords are better to use and which ones just are not good enough. The tool has clear value. Once you are comfortable with it, we encourage you to share this tool with colleagues and family members. This lab should work with any web browser.

1. **Open Chrome**

   Launch Chrome by double-clicking on the desktop icon.

2. **Open the Haystack Lab Page**

   From the SEC301 lab server website, click on the tab near the top of the browser labeled **Haystack Lab*.***



3. **Start the Haystack Application**

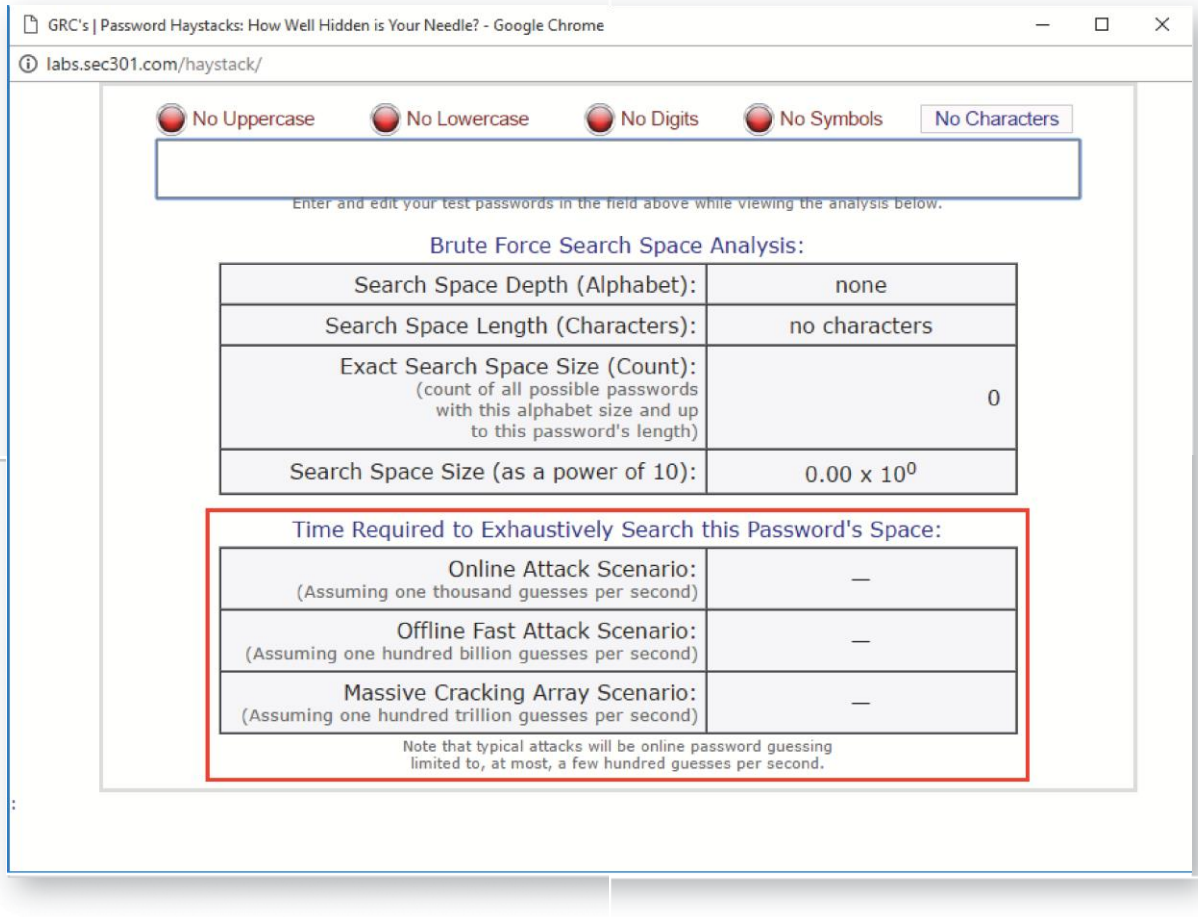
   Read the lab explanation on the Haystack Lab page. When you finish, click the **Open Application** button to start the Haystack application.

4. **Examine Attack Scenarios**

As part of our lecture in class, we discussed the brute force attack against passwords. Remember that this form of attack tries every possible combination of password until one of them works. The Haystack tool measures how long that type of attack would take given three different scenarios:

1. The first scenario is the *Online Attack Scenario*, which assumes one thousand guesses per second. That scenario uses a tool that attempts to log in to an online account as quickly as it is able, each time using a different password from a list. This would be a very fast example of the attack used in 2014 when over 100 celebrities had their nude photos from iCloud publicly posted.
2. The second scenario is the *Offline Fast Attack Scenario*, which assumes one hundred billion guesses per second. This attack utilizes a computer explicitly built for cracking passwords. It would typically have several GPU processors working in parallel to make many, many guesses simultaneously and at an extremely high rate. (A GPU is a Graphics Processor Unit, the processor in a video card. They optimize this processor for the kind of math necessary for high-speed video display, which just happens to be the same math required to make very fast password guesses. Multiple GPUs in a system can make a lot of password guesses very quickly.)
3. The third scenario is the *Massive Cracking Array Scenario*, which proposes one hundred trillion guesses per second. This would require approximately a 5-million machine botnet (in other words, 5 million hacker-controlled computers) doing distributed processing. Each of the 5 million computers would be making as many password guesses as it is capable of, all simultaneously. (Yes, 5 million machine botnets do exist, and serve this purpose.)

Identify where the exhaustive search time is presented for each of these attacks, as shown in the screen capture for this lab.

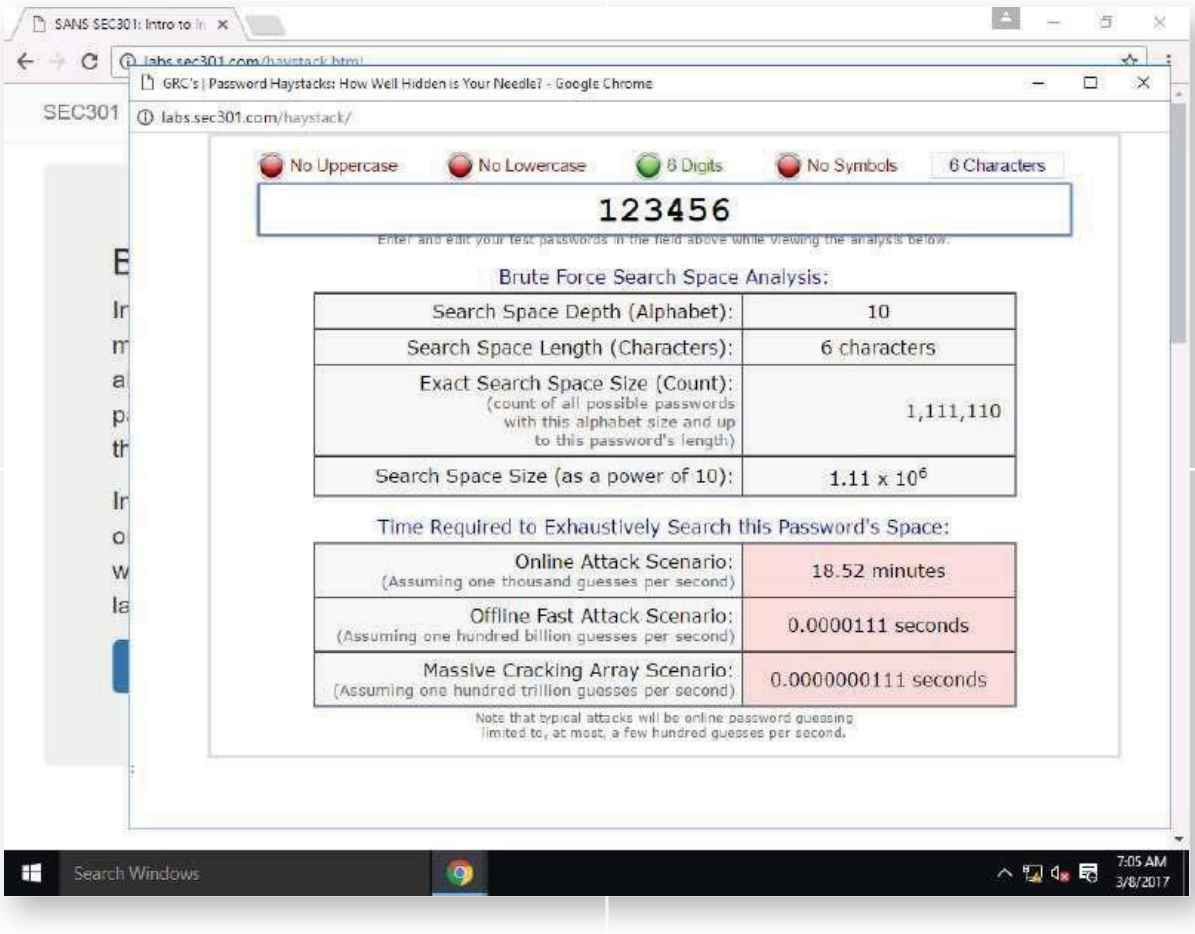


5. **Enter a Weak, Numeric Password**

   In the entry field for the Haystack application, enter the password string **123456**. This is the most common password in use today.

> ⚠ WARNING: At no time during the lab that follows should you enter a real password from your online banking, an ecommerce site (such as amazon.com), a credit card site, or from your office. While in the classroom, other students can see what you are typing!

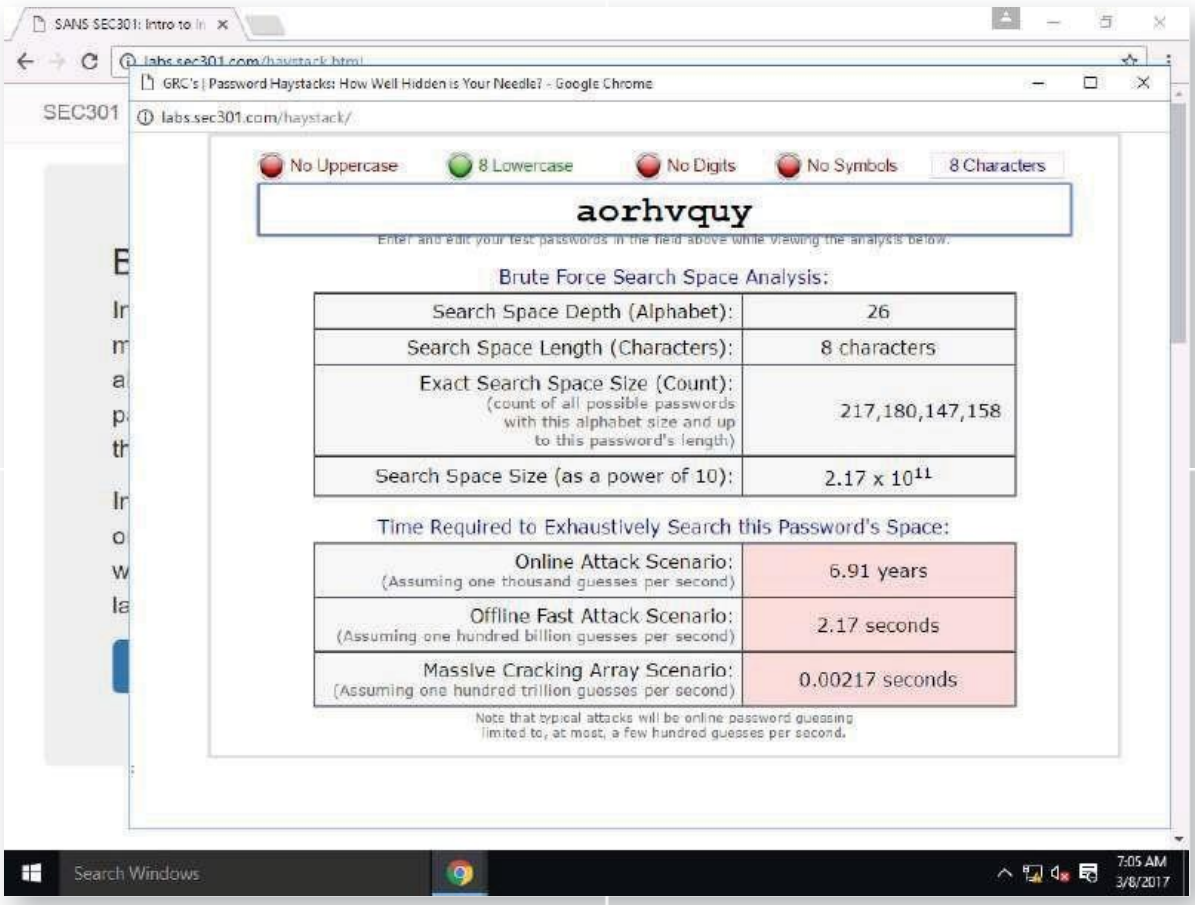
> As you type the string 123456, the Haystack application calculates the total number of characters that have to be guessed and how long it would take to do so given the speed in the three scenarios we described. Notice that you can see the green dot and "6 digits." This indicates that you are using only numbers and how many. The way the Haystack application works is that it assumes the attacker knows the character set you used to create your password. Because the vast majority of discovered passwords over the years have been either all numeric or all lowercase letters, it makes sense for the site to do it this way.



6. **Enter a Weak Password, Alphabetic**

   Change the password 123456 to one of your choosing. Make the new password 8 characters in length, all lower case letters. Examine the *Time Required to Exhaustively Search* metrics.

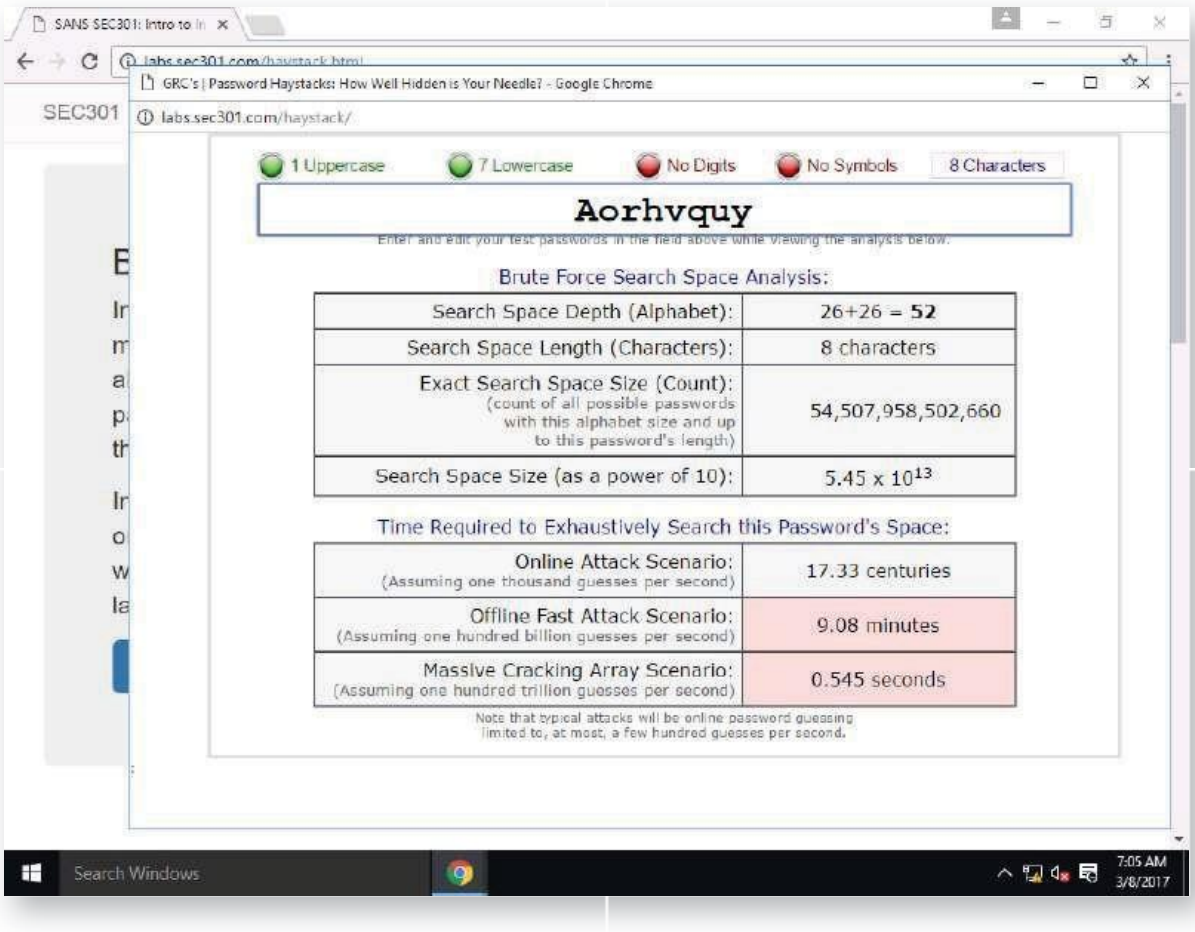
> When the password was a 6-digit value, the required time value was 18.5 minutes for an online attack, while the offline fast and massive cracking array attacks were both well below 1 second each. If you change the password to an 8-character, all lowercase, alphabetic string improves the time required metrics but is still easily within reach for an attacker.



7. **Change One Character to Uppercase**

   Reusing the previously selected password, change the first letter to the uppercase equivalent.

> When uppercase and lowercase letters are both used in a password, the search space (the number of potential passwords that a hacker has to try) doubles, since there are no longer only 26 potential letters, but 26 uppercase and 26 lowercase letters. This change dramatically alters the required time metric as well, with an online attack requiring 17.3 centuries to complete. However, the offline fast and massive cracking array attacks are still possibilities for an attacker.



8. **Change One Additional Character**

   Now change one of the other eight letters to a special character (such as the hash symbol [#]). Watch the times change as the potential search space increases further.

   Even with a reasonably complex password consisting of 8 characters and a combination of uppercase, lowercase, and special characters, the password still isn't strong enough to overcome an offline fast or massive cracking array attack.

9. **Add More Characters**

   Keep adding characters to the end of your password to figure out how many characters are needed to defeat all the password attacks (none of the boxes are red)

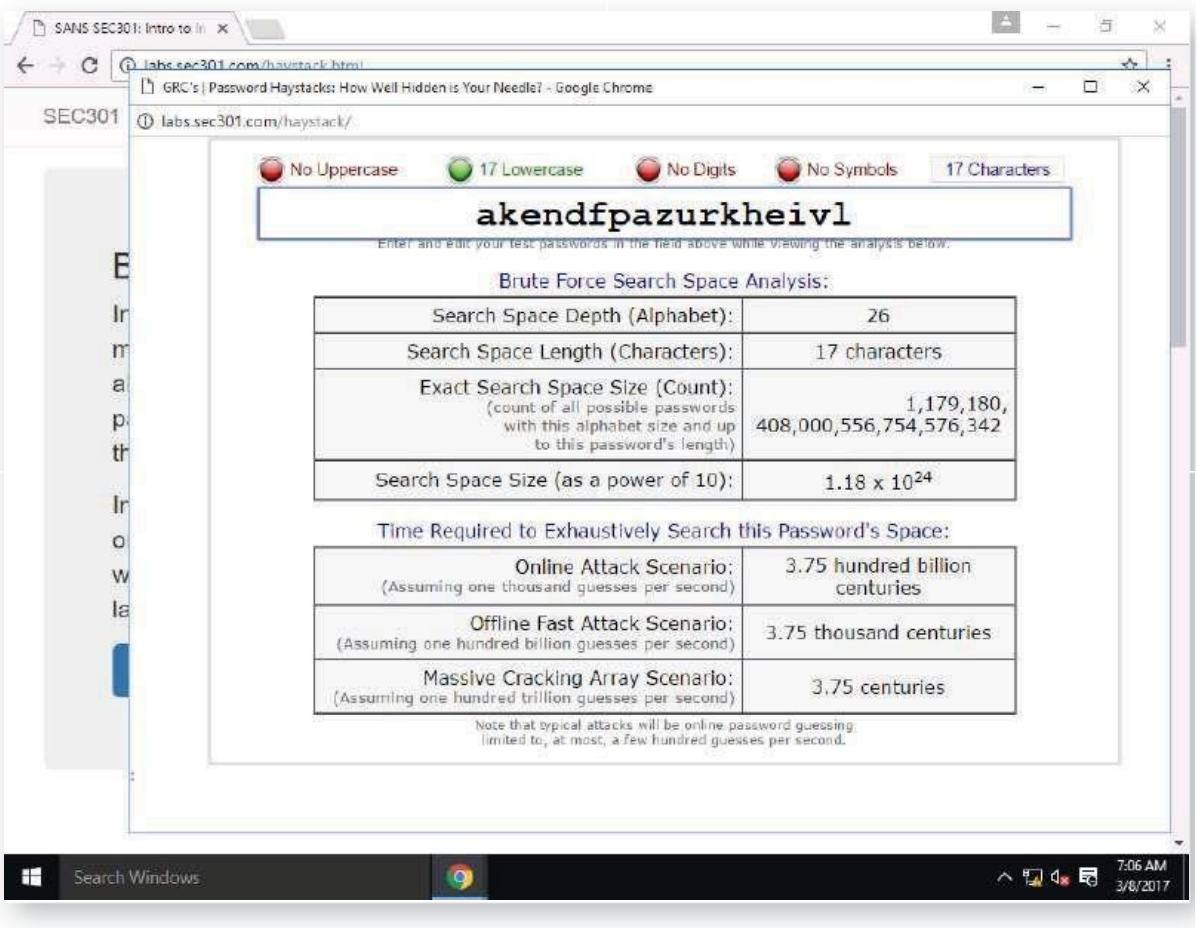
The offline fast attack continues to be an option for an attacker with passwords up to 10 characters in length. At 11 characters, an offline fast attack is no longer an option, but the massive cracking array remains possible. At 12 characters, with the complexity of uppercase, lowercase, and special characters, none of the attacks remain viable.

Note that these options are dependent on the characters you select for your password. Removing entropy will require longer passwords to defeat the attacks. Adding entropy will allow for shorter passwords.

10. **Alphabetic, Lowercase Only Password**

How many characters are necessary to defeat all of the attacks (make all of the red boxes turn white) using only alphabetic, lowercase characters?

Users would need to choose and remember a password of 17 lowercase letters to defeat all the attacks. Such a password is unlikely to be recalled quickly.



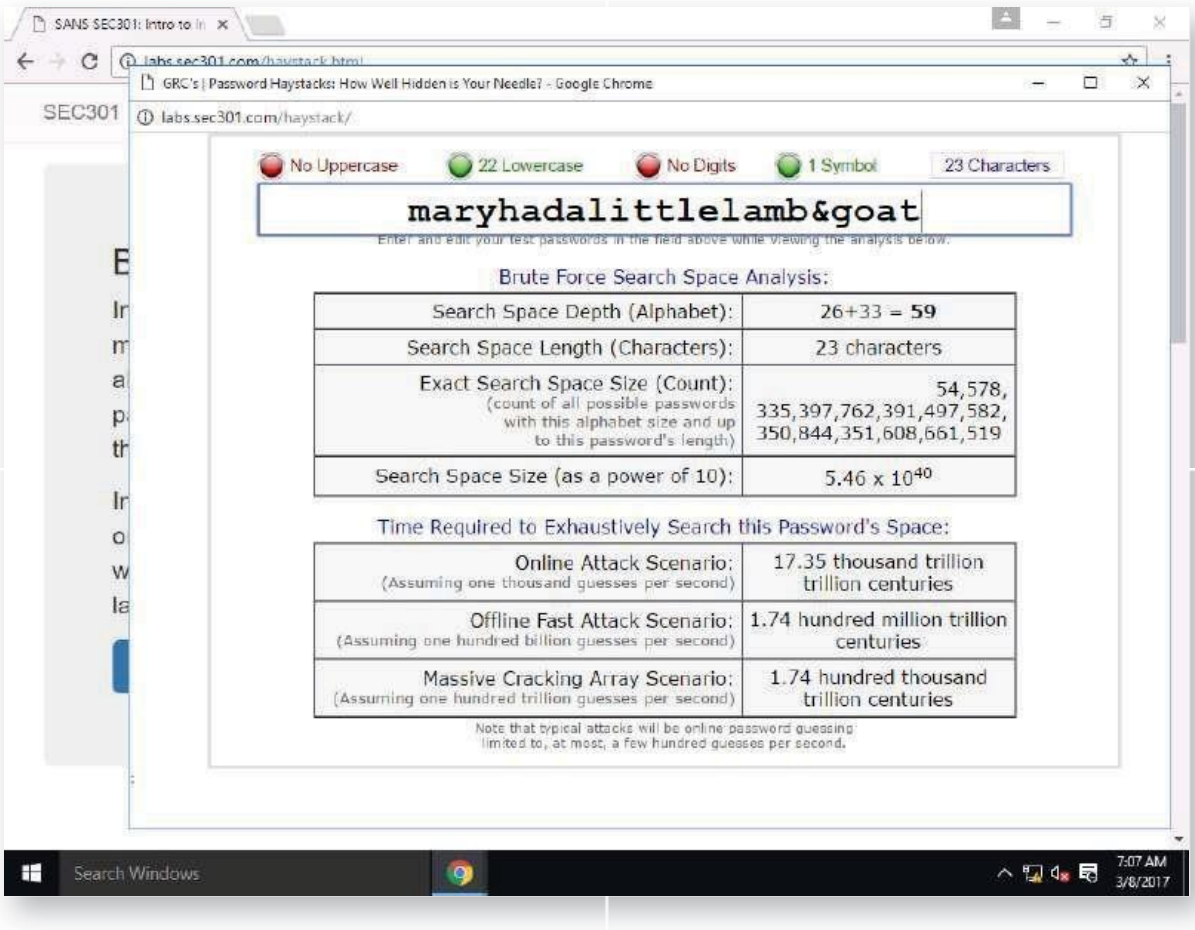
11. **Choose a New Password**

Enter a new password using a random combination of any characters. Keep entering characters until all of the attacks are defeated. How easy will it be to recall the entered password?

12. **Enter the Given Password**

    Enter the 23-character password: **`Maryhadalittlelamb&goat`** What is the fastest opportunity for an attacker to crack the password?

> Using a more memorable password such as the example used here is not only easier to remember, it is easier to type (since it consists of common words) and contains uppercase letters, lowercase letters, and special characters. The fastest attack option requires 7.66 hundred million trillion centuries to crack.



13. **Add Two Numbers**

    The password; **`Maryhadalittlelamb&goat`** is 23 characters, but lacks any numbers. Add the number 23 to the end of the password and observe the changes to the password cracking duration.

    The password cracking duration changes to 89.4 trillion trillion centuries. This increase in cracking time is due to the added password length, but also from the increase in the search space through the introduction of numbers in the password.

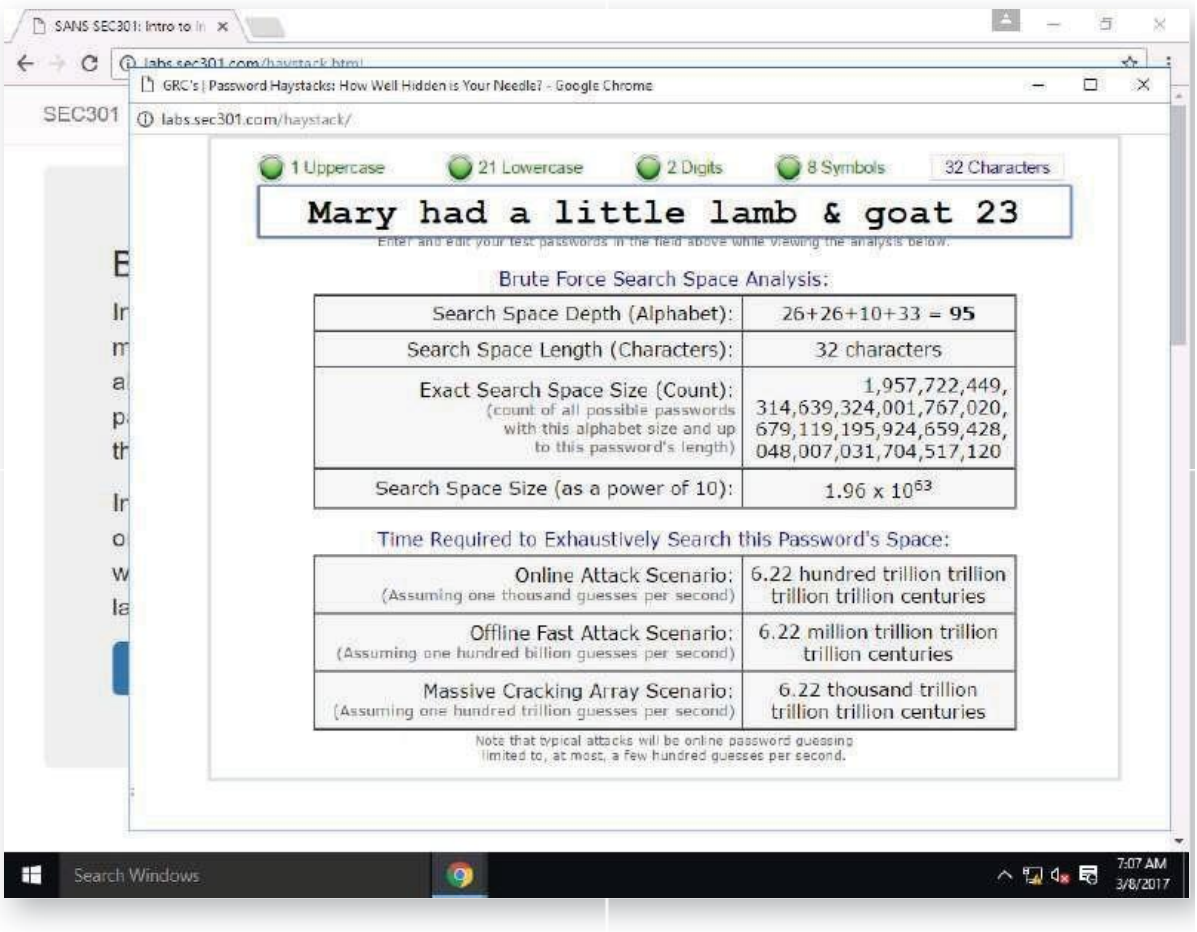
14. **Introduce Spaces**

    This time, type the string exactly as shown below, including the spaces, (the M is the first character, and the three is the last—there are no spaces before or after the string).

    `Mary had a little lamb & goat 23`

    How long would the new password take to crack?

To crack the password, it would take an attacker 6.22 thousand trillion trillion centuries.
Note that many websites do not allow for spaces in passwords, but some operating systems do (most Windows versions, for example). Anytime you are allowed to use spaces in your password, you indeed should.



Good passwords do not have to be impossible to remember and type to be good passwords. They do have to be unpredictable (also known as a high degree of entropy). Obtaining unpredictability (entropy) involves using sufficiently long strings—for example, 25 characters or longer—that are not likely to be in a dictionary or otherwise easily guessed.

As stated at the beginning of this lab, the "Haystack" page functionality comes from the Gibson Research web site with the permission of Steve Gibson. You can access this page at [http://www.grc.com/haystack.](http://www.grc.com/haystack) We recommend you show this page to your coworkers and family.

This page intentionally left blank.

SEC301-2.1: Exercise - Converting Number Systems & Decoding ASCII

Objective

This lab is designed to help students get a better grasp of the process of converting from one numbering system to another – decimal to binary, binary to hexadecimal, and so on. We will also obtain a better grasp of ASCII encoding. You'll work through the exercise steps to build your familiarity, then apply the techniques and tools to decode an encoded, secret message.

Scenario

Virtual Machines

1. SEC301-Win10
2. SEC301-LabServer

Cashbux - Decoding Numbering Systems

From the lecture, you recall the Dec-A-Bux, Bin-A-Bux, and Hex-A-Bux example. In this lab, we will continue that example. You fulfill the role of the world's unluckiest cashier. We present you with different numbering system examples, and you will complete the missing fields using the Dec-A-Bux, Bin-A-Bux, and Hex-A-Bux conventions.

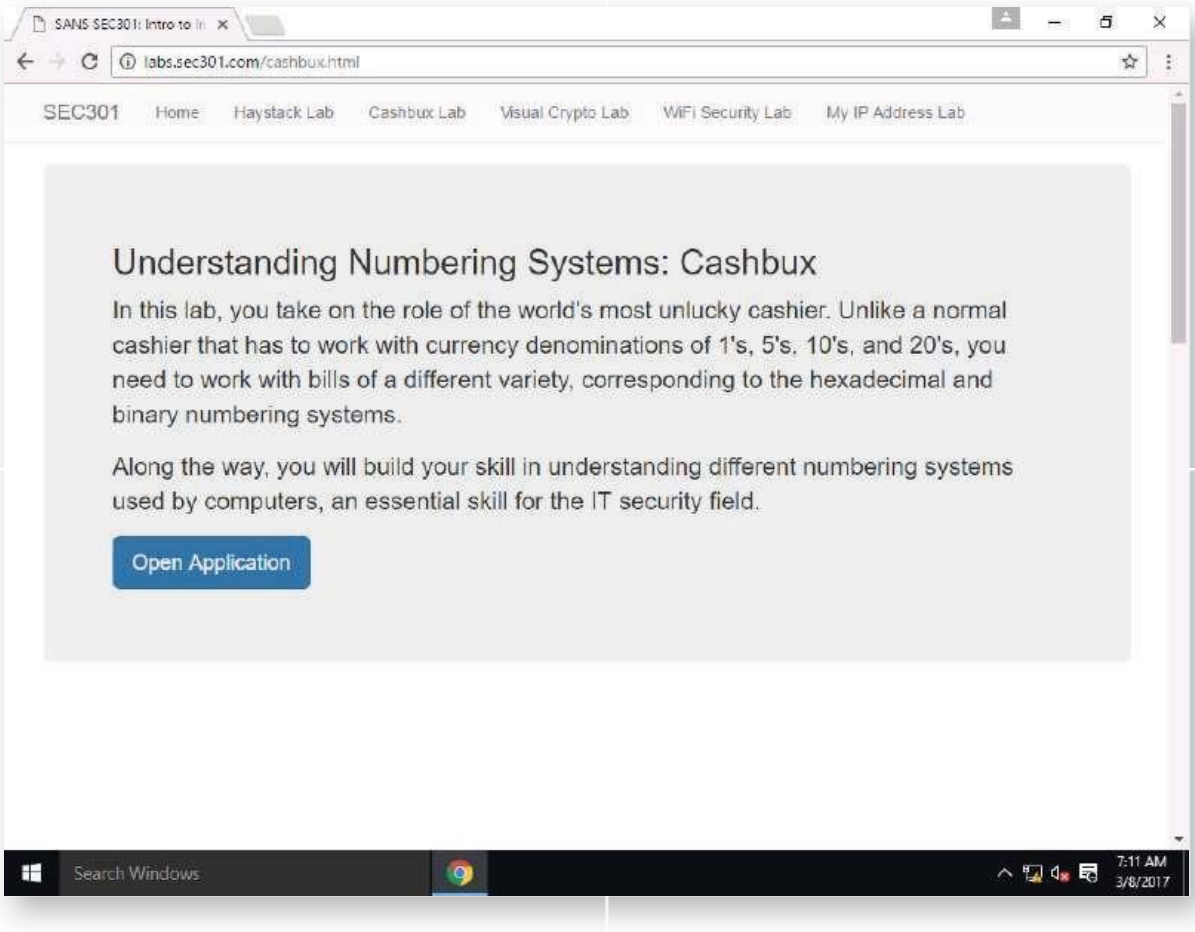
1. **Open Chrome**

   Launch Chrome by double-clicking on the desktop icon.

   ⚠ **NOTE:** In this lab, it is common for students to forget to click on Done after each step. While you can easily complete the lab this way, doing so means you will miss out. Several of the steps have Tips (the head in the lower-left highlights) that explain the answer.

2. **Open the Cashbux Lab Page**

   From the SEC301 lab server website, click on the tab near the top of the browser labeled **Cashbux Lab**.



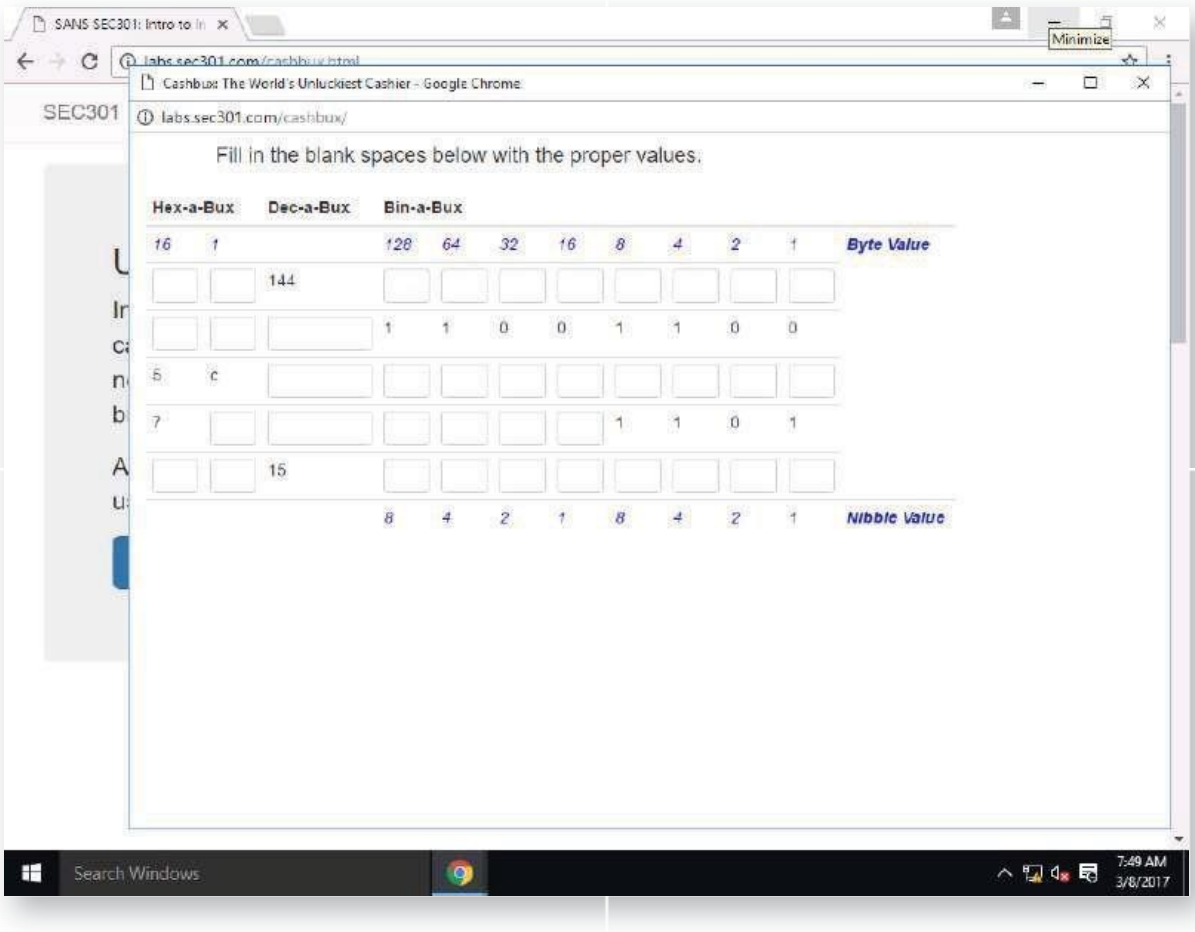
3. **Start the Cashbux Application**

   Read the lab explanation on the Cashbux lab page. When you are finished, click the **Open Application** button to start the Cashbux application.

4. **Examine Cashbux Form**

   The Cashbux application is designed to challenge you (a little) while helping to explain different numbering systems. Remember that the three major columns (Hex-a-Bux, Dec-a-Bux, and Bin-a-Bux) represent the three different numbering systems we discussed in the lecture (hexadecimal, decimal, and binary). As a cashier, you are asked to fill in the missing fields in each column to match the specified values using different numbering systems.

   Below the column headers (Hex-a-Bux, and so on) are the values in decimal for each of the columns (representing different currency in your cash drawer). As you fill in each of the missing values for each of the numbering systems, correct answers will be marked in green, while incorrect answers will be red. Incomplete answers (one or more missing values) are white.



5. **Complete Row 1 Hex-a-Bux**

   Fill in the two missing values in the first row for the Hex-a-Bux column. If you get stuck, click on the Knowledge icon for this task.

   In the first row, you see the Dec-a-Bux value of 144. Converting this to Hex-a-Bux, we would take 9 bills from the 16's compartment (9*16 is 144), and 0 bills from the 1's compartment.

   Enter a 9 in the 16's column for the first row, and a 0 for the 1's column. When you enter both values, the cells will turn green to indicate that you are correct.

6. **Complete Row 1 Bin-a-Bux**

   Fill in the eight missing values in the first row for the Bin-a-Bux column. If you get stuck, click on the Knowledge icon for this task. Remember that each field can only be a 1 or a 0.

For the Bin-a-Bux cash drawer, your currency is in the form of 128, 64, 32, 16, 8, 4, 2, and 1 dollar bills. To make 144 dollars, you need a 128 dollar bill, and a 16 dollar bill (128 + 16 = 144). Enter 1's in the 128 column and the 16 column. Enter 0's in the remaining columns.

7. **Complete Row 2 Dec-a-Bux**

   Fill in the missing value in the second row for the Dec-a-Bux column. If you get stuck, click on the Knowledge icon for this task.

   In the second row, you have the Bin-a-Bux value and need to convert it to Dec-a-Bux. Remember that each 1 adds the value at the top of the column to the total. This produces 204 (128 + 64 + 8 + 4). Enter 204 in the Dec-a-Bux column.

8. **Complete Row 2 Hex-a-Bux**

   Fill in the two missing values in the second row for the Hex-a-Bux column. If you get stuck, click on the Knowledge icon for this task.

   Now that you know the Dec-a-Bux value is 204, the next step is to convert it to the Hex-a-Bux value. You need 12x16 to make 192, plus 12x1 to reach 204. In other words, the first Hex column requires a value of 12 and the second hex column needs a value of 12.

   BUT - There is no 12 in hexadecimal. Instead, we continue counting after 9 with a (10), b (11), c (12), d (13), e (14), and f (15). So because C represents the numeric value 12, we can enter C into the first Hex column and C into the second Hex column.

   Cx16 = 192 + C = 204

9. **Complete Row 3 Dec-a-Bux**

   Fill in the missing value in the third row for the Dec-a-Bux column. If you get stuck, click on the Knowledge icon for this task.

   Here you are working with the Hex-a-Bux value and need to convert it to Dec-a-Bux. The first column in the Hex-a-Bux value is 5 -- multiply 5 by 16 to produce 80. Next, the second column is c, which as we saw in the previous row, is 12 in decimal. Add 80 and 12 together to produce 92. Enter 92 in the Dec-a-Bux column.

10. **Complete Row 3 Bin-a-Bux**

Fill in the 8 missing values in the third row for the Bin-a-Bux column. If you get stuck, click on the Knowledge icon for this task. Remember that each field can only be a 1 or a 0.

In this case, it's easier to work from the Hex-a-Bux value using the Bin-a-Bux nibble values than it is to work with the whole 8-bit value at once. The first nibble in the Hex-a-Bux column is 5. Using the Bin-a-Bux nibble values, that is the same as 0101 (0 + 4 + 0 + 1). Enter 0101 in the first four Bin-a-Bux fields.
The second nibble in the Hex-a-Bux column is C, or 12 decimal. Using the Bin-a-Bux nibble values, that is the same as 1100 (8 + 4 + 0 + 0). Enter 1100 in the remaining four Bin-a-Bux fields.

11. **Complete Row 4 Hex-a-Bux**

    Fill in the two missing values in the fourth row for the Hex-a-Bux column. If you get stuck, click on the Knowledge icon for this task.

Here you already have the high-order nibble for the Hex-a-Bux value. You only need to solve for the low-order nibble. The Bin-a-Bux low-order nibble is 1101, which is 8+4+0+1, or 13. 13 in decimal is d in hexadecimal. Enter d in the midding Hex-a-Bux column.

12. **Complete Row 4 Bin-a-Bux**

    Fill in the missing value in the fourth row for the Bin-a-Bux column. If you get stuck, click on the Knowledge icon for this task.

Given the high-order nibble of the Hex-a-Bux value (7), we can complete the missing Bin-a-Bux values. Here, 7 in hexadecimal is 0111 in binary (0+4+2+1). Enter 0111 in the Bin-a-Bux fields.

13. **Complete Row 5 Hex-a-Bux**

    Fill in the two missing values in the fifth row for the Hex-a-Bux column. If you get stuck, click on the Knowledge icon for this task.

Here again, we start with the Dec-a-Bux value, 15. 15 is less than the 16 in the Hex-a-Bux column, so the high-order nibble is 0. 15 in decimal is f in hexadecimal. Enter f in the low-order nibble field.

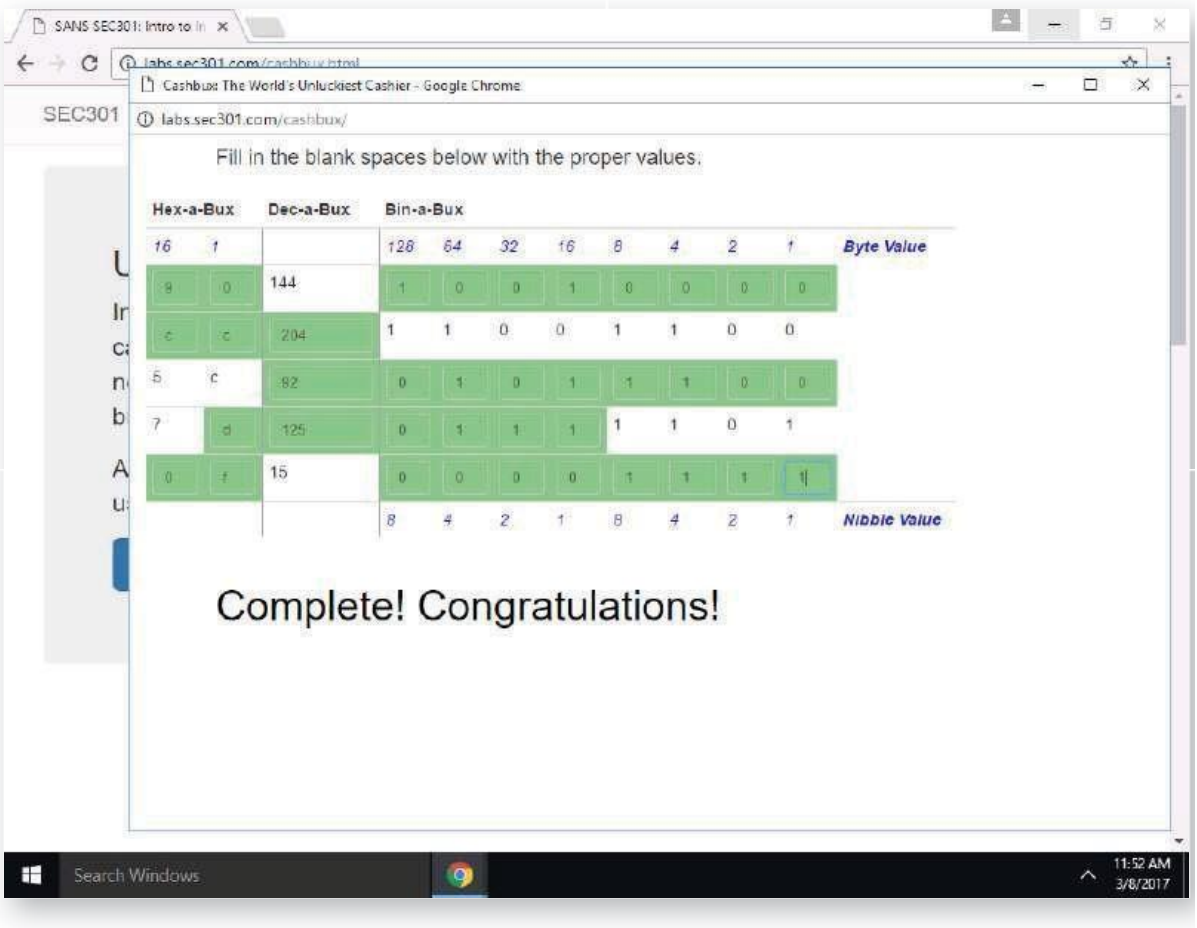
14. **Complete Row 5 Bin-a-Bux**

    Fill in the remaining missing values in the fifth row. If you get stuck, click on the Knowledge icon for this task.

Row five's Bin-a-Bux is probably the most straightforward part of the exercise. The high-order nibble of the Hex-a-Bux field is 0, so the first four bits are 0000. The low-order nibble of the Hex-a-Bux field is 15, which is the highest value that we express in one nibble. Since it is the highest value that we can represent, all the bits are on (or 1), producing 1111.

15. **Check Your Work**

    If you completed all the steps successfully, you will see a congratulatory message beneath the form. Compare your results to the screenshot in this step.



This lab is designed to help students gain a better grasp of the process of converting from one numbering system to another – decimal to binary, binary to hexadecimal, and so on. Having a strong understanding of these concepts is essential if you work in or around the Information Technology industry. Understanding these concepts will also be necessary for any more advanced courses you might attend.

SEC301-2.2: Exercise - Networking

Objective

Complete the steps in this exercise to build hands-on familiarity with networking settings including accessing command-line networking tools and identifying IP address, network mask, default gateway, DNS cache, ARP cache, and routing information. Change the settings controlled by a network router to examine the changes introduced by Network Address Translation.

Scenario

Virtual Machines

1. SEC301-Win10
2. SEC301-LabServer
3. SEC301-WIMIP
4. SEC301-SmallishRouter

Networking

Throughout the lecture, you have gained a solid understanding of basic networking. You know about IP addresses, network masks, default gateways, and Network Address Translation (NAT). In this lab, you discover how to find your computer's network configuration information and find out what (if any) address your internet traffic is NAT'ed to as it leaves the facility you are in.

1. **Open a Command Prompt**

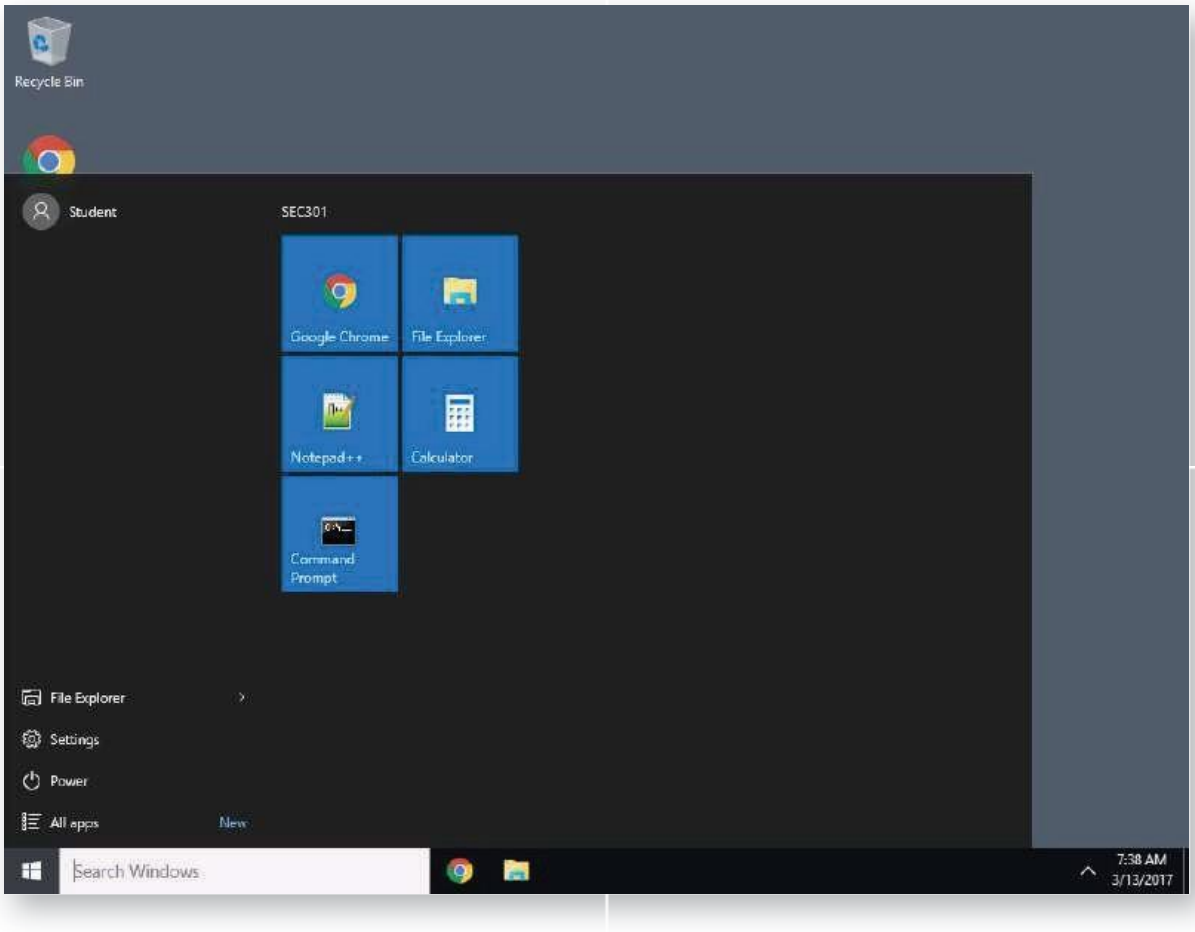
   In this lab, you work with a command prompt window. This enables you to enter commands directly into the operating system.

   To access the command prompt, click the Start button, then click on the **Command Prompt** icon.

   For this exercise, we have added the Command Prompt icon to the Start menu to make it quick and easy to launch. However, not all systems will have the Command Prompt pinned to the Start menu. You can always find the Command Prompt application on Windows by clicking the Start button, then enter `command` or `cmd` in the *Search Windows* dialog.

   For these exercises, we changed the Command Prompt colors to black text with a white background to make the content easier to read. Default Command Prompt settings use grey text with a black background.



2. **Examine Command Prompt**

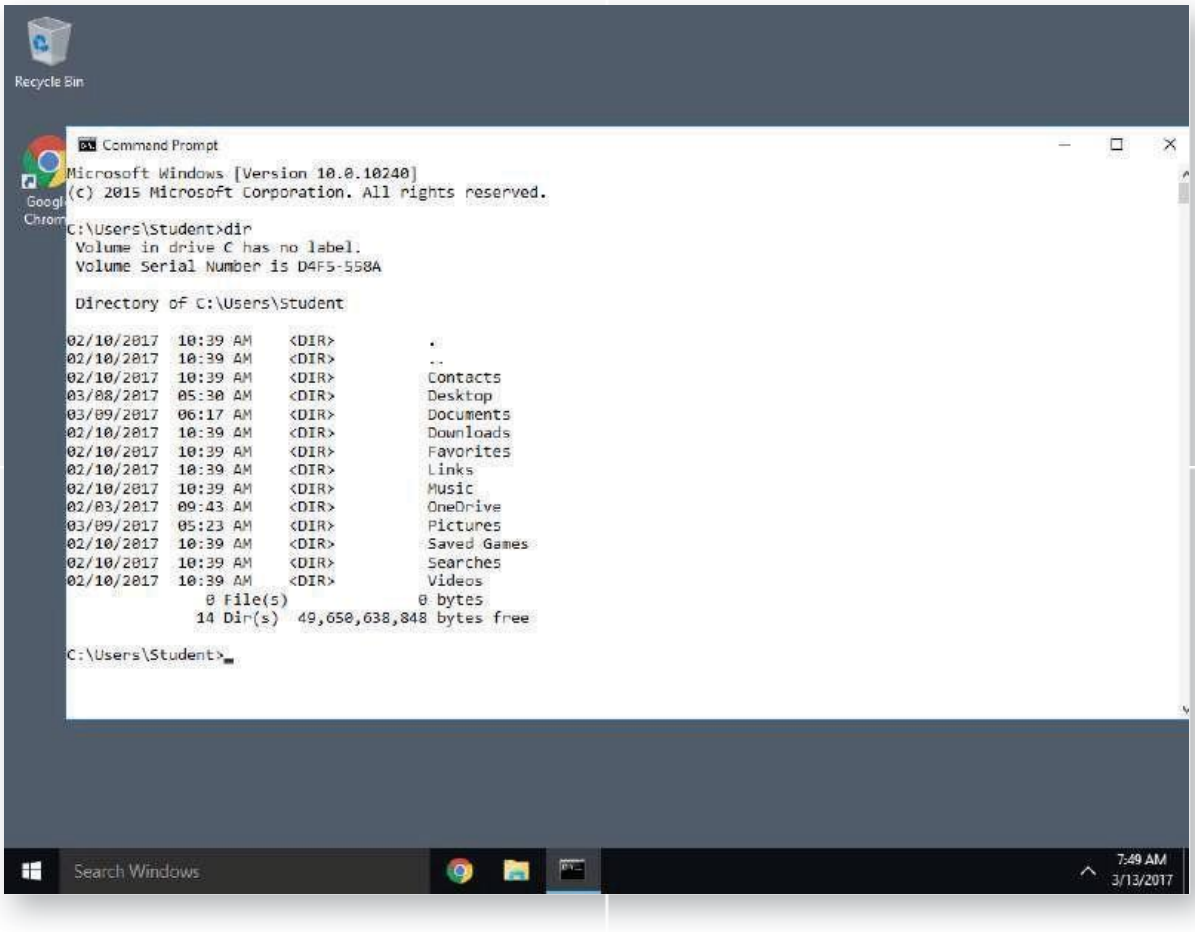
   From the Command Prompt window, you can run many different commands by typing the command name and then pressing Enter. Your computer interprets the commands you type and, if you type the commands correctly, displays information to you.

   The Command Prompt doesn't offer any guidance on what commands to run, nor does it explain what commands are available. External documentation (such as the steps in this exercise) and experience working with the Command Prompt are useful in getting valuable information from this tool.

   First, run a simple command to list files. Type the command, **`dir`** then press **Enter**, as shown below:

   ```
   C:\Users\Student>dir
   ```

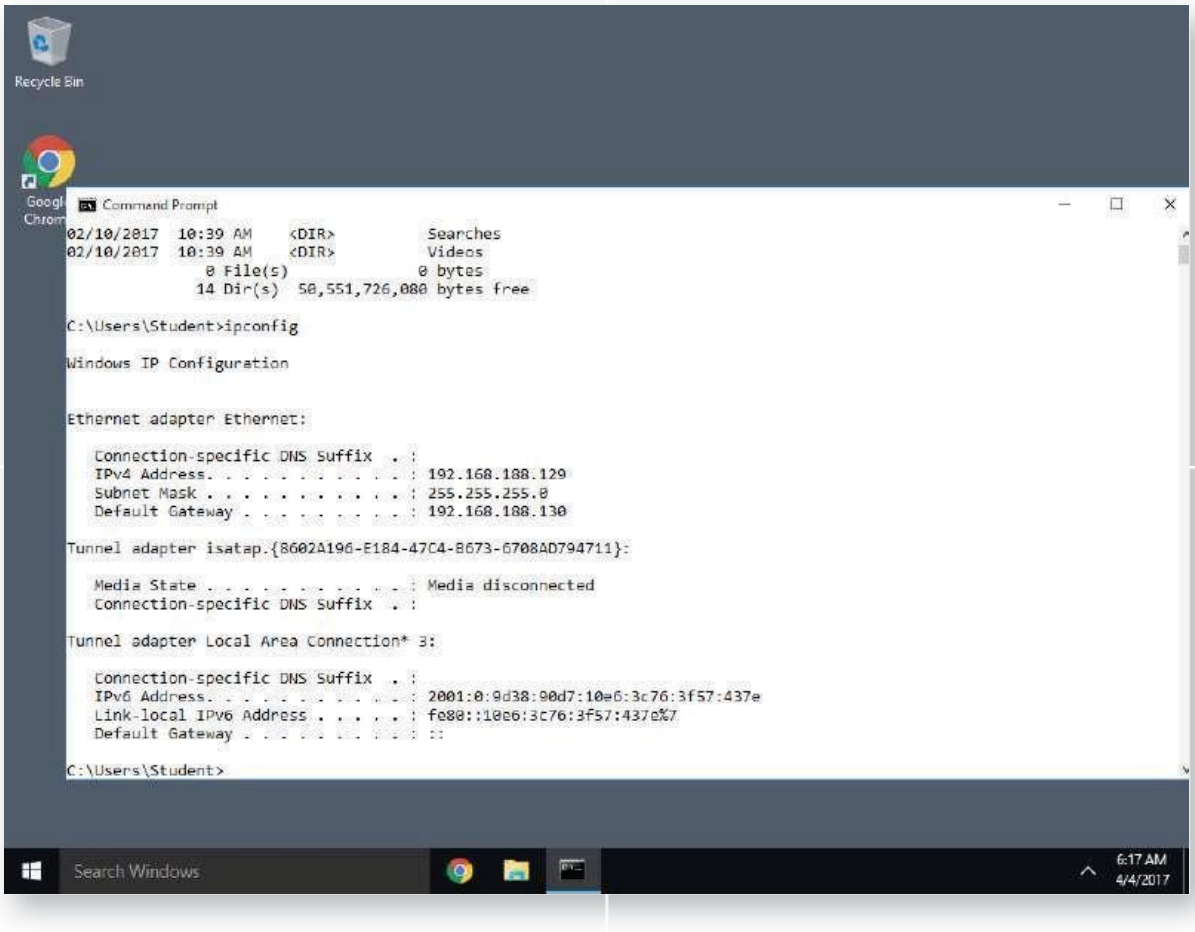
   The `dir` command lists the files in the current directory, or a different directory when specified.



3. **Examine Summary Network Information**

   Continuing to work from the Command Prompt, run the **`ipconfig`** command to list summary information about the local network adapters on the Windows system, then press **Enter**:

   ```
   C:\Users\Student>ipconfig
   ```



4. **Examine Ipconfig Output**

   Examine the output from the `ipconfig` command, identifying the following information:

   - IP Address
   - Subnet Mask
   - Default Gateway

> ⚠ Be sure to look at the Tip's throughout this lab for extra explanation. For example, there is a Tip for this task that explains the output you see.

The output from the `ipconfig` command reveals several details about the Windows system's networking configuration. First, we see that the network adapter is Ethernet based, and has the name *Ethernet* (`Ethernet adapter` **`Ethernet`**). We also see that there is no DNS suffix (the system does not have a DNS domain name associated with the network settings), and has an IP address of 192.168.188.129. The network mask is 255.255.255.0, and the default gateway is 192.168.188.130.

There is a second network adapter also present on the system called *Tunnel adapter isatap.{860...711}*. This second network adapter is a virtual network adapter used for encrypted Virtual Private Network (VPN) connections. For simplicity, just ignore this unused adapter for the exercise.

A third adapter is also present called *Tunnel adapter Local Area Connection* 3* (or similar). This adapter's configuration is for IPv6 access using an IPv6-in-IPv4 tunneling service.

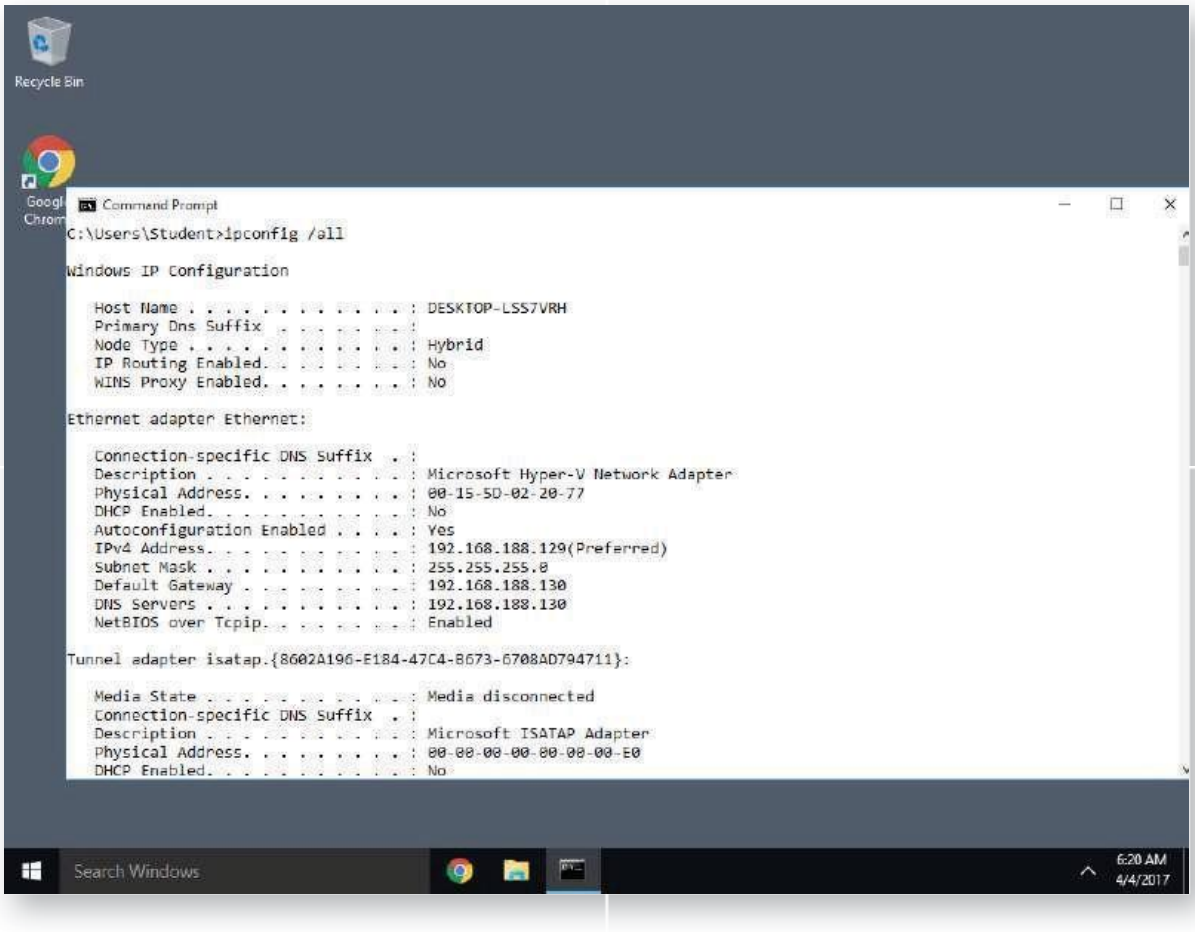
Again, for our purposes here, we are only looking at the **Ethernet adapter**.

5. **Examine Detailed Network Information**

   The output from the `ipconfig` command is brief and doesn't give us any information about DNS settings, MAC address information, or other interesting network configuration details. We can add an additional argument to the `ipconfig` command (`/all`) to show additional details.

   Type the `ipconfig` command again, this time adding the `/all` argument to retrieve detailed network configuration information, then press Enter.

   ```
   C:\Users\student>ipconfig /all
   ```



6. **Examine Detailed Ipconfig Output**

   Using the detailed `ipconfig` output, answer the following questions:

   - Is the system configured to use the Dynamic Host Configuration Protocol (DHCP)?
   - Is the system configured to use a DNS server?
   - Are the DNS server and the default gateway two different devices?

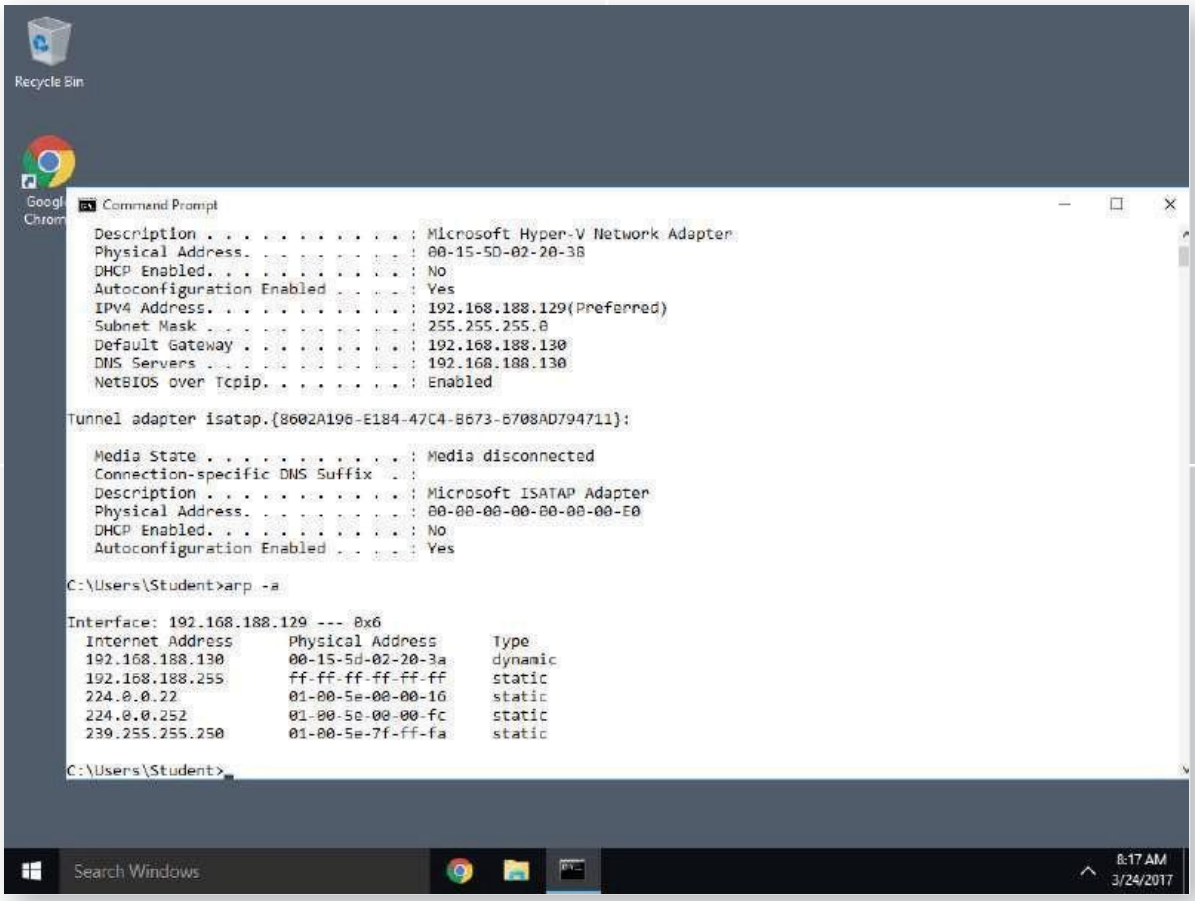
   Running **`ipconfig`** with the **`/all`** argument reveals detailed network configuration settings. From the output, we see that the system is not configured to use DHCP (it uses a static, or manually-assigned IP address), and it is configured to use DNS (denoted by the presence of the DNS server's IP address in the network settings). Further, we see that the IP address of the DNS server and the default gateway are the same, indicating that the default gateway is also offering DNS services.

7. **Examine ARP Information**

Next, examine the cached Address Resolution Protocol (ARP) information. Enter the `arp` command with the -a argument, as shown, then press **Enter**.

```
C:\Users\Student>arp -a
```

The output of the arp -a command shows the IP address to MAC address mapping information. Here we see a two client IP addresses (192.168.188.130, the default gateway), a broadcast address (192.168.188.255) and several multicast addresses (used for communicating to groups of systems).



8. **Examine DNS Cache Information**

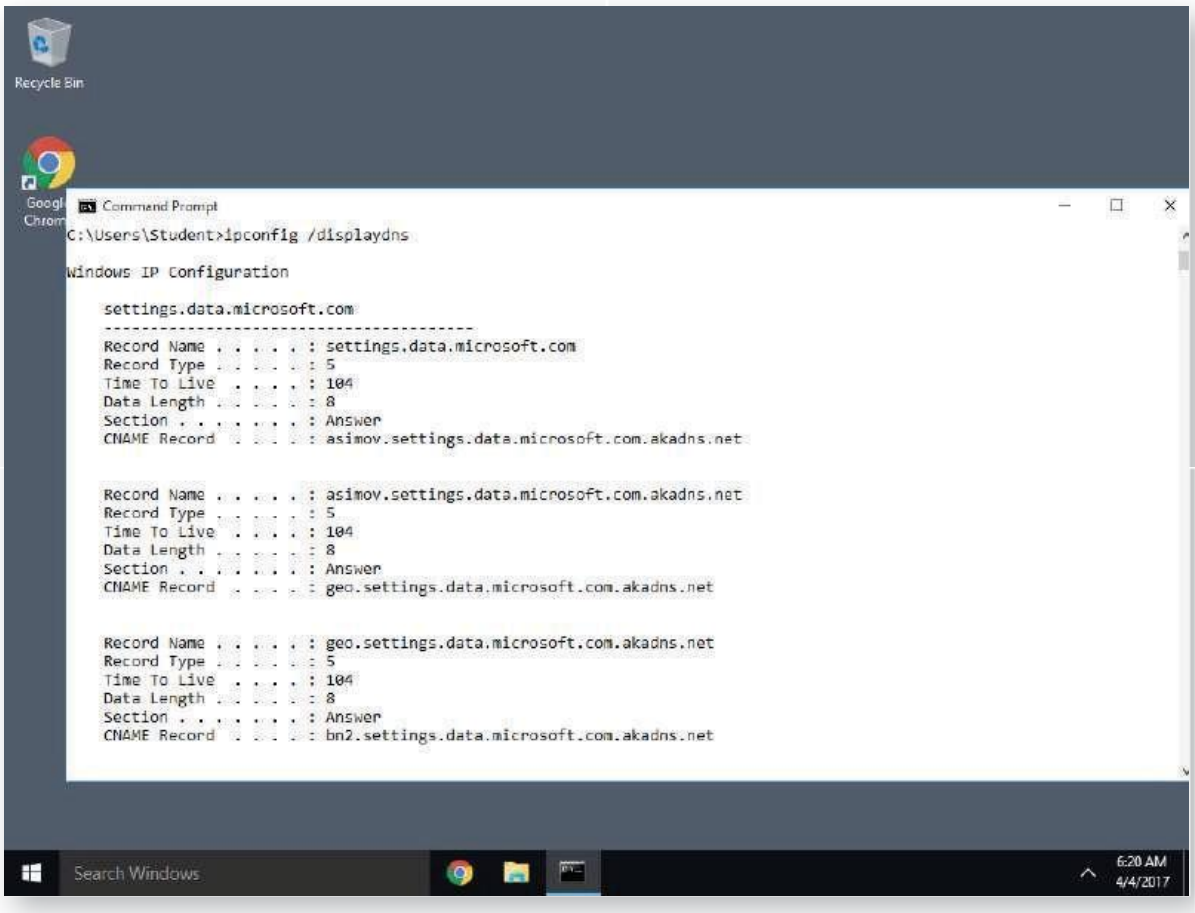
When your system looks up an IP address associated with a hostname, it stores a cached entry that can be reused again until the cache duration period expires. You can examine this information by running the `ipconfig command` again, this time with the addition of `/displaydns`.

From the Command Prompt, enter the **`ipconfig`** command with the **`/displaydns`** argument, then press **Enter**:

```
C:\Users\Student>ipconfig /displaydns
```

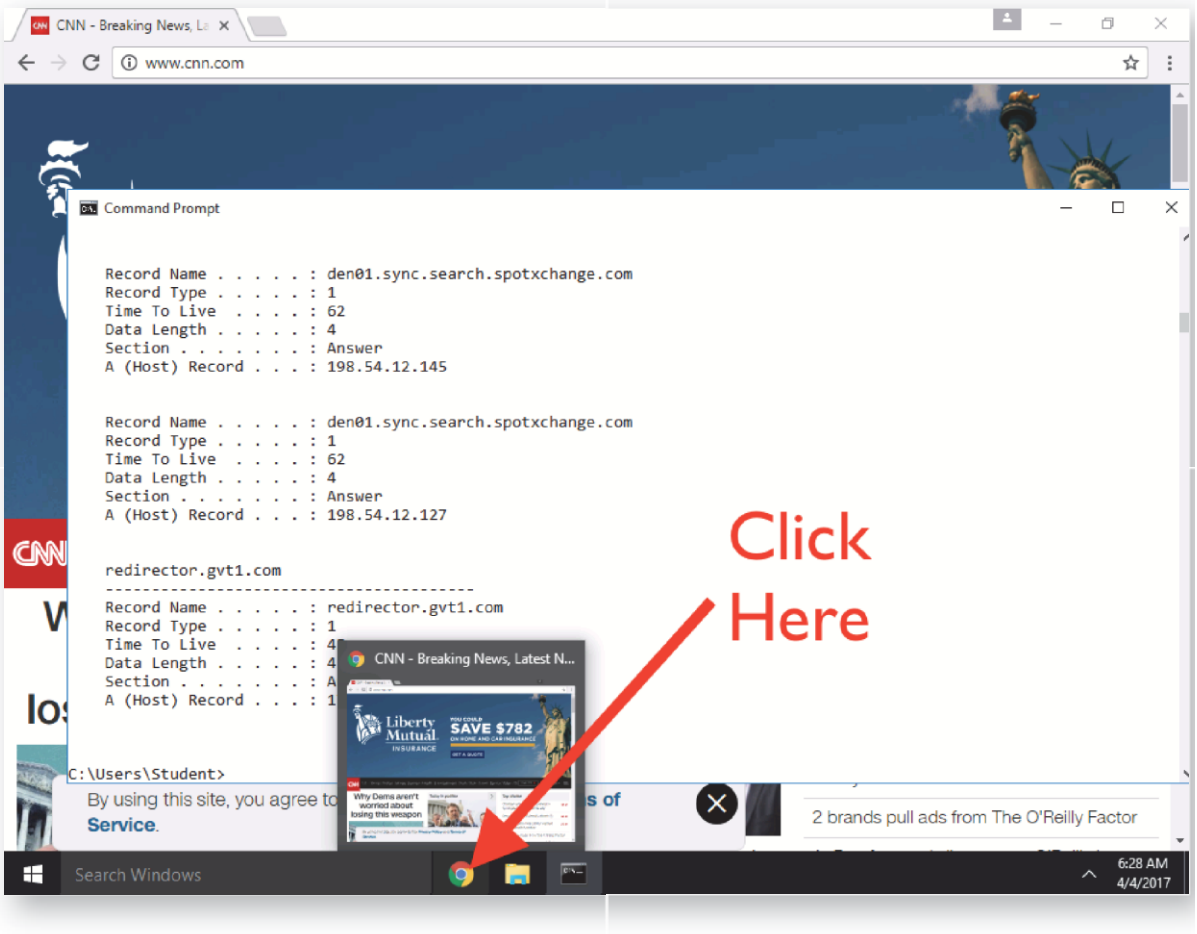

In this example, the command returns several DNS cache entries associated with systems used for the SEC301 lab exercises in LODS. The Windows 10 system recognizes these sites and prepopulates the DNS cache with the entries, even if you haven't yet connected to these sites.

Other sites, such as **microsoft.com** and **google.com** load automatically via the DNS process as you attempt to access them.



9. **Open the Chrome browser**

Open the Chrome browser by clicking the Chrome browser icon on the application taskbar to the right of the Start menu (as shown in the screenshot for this task).



10. **Visit whatismyip.sec301.com**

    From Chrome, browse to http://whatismyip.sec301.com.

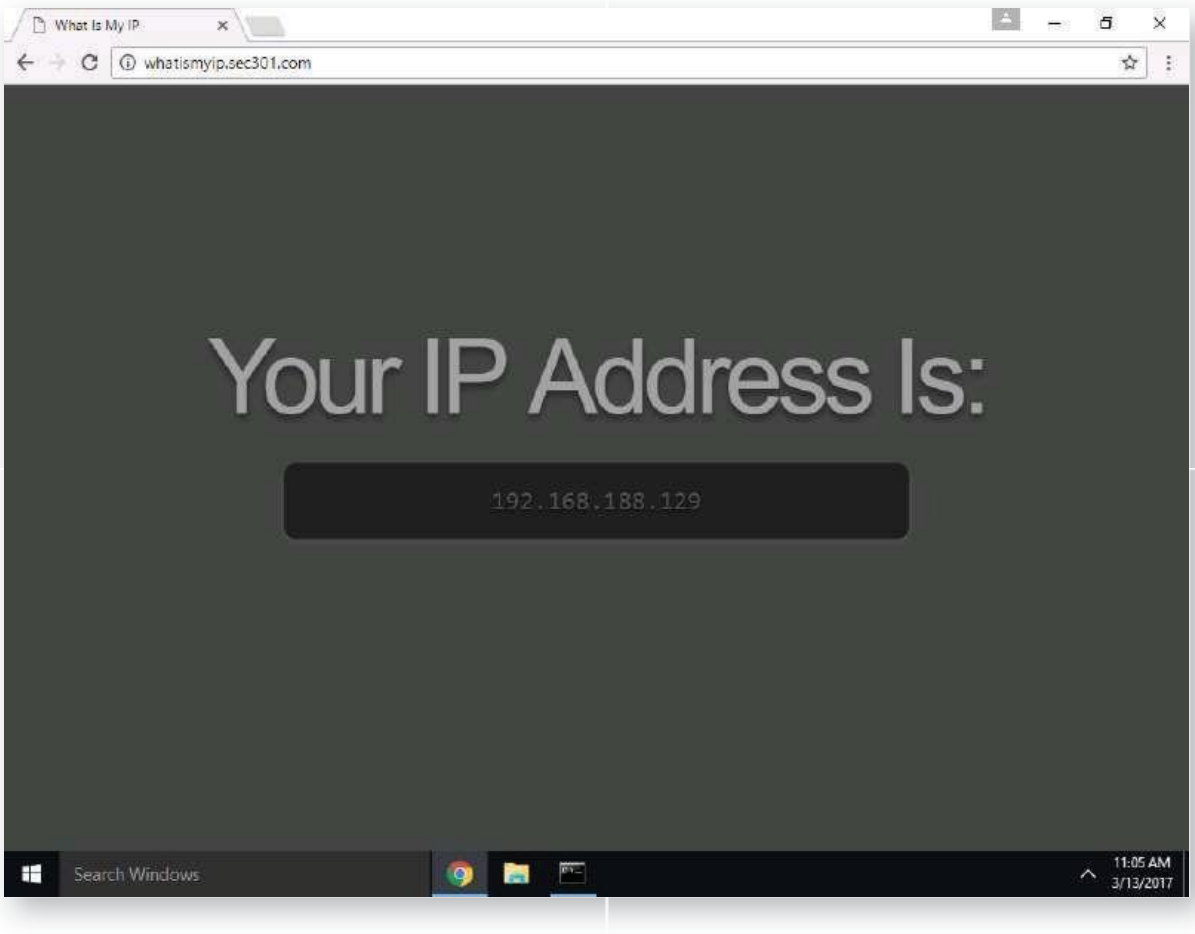
    You can view your public IP address when you are not working in the LODS environment.

    Simply go to www.google.com and type in "what is my ip address". Google will tell you the IP address that everyone on the Internet sees you coming from.

11. **Identify IP Address**

The server at whatismyip.sec301.com mimics popular online sites such as canihazip.com and whatismyipaddress.com or the even easier Google search for "What is my IP address". The server reports the same IP address that you saw in the `ipconfig` output: 192.168.188.129.

The fact that Whatismyip.sec301.com and the `ipconfig` command provide the same IP address tells us that Network Address Translation is not currently in use. In the following tasks, you will change this.



12. **Browse to linksnips.sec301.com**

    In the Chrome browser, browse to the http://linksnips.sec301.com site.

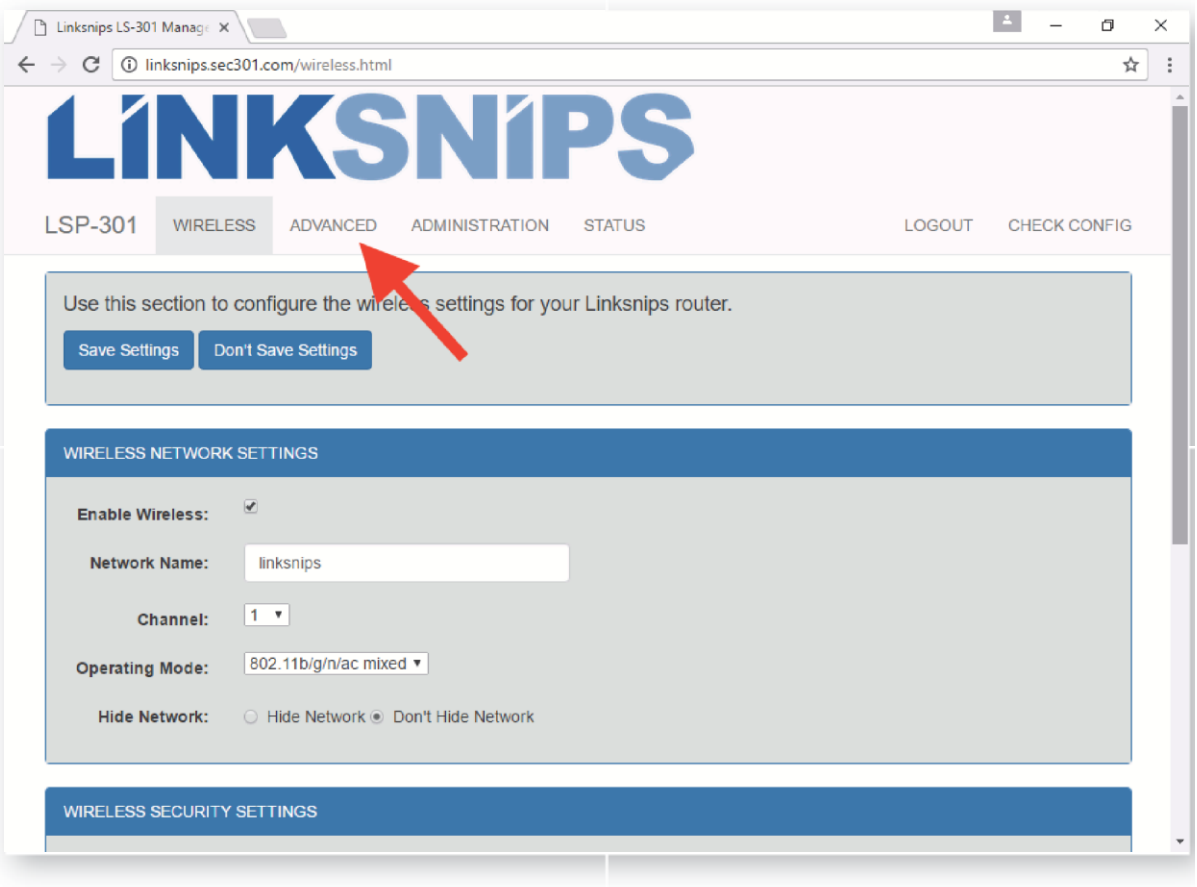
    ⚠ In this task, you will use our "Linksnips" wireless router interface and change a configuration (turn on NAT). You may have a curiosity about the rest of the configuration options available. In a later lab, you configure the Linksnips router from its default, insecure configuration into a secure configuration.

13. **Login to Router**

In the Chrome browser, login to the web page at linksnips.sec301.com. Enter the password **admin**, then click the **Login** button.

14. **Click on the Advanced Tab**

    In the Linksnips router configuration application, click on the ADVANCED menu.



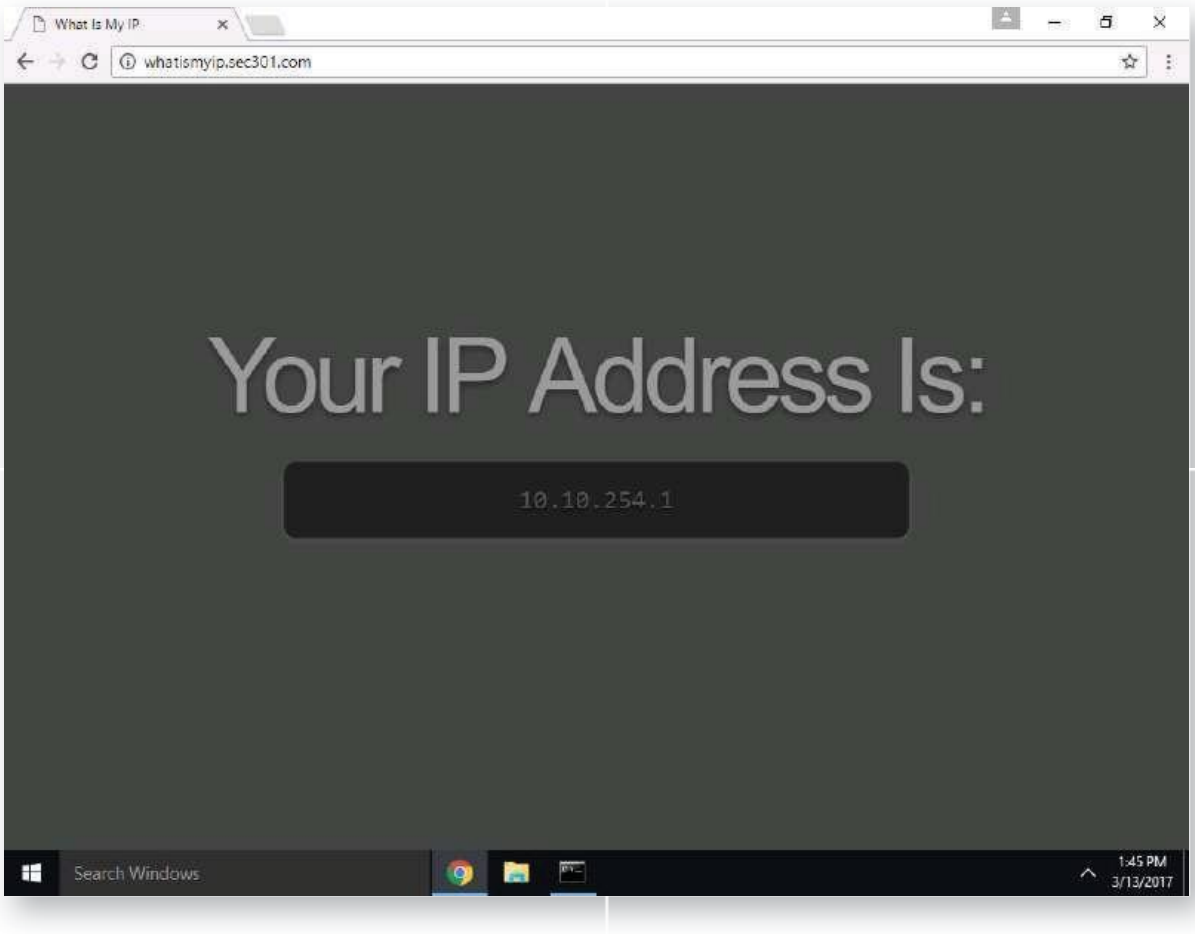
15. **Turn NAT On**

    Scroll to the bottom of the Linksnips advanced configuration options and turn on the **NAT/PAT** service. After turning the feature on, scroll back to the top of the screen and click the **Save Settings** button.

16. **Return to whatismyip.sec301.com**

    From Chrome, browse to [http://whatismyip.sec301.com.](http://whatismyip.sec301.com)

> After turning NAT on, whatismyip.sec301.com reports your IP address as 10.10.254.1 (which is the external IP address of the Linksnips router).
>
> Through the use of NAT, one or more clients can share the same external IP address for network access. Feel free to continue testing, turning NAT off and saving the configuration, then returning to whatismyip.sec301.com again to see the effect of the change.



In this exercise, you used command-line tools to examine many of the networking settings associated with a Windows system. Identifying the IP address, network mask, and gateway are common tasks for troubleshooting, network configuration, and security auditing and analysis. Examining DNS cache, ARP entries, and routing information are less widely known, but very useful for a variety of analysis needs including forensics and incident response.

Looking at the settings associated with the network router, you can turn NAT on and off. Using the remote website reporting your IP address, you can see the effect of NAT, masquerading your actual IP address behind a router.

Congratulations on completing this exercise!

SEC301-3.1: Exercise - Crypto by Hand

Objective

Encrypting and decrypting data by hand is slow, but it gives you tremendous insight into what happens in cryptosystems, and how different encryption and decryption techniques work. In this exercise, you will apply the skills that you learned in the lecture to gain first-hand experience in working with cryptosystems.

Scenario

Virtual Machines

1. SEC301-Win10
2. SEC301-LabServer

Crypto by Hand

In this lab, you will work through a series of cryptography challenges to ensure you develop a keen understanding of the basics of cryptographic functions. At times, this may seem a bit like the Little Orphan Annie decoder ring scene in the movie A Christmas Story. However, that indeed is not the intent. By completing these tasks, you will gain an understanding of the foundational elements of modern cryptography.

In this exercise, you will begin by working with permutation ciphers and substitution ciphers. Next, you will work with poly-alphabetic ciphers. Finally, you will have the opportunity to practice using the Vigenere Cipher to accomplish triple encryption, similar to, but much more straightforward than what Triple DES does.

1. **Open Chrome**

   Launch Chrome by double-clicking on the desktop icon.

2. **Open the Crypto by Hand Lab Page**

   From the SEC301 lab server website, click on the tab near the top of the browser labeled **Crypto by Hand Lab.**

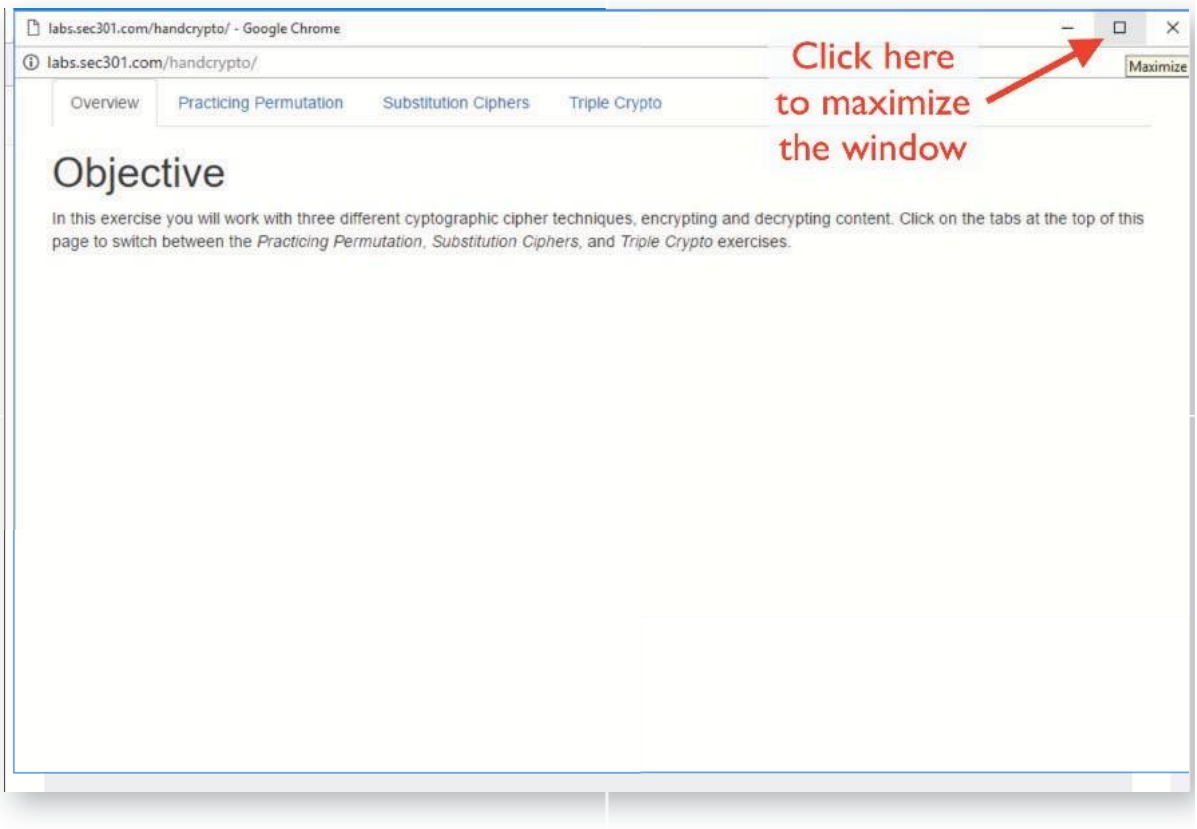
   ⚠ Notice there are two Crypto lab tabs. Be sure you click on "Crypto by Hand Lab" for this exercise.

3. **Start the Crypto by Hand Application**

   Read the lab explanation on the Crypto by Hand Lab page. When you finish, click the **Open Application** button to start the Crypto by Hand application.

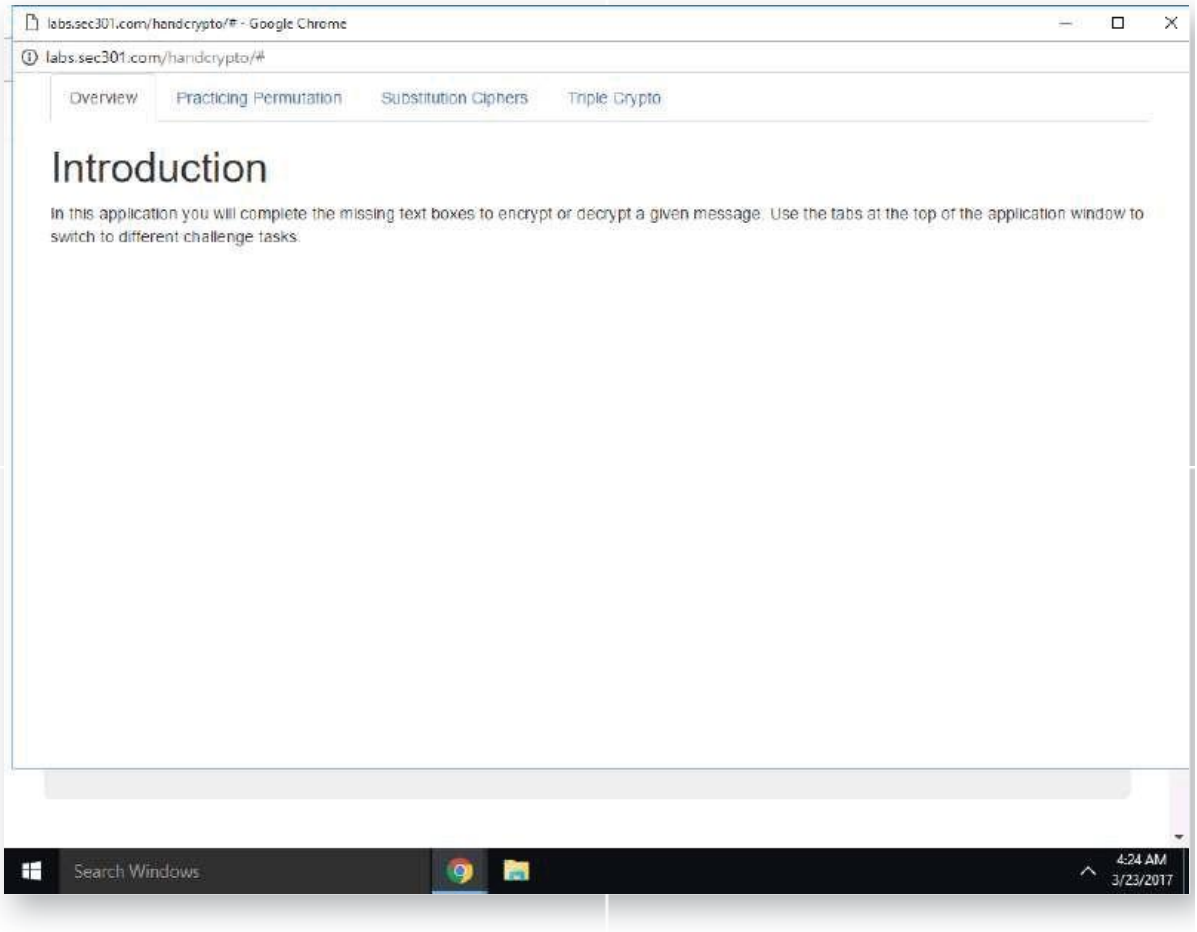
4. **Maximize Crypto by Hand Browser Window**

To maximize the available screen space for this busy exercise, maximize the Crypto by Hand window by clicking on the maximize button, as shown in the screenshot.



5. **Read Introduction Page**

   The Introduction page for the Crypto by Hand application points out the tabs at the top of the screen. In this exercise, you will complete three cryptography challenges using these tabs.

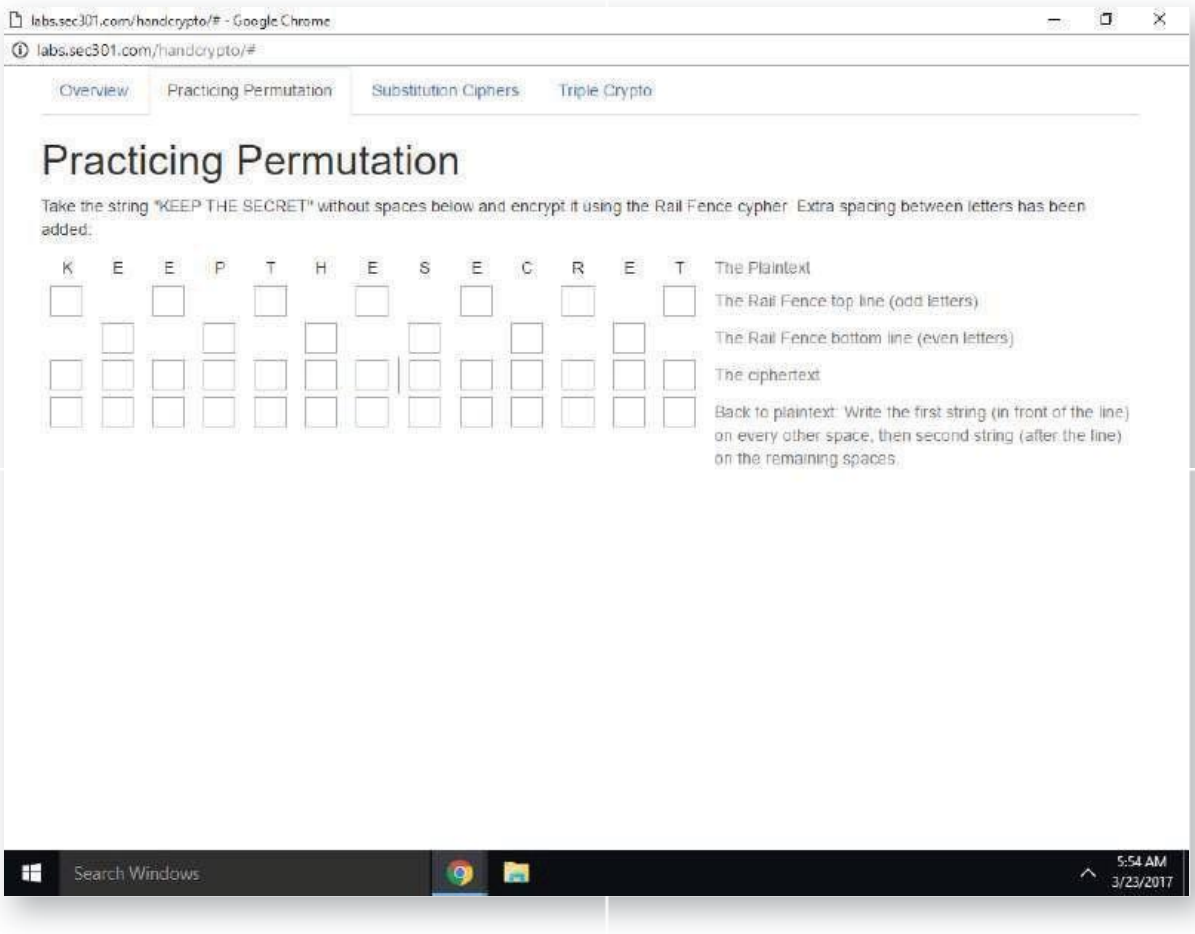


6. **Click on the Practicing Permutation Tab**

   For the first task, you work on the *Practicing Permutation* challenge. Click on the tab labeled **Practicing Permutation** to start the first task.

7. **Practicing Permutation Overview**

   First, you are going to use the Rail Fence cipher. You encrypt a string and then decrypt the string. As you hopefully recall from the lecture, to use the Rail Fence Cipher, you first picture in your mind a rail fence (or the more modern picket fence). You take a string of letters that make up the plaintext, then "throw the letters against the rails of a fence." Every other letter will strike a rail of the fence and stop, while every other letter keeps going between the rails. In other words, you wind up with a top line of every other letter beginning with the first (the odd numbered letters) and a bottom line of every other letter beginning with the second letter (the even numbered letters). You then take the top line, write those letters, then take the bottom line and write those letters after it—you separate the two strings with a line to keep them separate.



8. **Enter the Rail Fence Top Line**

   Fill in the boxes in the Rail Fence top line using every other letter of the plaintext string "KEEP THE SECRET." When you enter a letter correctly, the input box will change color to green. If you enter incorrectly, the input box will change color to red.

9. **Enter the Rail Fence Bottom Line**

   Fill in the boxes in the Rail Fence bottom line using every even letter of the plaintext string "KEEP THE SECRET" (where "K" is letter 1).

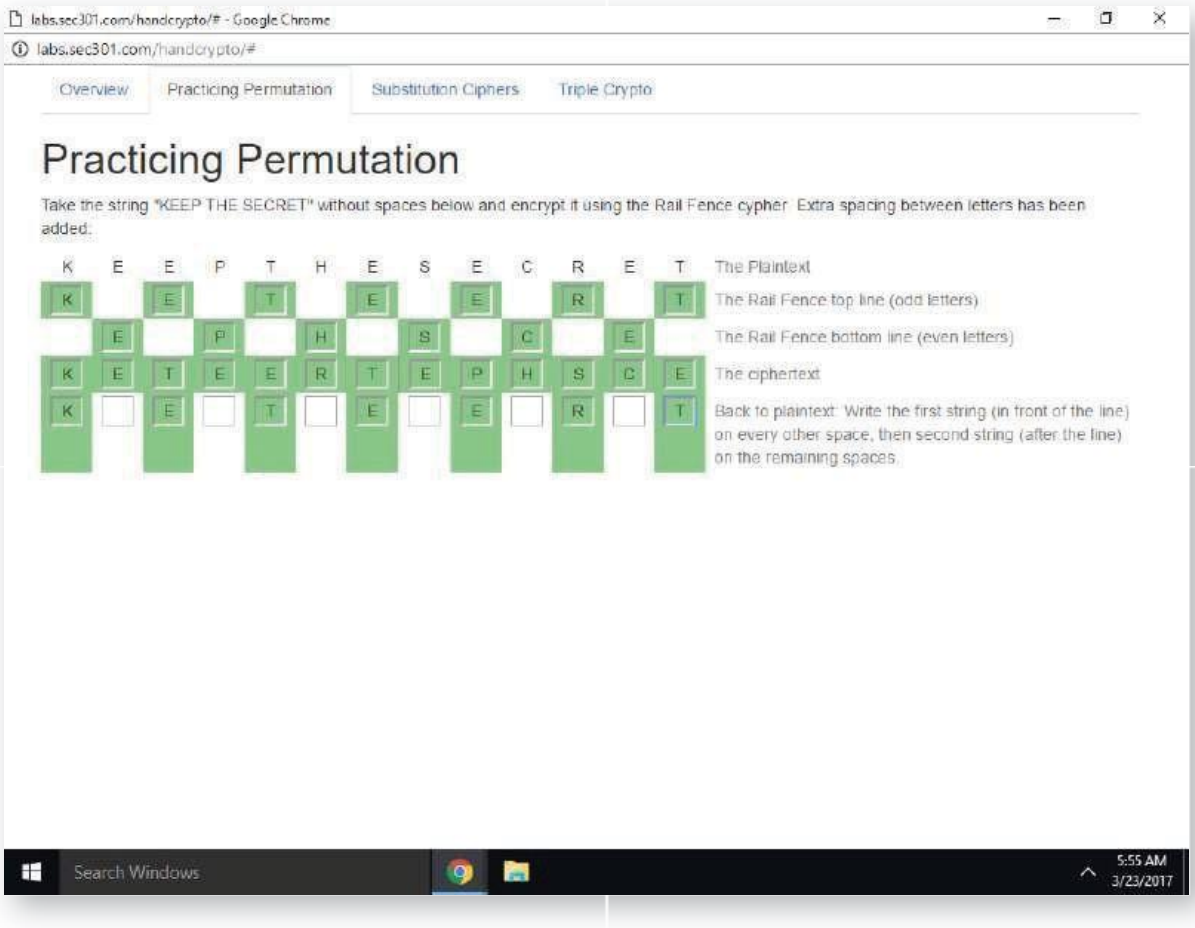
10. **Enter the Ciphertext**

    Next, combine the two rail fence lines, first entering the top rail fence line, followed by the bottom rail fence line in the ciphertext line.

    You have encrypted your first message, congratulations! The ciphertext is the content delivered to the recipient.

11. **Create First Half Plaintext**

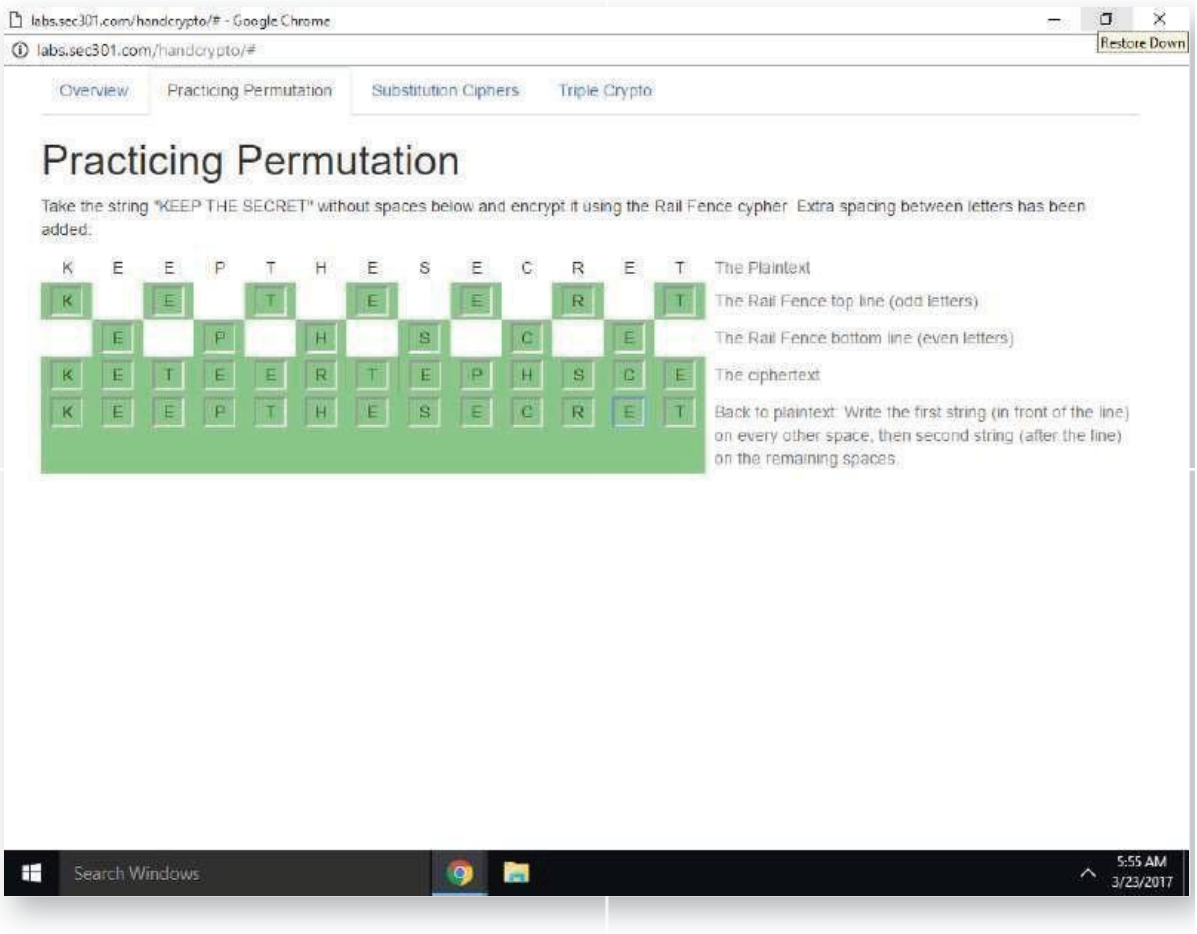
Next, you decrypt the ciphertext message. Using the ciphertext on the 3rd line, enter the first seven characters (KETEERT) in every other plaintext box, as shown in the screenshot.



12. **Complete Second Half Plaintext**

    Complete the decryption process by taking the 2nd half of the ciphertext (EPHSCE), entering the remaining letters in the available boxes.

> Congratulations, you have decrypted your first message!



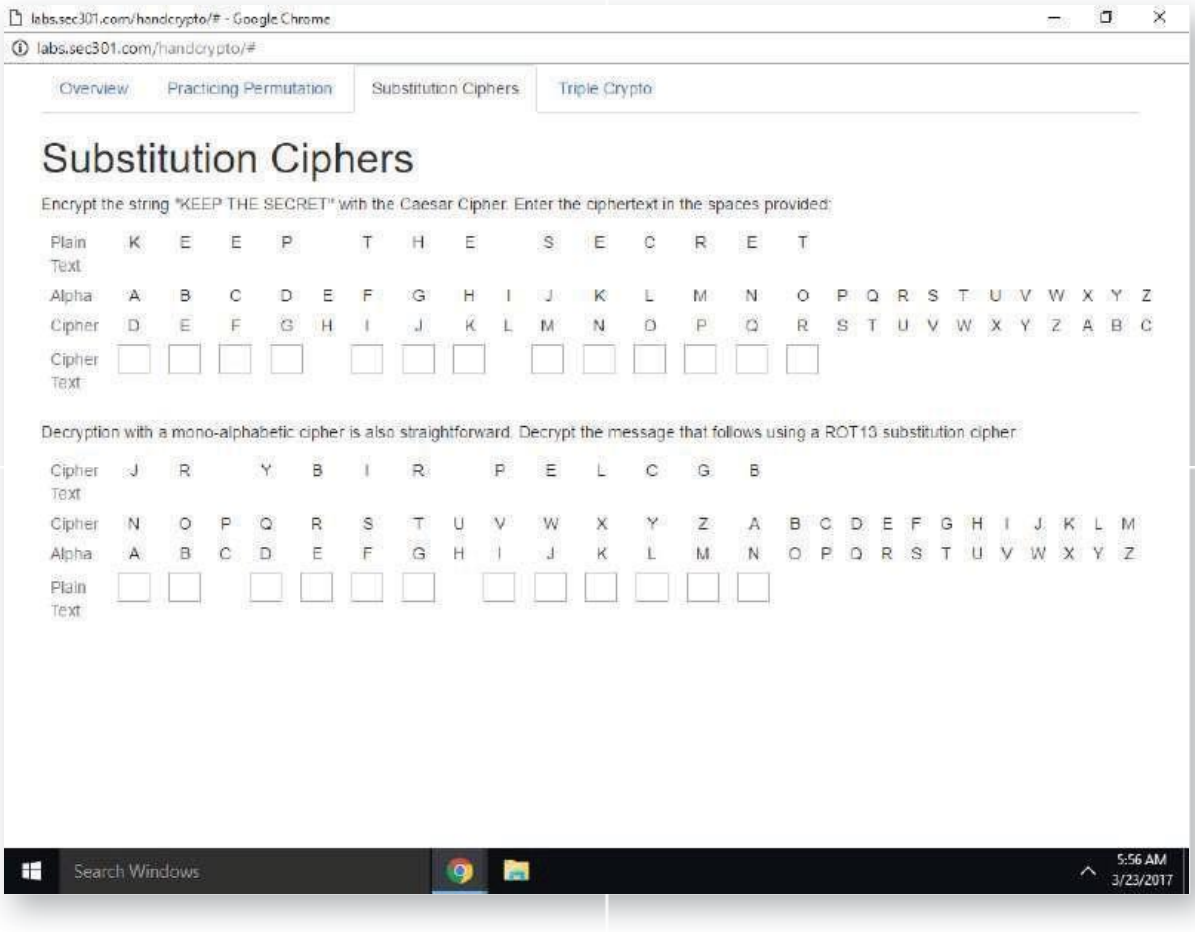
13. Click on the Substitution Ciphers Tab

Next, click on the tab marked **Substitution Ciphers**.

14. Substitution Cipher Encryption Overview

Next, you will use a simple substitution cipher. Specifically, you will encrypt the message *KEEP THE SECRET* using the Caesar Cipher we discussed in class.

To review, the Caesar Cipher is a mono-alphabetic cipher. Meaning that you create a cipher alphabet to replace the letters of the plaintext alphabet. In the case of this Caesar Cipher, it uses rotational substitution, rotating the alphabet by three positions to create the cipher alphabet. To use the cipher, you just replace the clear text letters with the corresponding cipher letters.



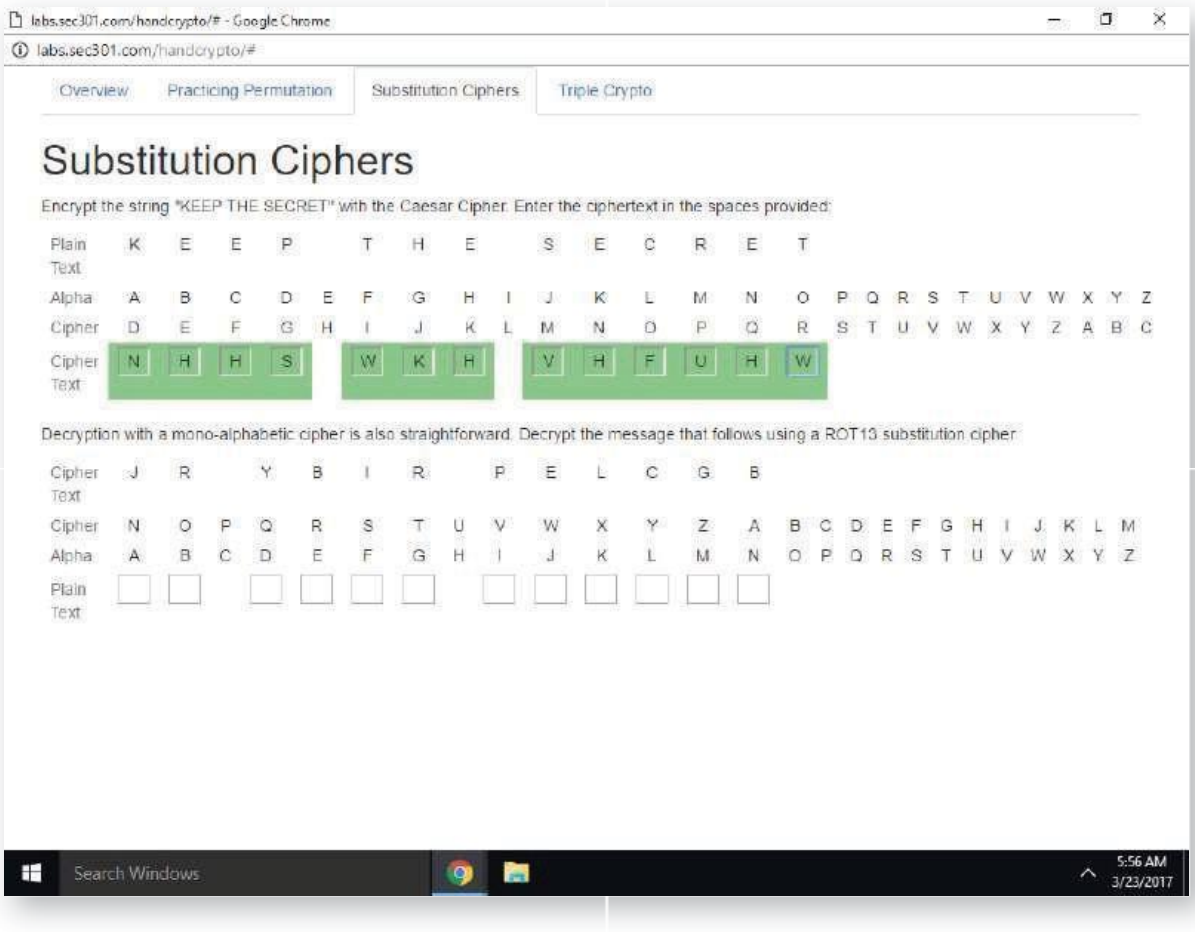
15. Encrypt a Substitution Cipher Message

Encrypt the string *KEEP THE SECRET* again, this time using the substitution cipher. Use the **Alpha** line to move to the plaintext character, then enter the corresponding character from the **Cipher** line in each input box.

16. Observing Patterns in the Ciphertext

Because we know the plaintext, it is easy to see a pattern in the resulting ciphertext (shown in the screenshot for this task). Specifically, note the recurrence of the letter H is replacing the letter E. The string *KEEP THE SECRET* is not long enough to allow for a statistical analysis attack. In this particular string, the letter E occurs over 38% of the time, as opposed to the 12.7% we would find with a more extended message. Still, you can see how the EE survives encryption, as does the TH digraph from the word *THE*, and so on. Given more ciphertext to work with (assuming the Caesar Cipher is used to create all the ciphertext), you could indeed begin to find the necessary patterns to break the code.

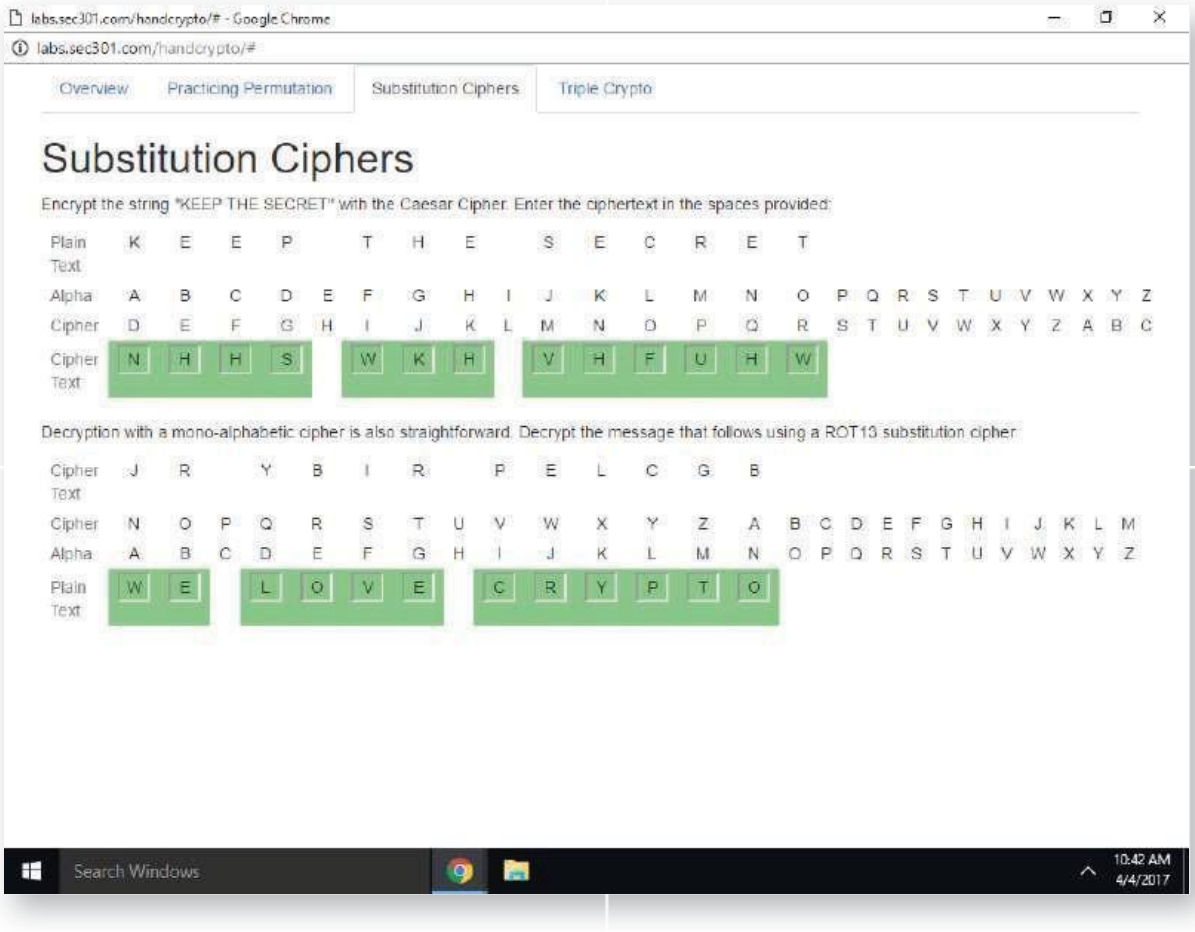
When you do encryption as previously shown with spaces between the words, this helps in statistical analysis. There is an entire area of study regarding the number of one-letter words, two-letter words, three-letter words, and so on and how often those appear statistically in a language. The researchers then take it to the next level and discover how often, for example, a two-letter word follows a three-letter word, and so on.



17. **Decrypt a Substitution Cipher Message**

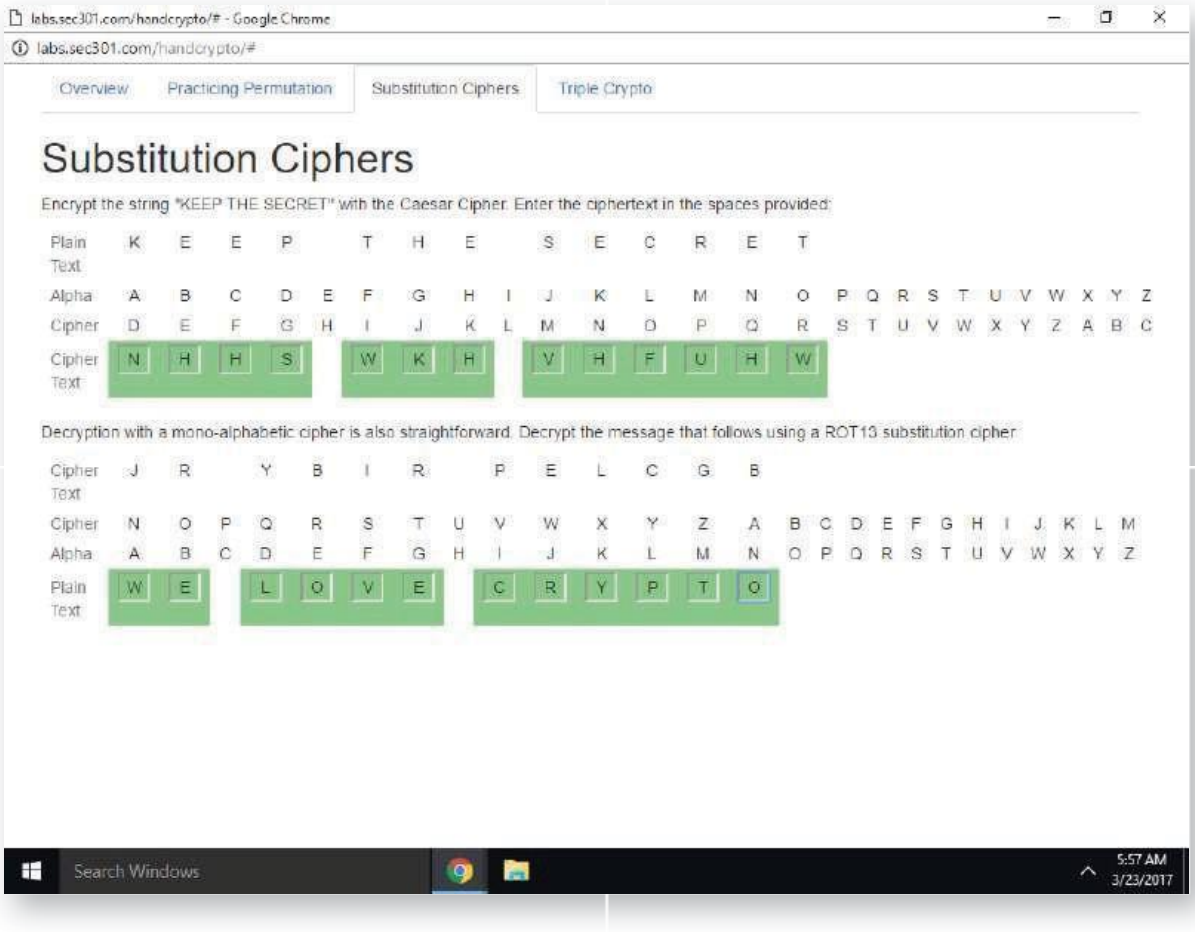
Decrypting with a mono-alphabetic cipher is also straightforward. Decrypt the string *JR YBIR PELCGB* using the substitution cipher with a ROT13 key.

Decryption of a substitution cipher is the same process as encryption. The only change is that the input is ciphertext, and the output is plaintext. Cryptosystems including AES and many others use this same encryption/decryption process, merely reversing the process as needed.



18. **Decrypted Substitution Cipher Answer**

    Click on the screenshot for this task to see the decrypted substitution cipher message.



19. **Click on the Triple Crypto Tab**

    Next, you will use the Vigenere Cipher to produce triple-encrypted content. Click on the **Triple Crypto** tab to continue.

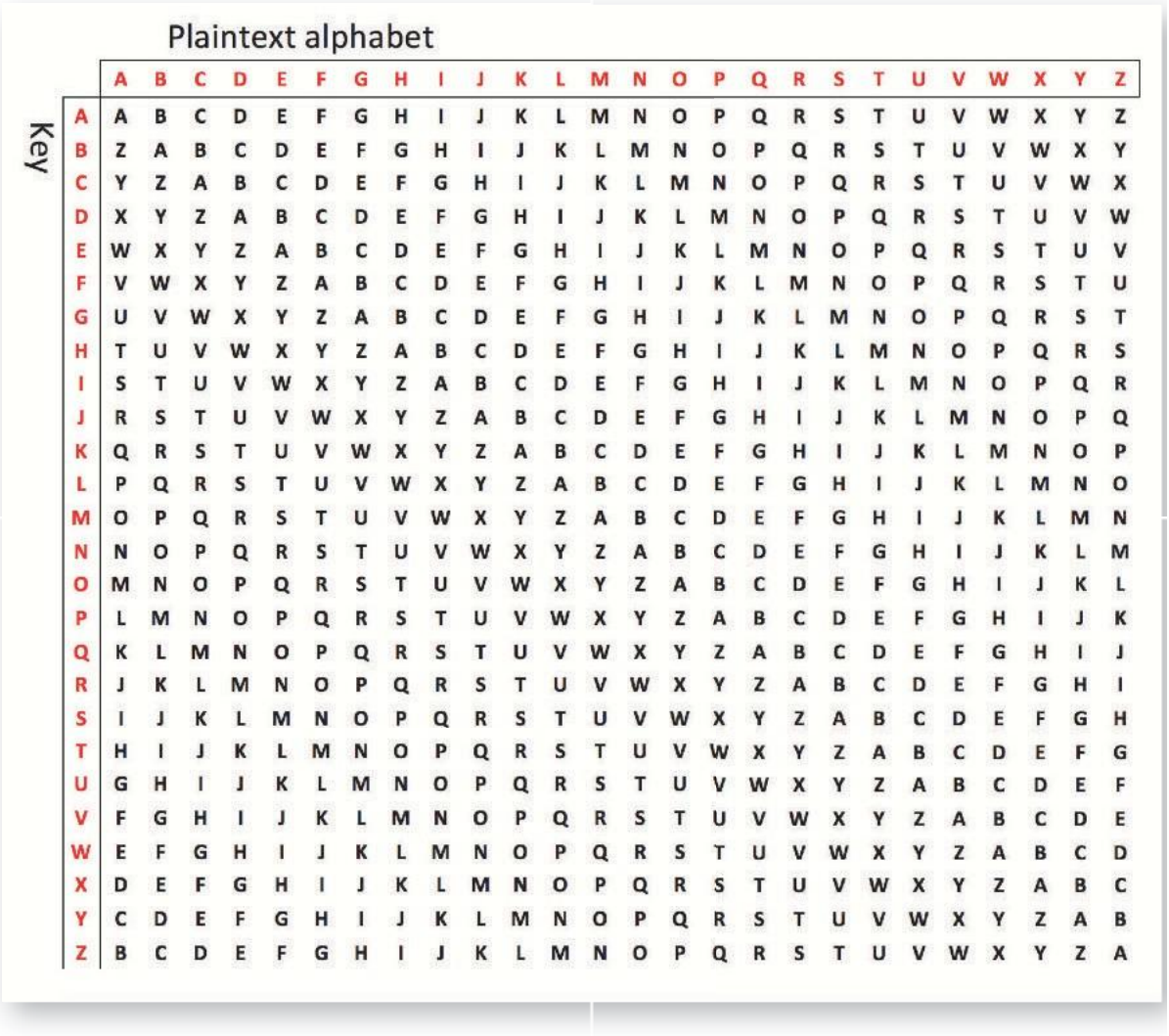
20. **Triple Crypto Overview**

In this lab, you will gain a complete understanding of how triple encryption, such as that used by Triple DES, works. Thankfully, instead of using the DES algorithm to illustrate triple encryption, you use the Vigenere Cipher.

To review, the Vigenere Cipher is a polyalphabetic cipher in that it replaces the plaintext alphabet with many cipher alphabets (in this case, 26 of them). It uses a table like the one included in the screenshot for this task. To use the table, find your plaintext letter across the top line to get the column and the letter of your key down the left side to get your row. Find the letter inside the table where that column and row meet. That is the ciphertext letter.

So, to encrypt CAB with a key of BUT, you would begin by finding the C column and the B row. These meet at the letter B inside the table, so the first ciphertext letter is B. The A column and the U row meet at the letter G (the second ciphertext letter). Finally, the B column and the T row meet at the letter I. So, the word CAB encrypted with the key of BUT becomes BGI in the ciphertext.

To decrypt, you need the key (BUT) and the ciphertext (BGI). Find the row of the first letter of the key (the B row). Move across that row until you hit the first letter of the ciphertext (also B in this case). After you find that column, the cleartext letter is at the top of the column; so in this case, that is the letter C.

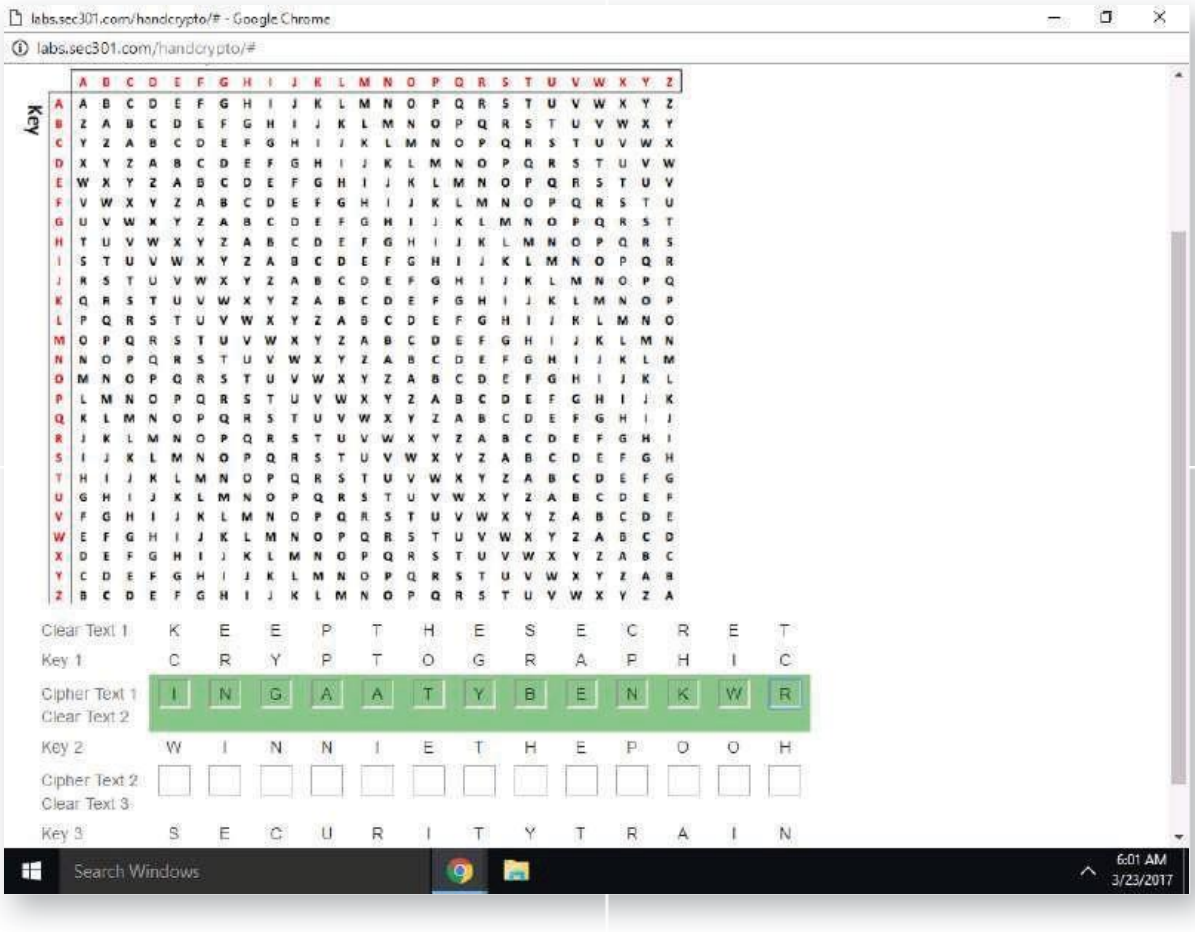


21. Encryption Pass 1

Encrypt the secret *KEEP THE SECRET* using the supplied Vigenere Cipher table with the key **CRYPTOGRAPHIC.**

22. Encryption Pass 1 Answer

Examine the screenshot in this task to see the answer for the first encryption pass.

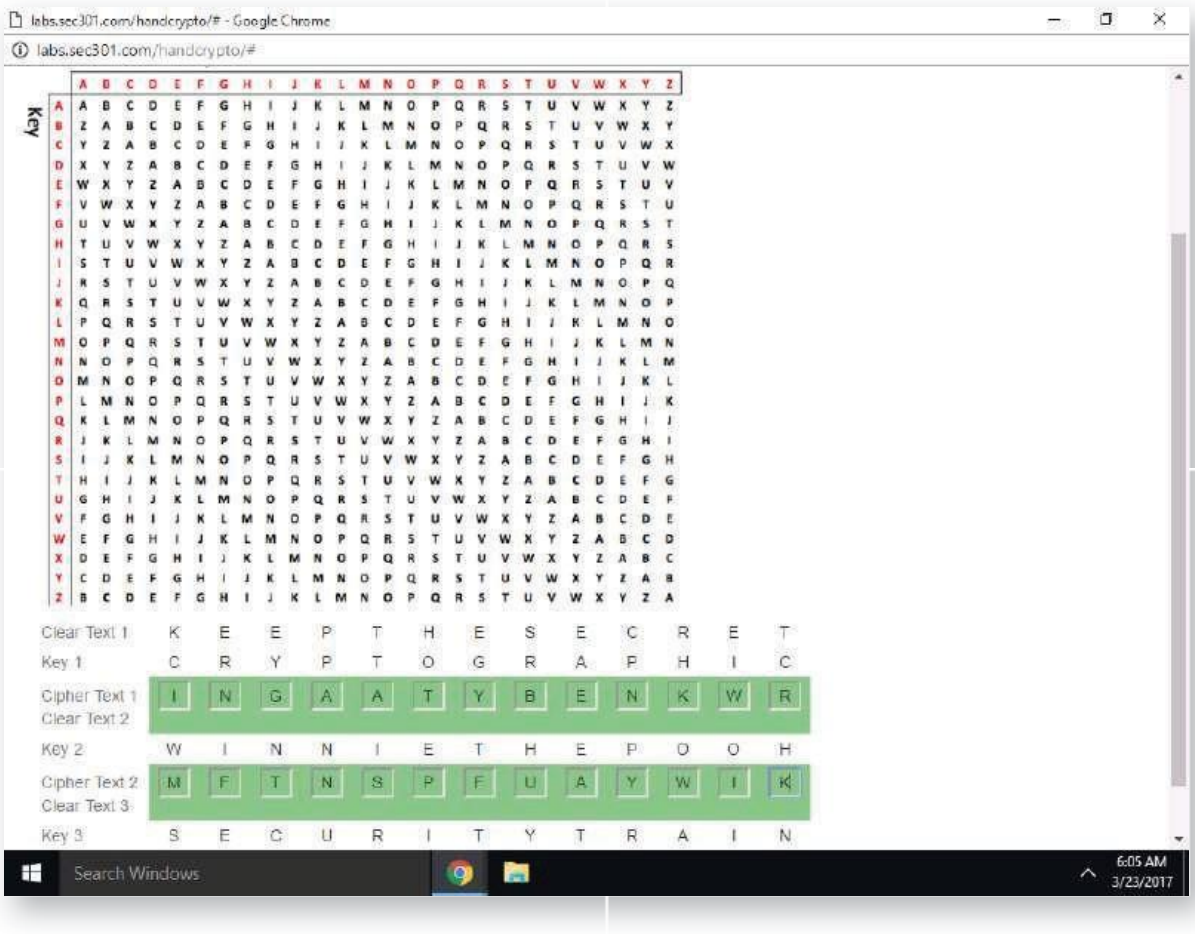


23. Encryption Pass 2

Next, encrypt the product of the first encryption pass *INGAATYBENKWR* with the key **WINNIE THE POOH**.

24. Encryption Pass 2 Answer

Examine the screenshot in this task to see the answer for the second encryption pass.

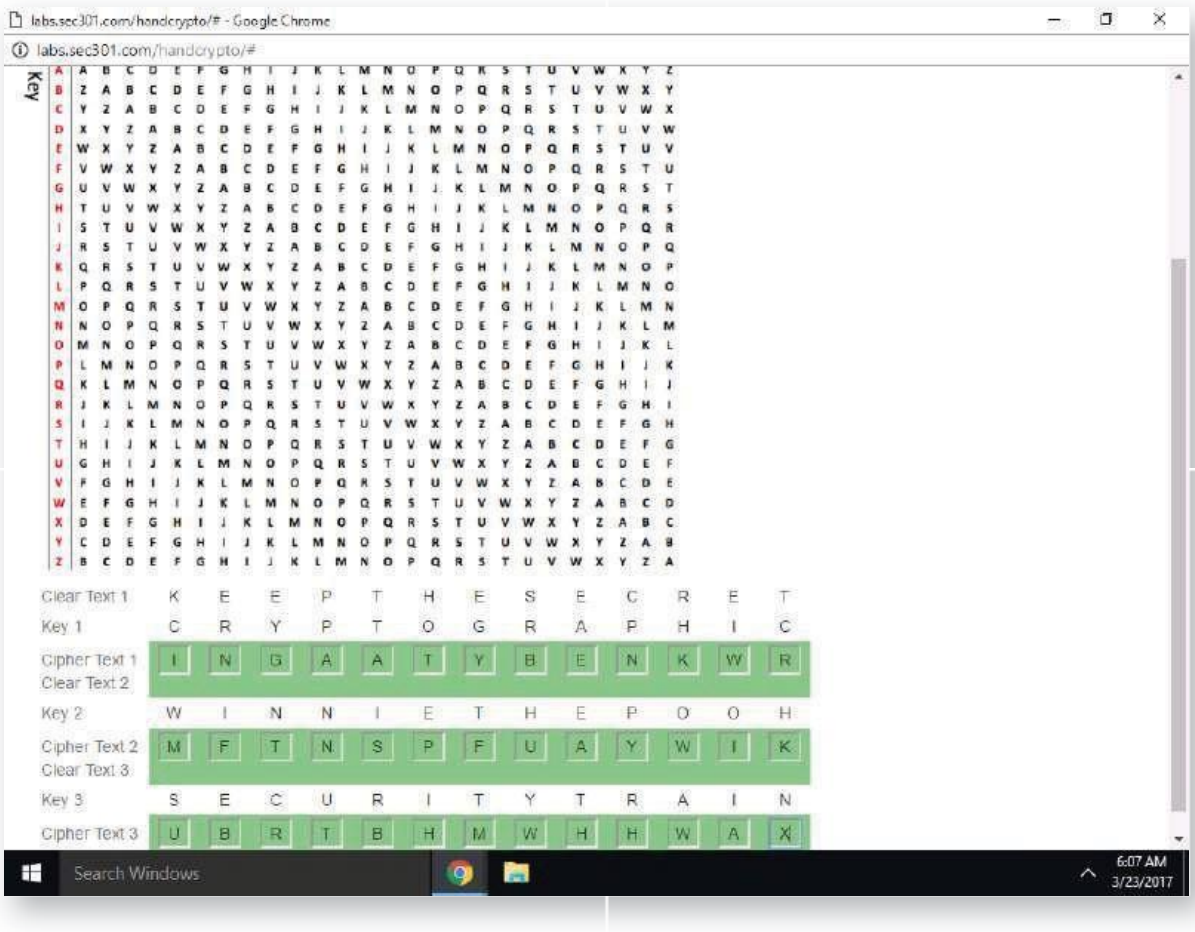


25. Encryption Pass 3

Next, encrypt the product of the second encryption pass *MFTNSPFUAYWIK* with the key **SECURITY TRAIN**.

26. Encryption Pass 3 Answer

Examine the screenshot in this task to see the answer for the third (and final) encryption pass.



27. **Triple Encryption Analysis**

    As you can see, the resulting ciphertext bears little resemblance to the original cleartext. With the first iteration of encryption, there was no obvious pattern, but with work, such a pattern may have emerged. (There is an advanced statistical analysis attack for doing exactly that.) With the second iteration, any such pattern would become more difficult to see, and by the third iteration, any pattern would become seriously hard to discover.

    As stated elsewhere in the course, Triple-DES does not use the Vigenere Cipher as any of its functions. That is not what we show here. This lab is merely illustrating how triple encryption works.

In this exercise, you worked through several challenges to understand the foundational elements of cryptography. Although the cryptography presented here was simple, the ability to recognize and apply the techniques of permutation ciphers, substitution ciphers, and polyalphabetic ciphers will give you insight into how cryptographic functions work. Applying the Vigenere Cipher, you were able to accomplish triple encryption, similar to what is used in modern cryptographic systems as well.

The challenges are more illustrative than they might appear. In addition to giving you the

opportunity to conduct encryption and decryption operations, you also observed real-world cryptographic concepts such as entropy and frequency analysis of encrypted data, the use of same procedure encryption and decryption operations, and the application of triple-crypto techniques using three distinct keys. Although we applied them in a simple manner here, each of these tasks has real-world application for modern cryptosystems.

Congratulations!

SEC301-3.2: Exercise - Visual Crypto

Objective

In this lab, you utilize a web application to enter various cleartext strings, generate a cryptographic key, and encrypt that cleartext into ciphertext. You also use that same cryptographic key to decrypt the ciphertext back into plaintext (note that plaintext and cleartext are synonymous terms).

Along the way, you gain a clearer understanding of what the instructor means during the lecture by "randomness in the ciphertext." Finally, you have an opportunity to observe what happens when something does not go right for encryption and decryption. For example, if there is a change to the cryptographic key between the time of encryption and decryption, what is the result? What happens when there is a modification to the ciphertext before the decryption operation?

This lab is one of those times where "doing" makes more sense than any written explanations. The lab, therefore, just guides you through the steps of what to click on next. However, the real story and where you need to pay the most attention is what is happening on your computer screen.

Scenario

Virtual Machines

1. SEC301-Win10
2. SEC301-LabServer

Visual Crypto

In this exercise, you use a custom web application to visualize and experiment with encryption and decryption routines. You supply the plaintext and the encryption key to produce ciphertext, then reverse the process to return to plaintext. You have the opportunity to visualize the randomness or entropy produced by the encryption routine, comparing it to the plaintext content. You also experiment with breaking the encryption and decryption process by manipulating the data elements.

1. **Open Chrome**

   Launch Chrome by double-clicking on the desktop icon.

2. **Open the Visual Crypto Lab Page**

   From the SEC301 lab server website, click on the tab near the top of the browser labeled **Visual Crypto Lab*.***

⚠ NOTE: There are two Crypto Tabs on the web page. This lab is on the **Visual Crypto tab.**

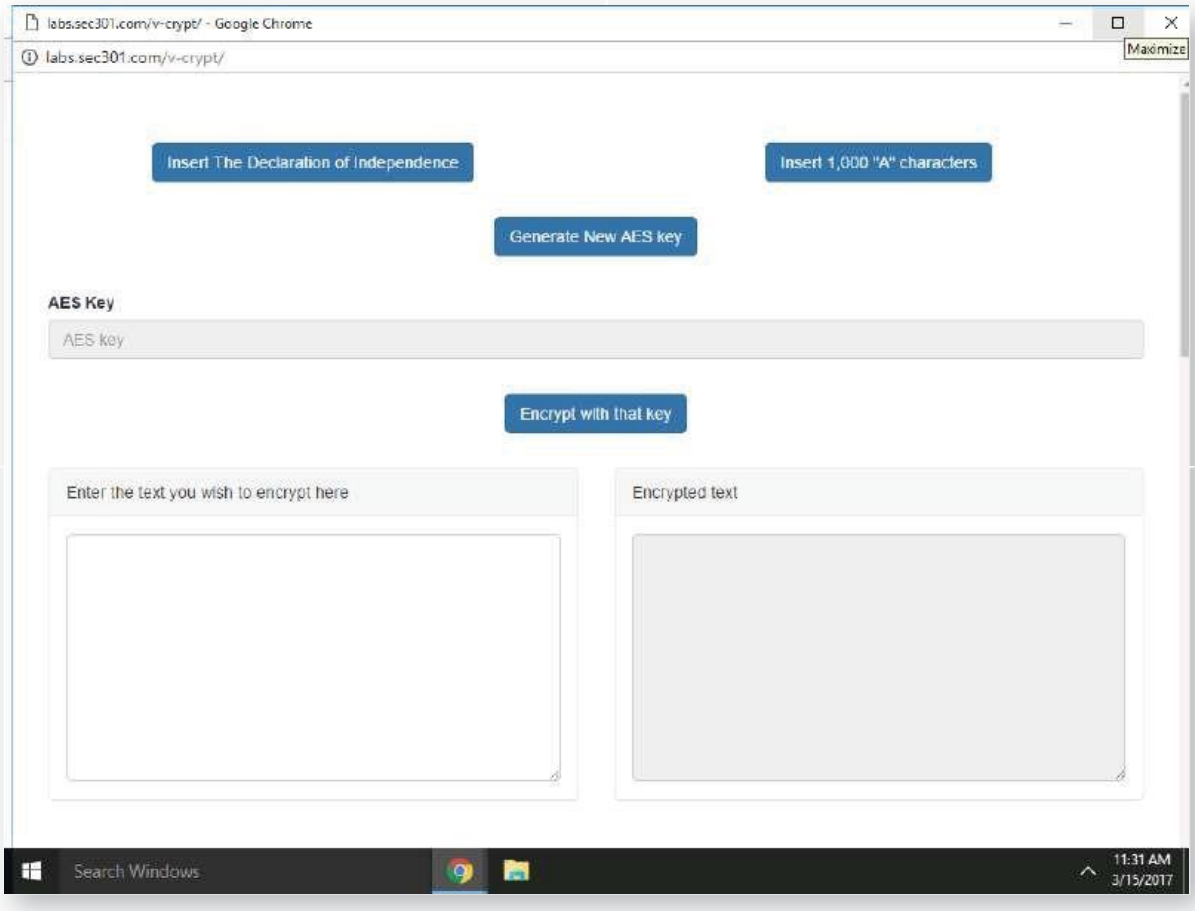
Also, remember to watch for the Tips and Screenshots offered throughout this lab!

3. **Start the Visual Crypto Application**

   Read the lab explanation on the Visual Crypto Lab page. When you finish, click the **Open Application** button to start the Visual Crypto application.

4. **Note the Application Buttons**

   Make particular note of the four blue buttons in the Visual Crypto application: Insert the Declaration of Independence, Insert 1,000 "A" characters, Generate New AES key, and Encrypt with that key. We use these buttons in the first part of this lab.



5. **Insert the Declaration of Independence**

Click the blue button labeled **Insert the Declaration of Independence**. When you click that button, you will see that document's text appear in the window labeled *Enter the text you wish to encrypt here*.

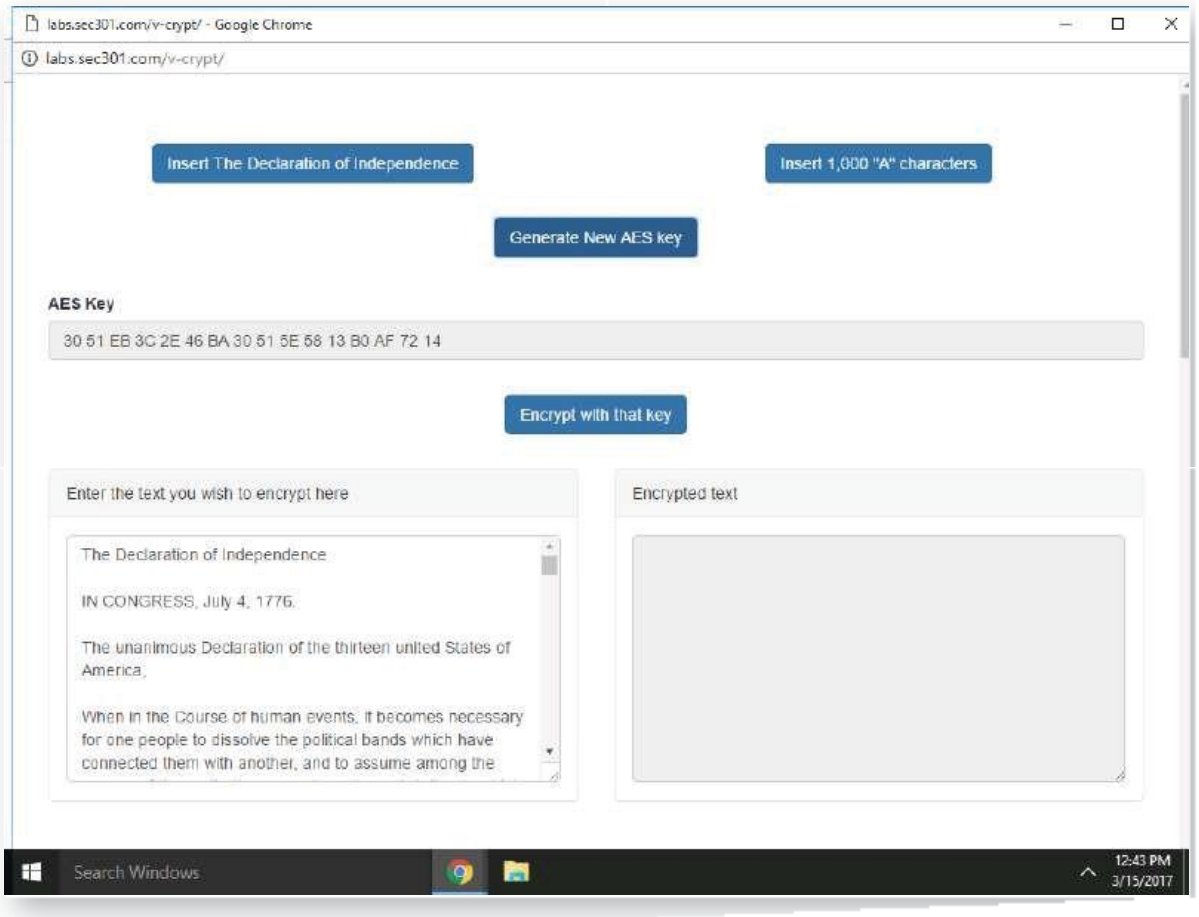
Note: The author of the course chose the Declaration of Independence for a simple reason. He needed a reasonably long document that he was sure does not fall under any copyright. The Declaration of Independence fits that requirement.

6. **Generate an AES Key**

   Next, click the **Generate New AES key** button. The application will generate a random 16-byte key populated in the AES key field.

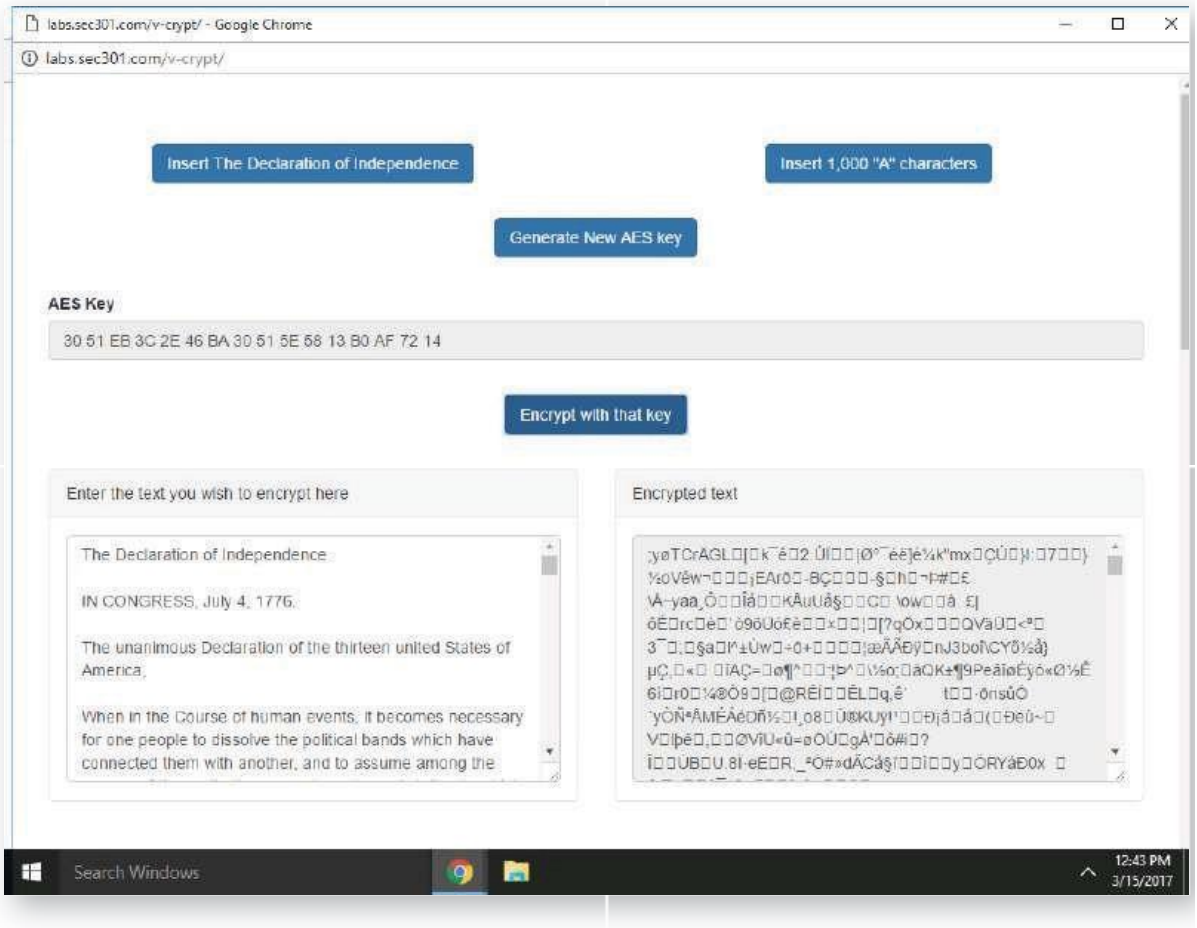
   Note: Your key will be different from the one shown here.

   Remember that two hex characters equal 1 byte or 8 bits. Therefore, the 16 bytes or 32 hex characters in the AES Key field are 128 bits.



7. **Encrypt the Plaintext**

Next, click the blue **Encrypt with that key** button. You see the encrypted text appear in the window labeled *Encrypted text*.



8. **Examine Unencrypted Byte Occurrence Frequency**

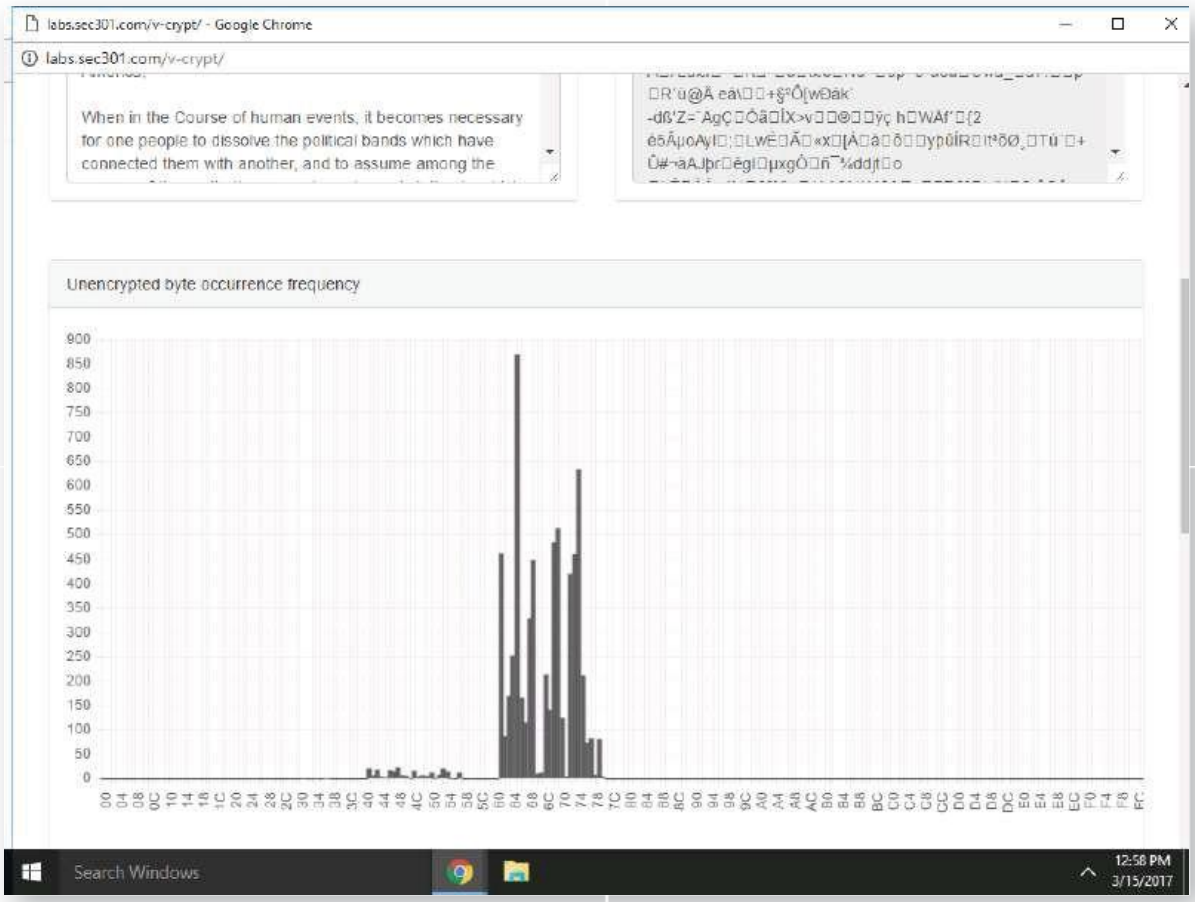
   Scroll below the plaintext and ciphertext windows to examine the chart labeled *Unencrypted byte occurrence frequency*. On the X-axis of this chart is the count of single-byte occurrences. On the Y-axis are the byte values themselves, from 0 to 255 (0x00 to 0xff). The vertical lines indicate the frequency of each byte value's occurrence.

   Examine the chart. What conclusions can you draw from the byte occurrence frequency? Click the Knowledge icon for our analysis of the data.

The character frequency is distributed in two primary groups, from 0x31 to 0x5a, and from 0x61 to 0x7a. These values correspond to the ASCII character set letters A-Z, and a-z.

The uppercase letters represent a much a smaller grouping of data compared to the lowercase letters. This is normal for a written letter (or declaration in this case) since lowercase letters appear much more frequently than uppercase letters.

The grouping of letters is typical for unencrypted ASCII-based documents.

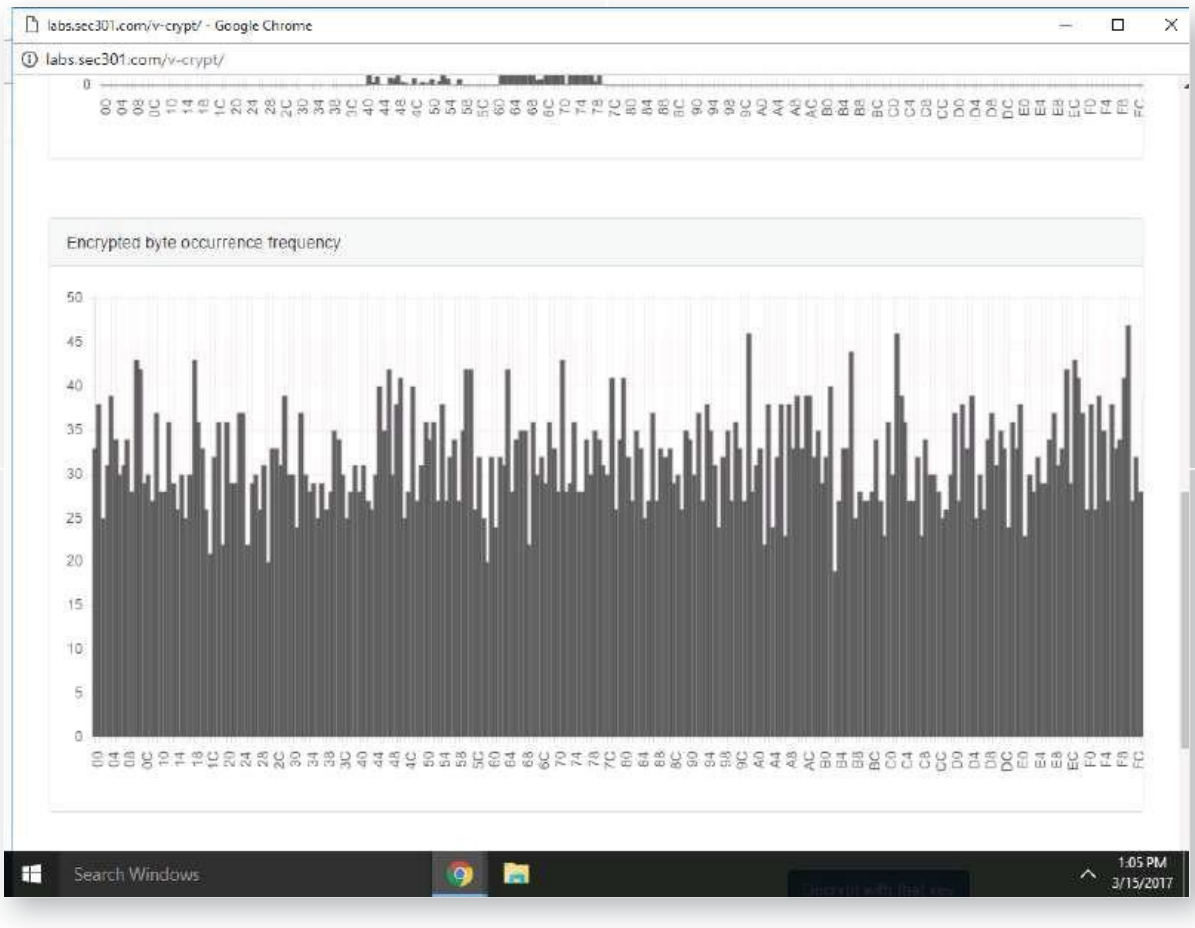


9. **Examine Encrypted Byte Occurrence Frequency**

   Scroll a little below the unencrypted byte occurrence frequency chart to examine the *encrypted* byte occurrence frequency chart. What conclusions can you draw from the encrypted byte histogram data? Click on the Knowledge button for our analysis of the data.

The character frequency in the *encrypted byte occurrence frequency* graph distributes much more evenly than the unencrypted content. Although there is substantial variation between bytes (some frequency is as high as 43 or as low as 20), the encrypted content frequency is much more even than the unencrypted content frequency.

The byte frequency achieved through encrypted content is indistinguishable from randomly generated content: no single byte is more or less likely to appear than any other byte. Compare this frequency to unencrypted content, where the byte 0x65 (ASCII lowercase e) is much more likely than byte 0x51 (ASCII uppercase Q).

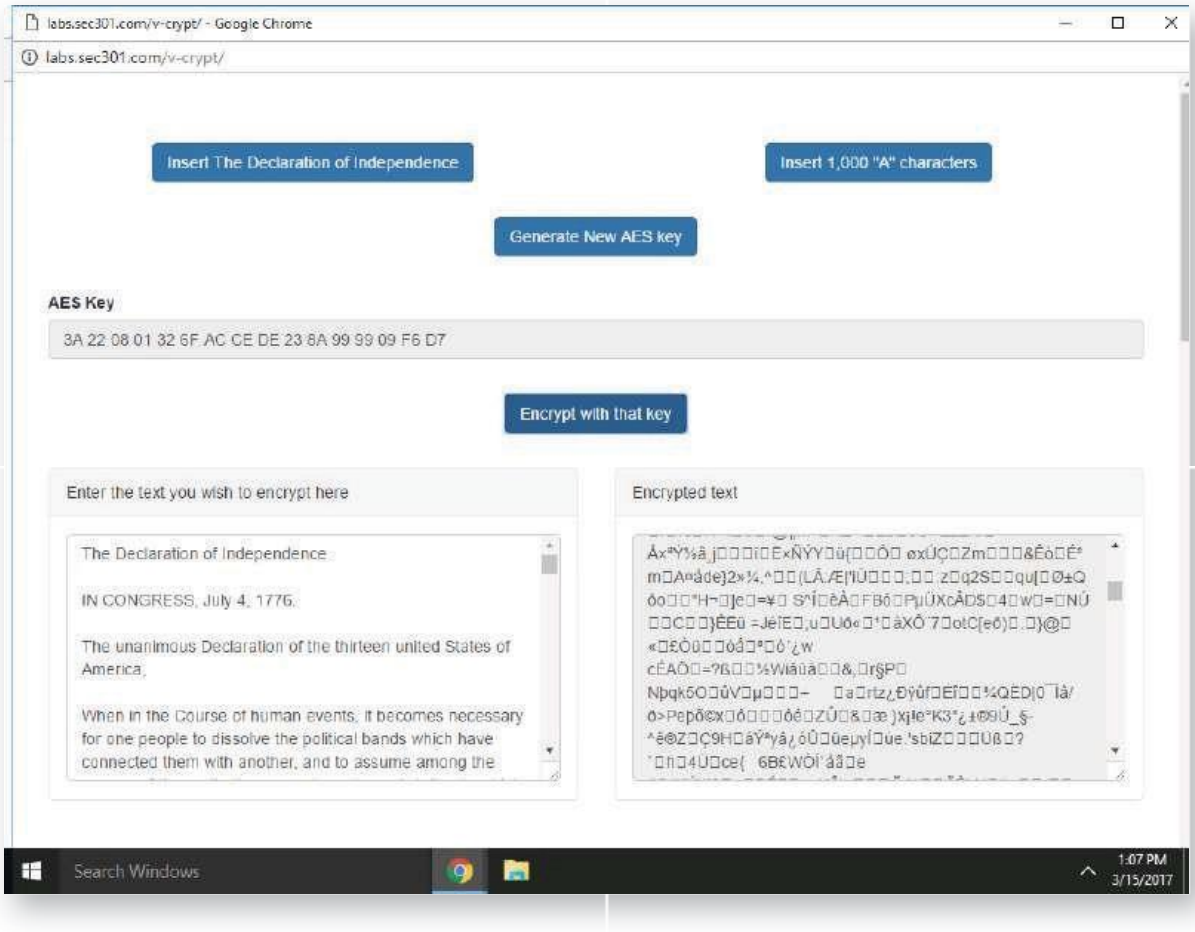


10. Generate a New AES Key

Scroll back up to the top of the application window. Click the **Generate New AES Key** button to generate a new key.

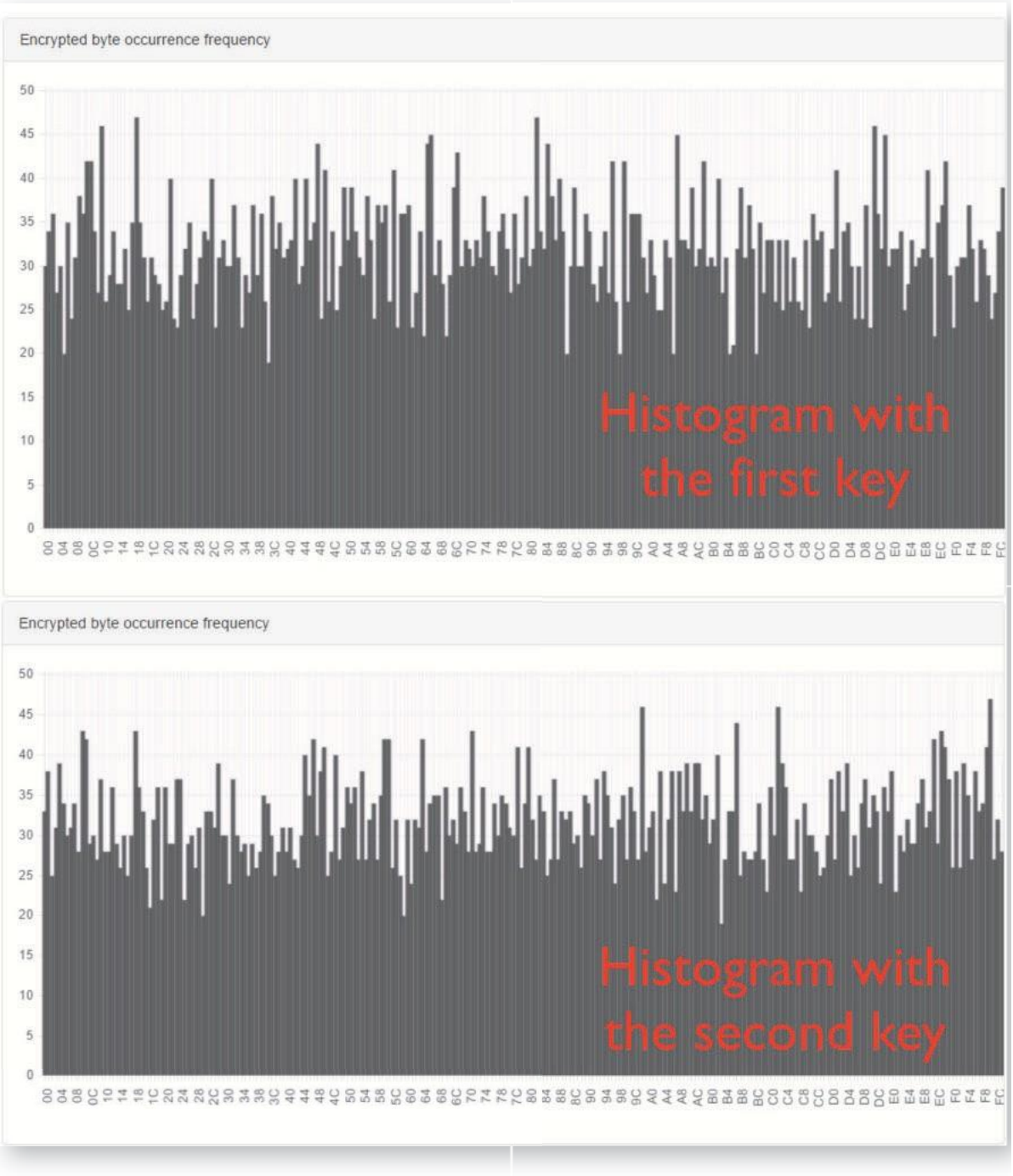
11. Encrypt with the New Key

Next, click the **Encrypt with that key** button. The encrypted content updates to reflect different content generated with the new key.



12. **Examine Byte Frequency**

    With the new key and encrypted data, examine the byte frequency again. The plaintext histogram stays the same, but the encrypted histogram should change somewhat. We show this with a side-by-side comparison in the screenshot (note: your histograms differ from ours since you are using a different key).

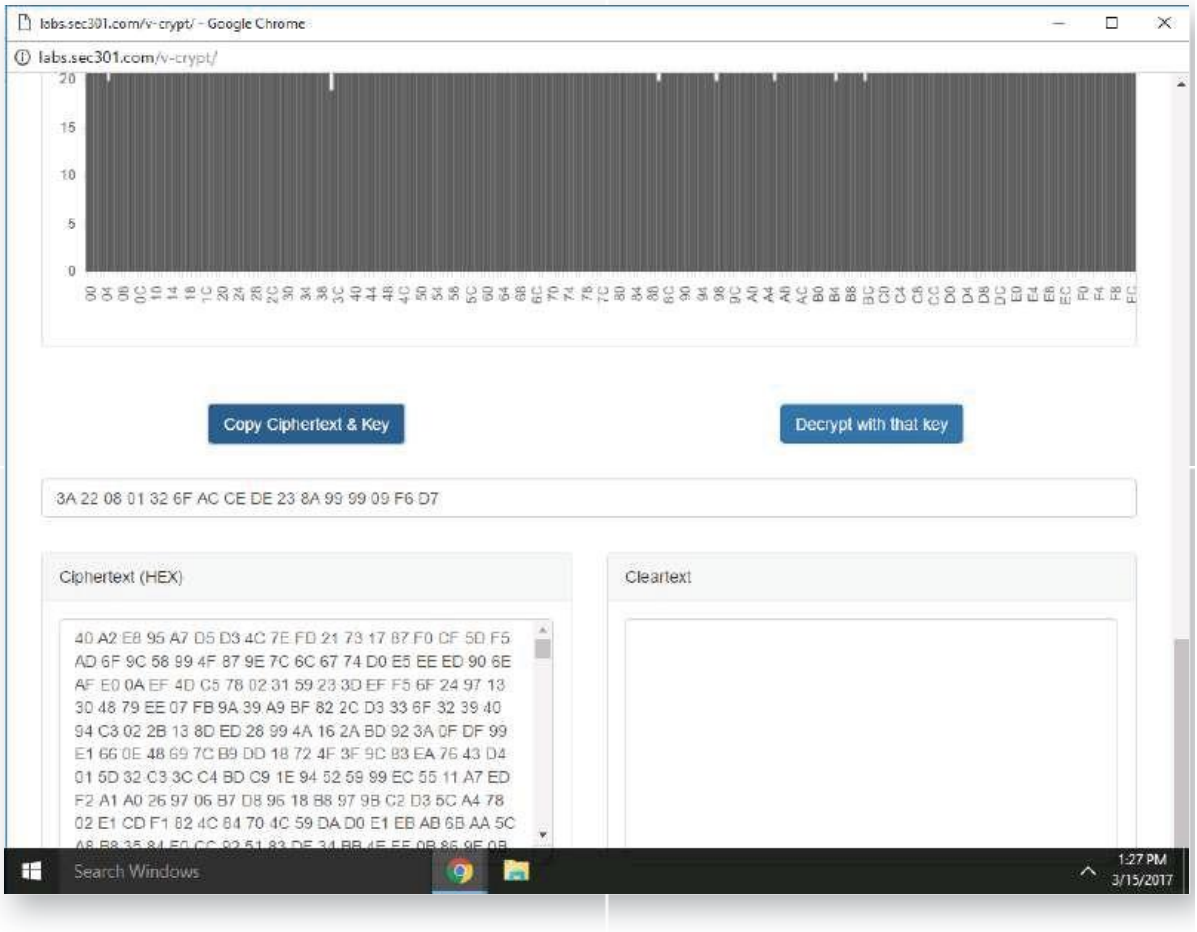


13. Scroll to the Bottom of the Application

Next, scroll to the bottom of the application. You will see two new buttons and the Ciphertext and Cleartext sections.

14. Populate Ciphertext and Key

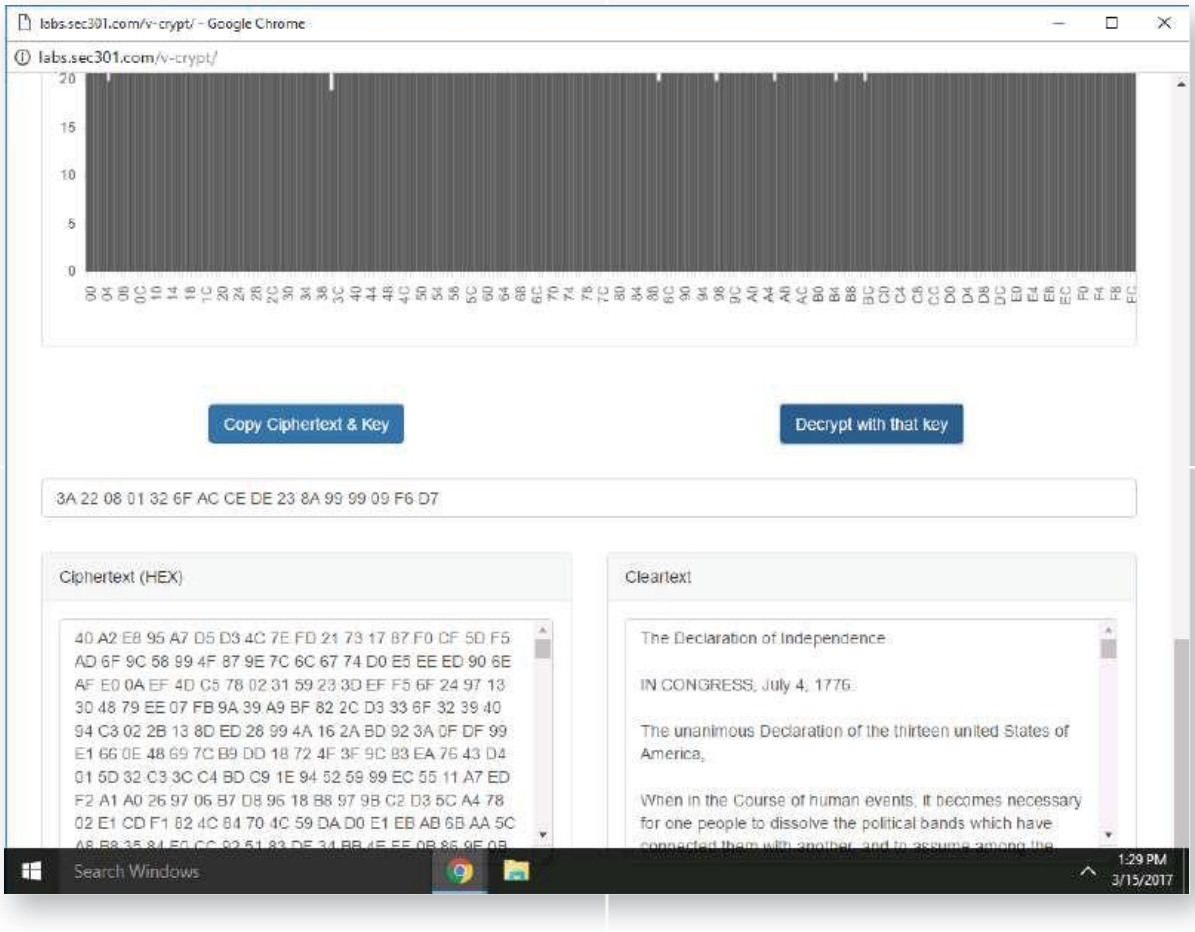
Click the **Copy Ciphertext & Key** button. The *AES key* and *Ciphertext (HEX)* fields populate with the previously generated content. Note that the ciphertext shows in hexadecimal format here, where it was displayed in ASCII format earlier.



15. **Decrypt Ciphertext**

    Click the **Decrypt with that key** button. The application uses the key to decrypt the content and populate the Cleartext block.

    With the ciphertext and the original AES key, you can decrypt the content, producing the original plaintext value again. The standard process of encryption and decryption operations is as follows; plaintext is encrypted with a key to produce ciphertext. The recipient uses the same key to decrypt the cyphertext and reproduce the plaintext.



16. **Edit the Key and Decrypt**

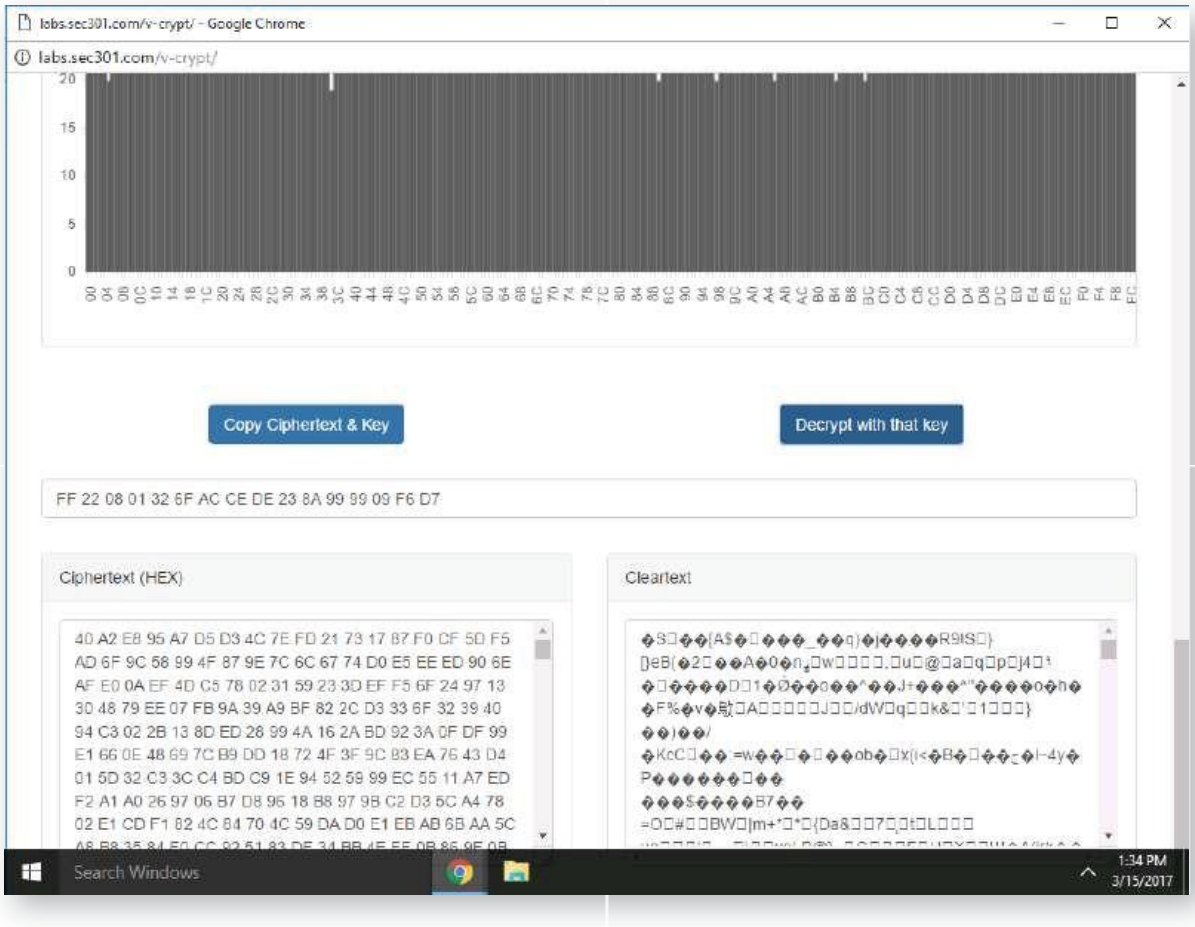
    Now that we have seen the product of encryption and decryption, let's experiment with manipulating the process. Change one byte of the AES key to another value (for example, change the first hex byte characters to FF). Click the **Decrypt with that key** button.

    What does the application produce when it tries to decrypt with the wrong key? Click on the Knowledge icon for our analysis.

AES is a symmetric encryption system, meaning that the key used to encrypt *must* be the same as the key used to decrypt. If the key is modified or damaged in any way, it "breaks" the decryption process.

When you try to decrypt the ciphertext with the wrong key, the application doesn't complain and attempts to decrypt the ciphertext like normal. However, the product of the decryption operation is not the valid ASCII we saw earlier. Instead, it produces unknown values commonly referred to as "garble."

The decryption operation worked as expected: when the wrong key is specified, we do not retrieve the original plaintext. Instead, the ciphertext decrypts to unknown values. Note that other applications may raise an error when using the wrong key.



17. **Populate Ciphertext and Key**

    Return to the original key value by clicking the **Copy Ciphertext & Key** button again.

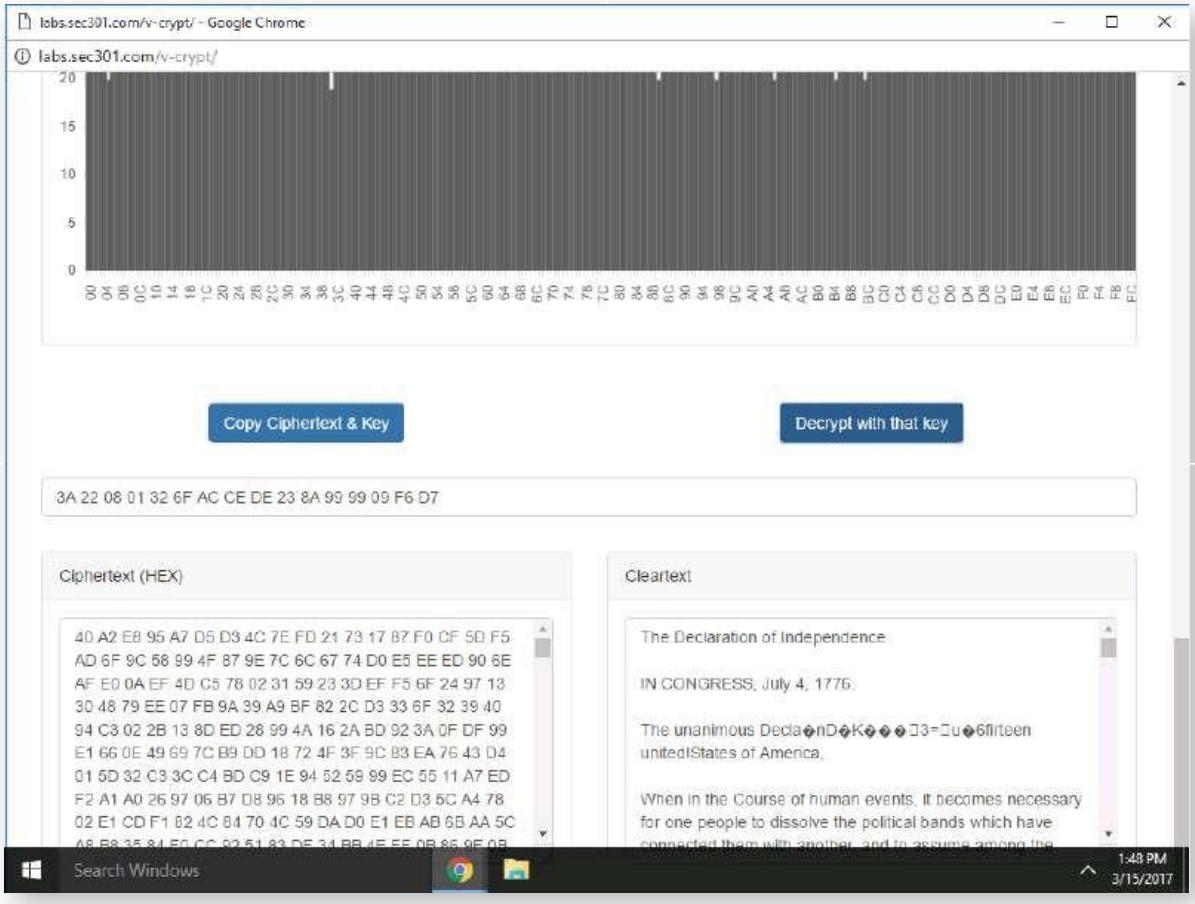
18. **Change Ciphertext and Decrypt**

Next, click on the Ciphertext (HEX) window someplace. Go down several lines and change a single hex character to a different hex character. (Note: You must use HEX characters here—0 through 9 and A through F). For example, in the screenshot, we changed the 4th byte of the 6th row from a 48 to 49. Then, click on the Decrypt with that key button.

What can you determine from the output that generates with this modification in place? Click on the Knowledge button for this task for our analysis.

AES is a block cipher, meaning that it encrypts (and decrypts) a block of text at a time. With 128-bit AES, which we use here, the size of the block is 128 bits, (with AES, key size and block size are always the same). Therefore, even though we changed only one HEX character, it results in an entire block of garbled text.

Although digital signatures give us absolute mathematical guarantees of message integrity, we do get a high level of assurance that a message is pristine and not tampered just by using encryption. Anytime you can decrypt ciphertext into 100% plaintext; it is an almost certainty that no tampering with the message occurred.



19. **Scroll to the Top of the Application**

    Let's do another test in this exercise. Scroll back up to the top of the application.

20. **Insert 1,000 "A" Characters**

    Next, replace the Declaration of Independence plaintext with 1,000 "A" characters. Click the **Insert 1000 "A" characters** button. Optionally, you can also generate a new AES key.

21. **Encrypt Plaintext**

    Next, click the **Encrypt with that key** button to encrypt the 1,000 "A" character plaintext.

22. **Examine Plaintext Byte Frequency**

    Scroll down to examine the plaintext byte frequency. Since we only have a single repeated character in the plaintext, there is a massive spike at byte value 0x41 (the ASCII code for uppercase A).

    In this example, the plaintext content has no variation. It is a single character repeated 1,000 times. The histogram is a stark contrast to the Declaration of Independence histogram.

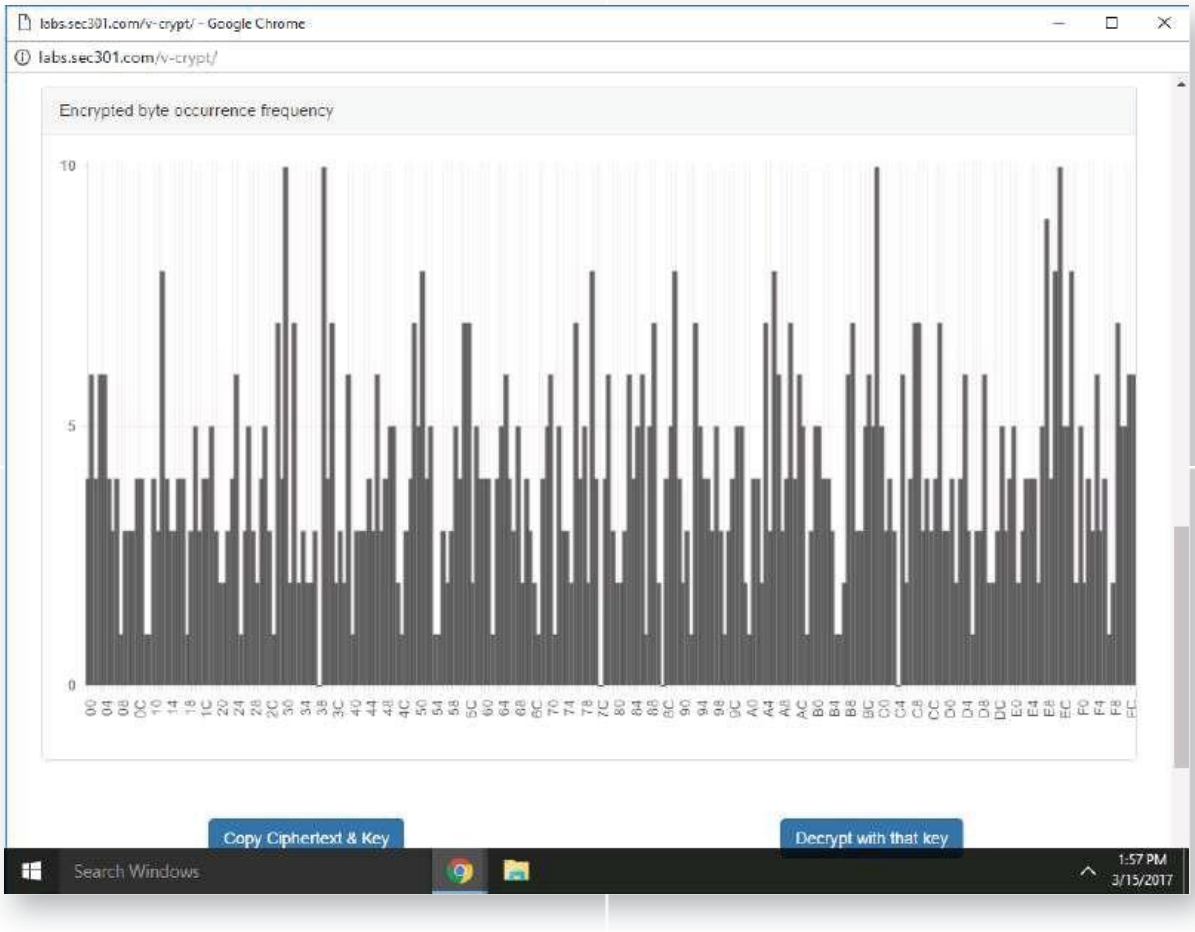
23. **Examine Ciphertext Byte Frequency**

    Scroll down a little more to examine the ciphertext byte frequency. What can you deduce from this output? Click on the Knowledge button for our analysis.

    The ciphertext content generated by the 1,000 "A" characters has a much more random entropy distribution. Although some values are more frequent than others, they are relatively close (compared to the entropy distribution of 1,000 "A" characters.)

    Regardless of the variation (or lack of variation) of the input value, the product of the encryption operation generates nonrepetitive content. High randomness is one of the strengths of a secure encryption algorithm: *no patterns are discernable in the ciphertext that corresponds to the input plaintext.*

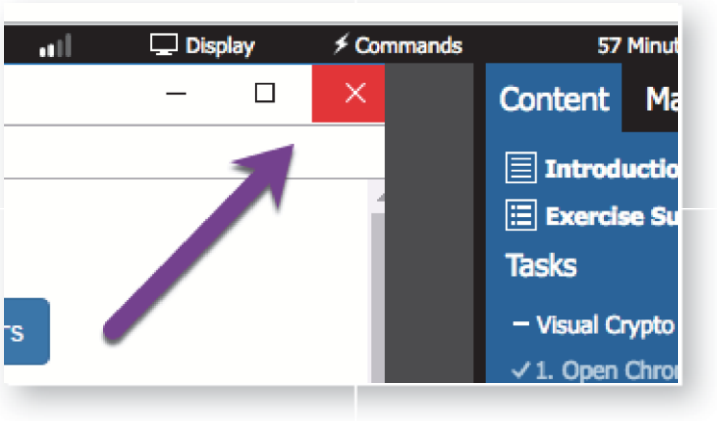


24. **Reload the Web Page**

    Before doing our last test for this lab, we need to clean up our work so far. The quickest and easiest way to accomplish this is to close the Visual Crypto App by clicking the **X** in the upper-right corner (see the screenshot).

    Once you close the Visual Crypto App, you will see the blue button to **Open Application**. Click that button to reopen the Visual Crypto App.

    When we close the Visual Crypto App window, all of our work to this point goes away. When we reopen the application, we have a clean slate to work with during our next section.



25. **Create a new Crypto Key**

    Next, please click on **Generate New AES key**.

> ⚠ Please DO NOT click on the buttons to insert the Declaration of Independence or the 1,000 A's.

26. **Insert a single character**

    In the window labeled "Enter the text you wish to encrypt here", enter any **single** character on the keyboard.

27. **Encrypt one Character**

    Click on the **Encrypt with that key** button. This will generate some Cyphertext.

28. **Examine the result**

    Examine the cyphertext that results from encrypting a single letter (a single byte). Feel free to also look at the windows for "Unencrypted byte occurrence frequency" and "Encrypted byte occurrence frequency".

    You encrypted a single character or byte - but you wound up with many characters (bytes) of cyphertext. Why?

    After thinking about it for a while, feel free to click on the Tip for this task.

AES is a block cipher. Since we are using a 128-bit key, we also have a 128-bit block size (in AES, the key and block size are always the same). When you encrypt with a block cipher, the plaintext size must be an exact multiple of the block size.

So you entered 8-bits. The AES algorithm padded that with an additional 120-bits to make the input a total of the 128-bit block size and encrypted the full 128-bit string.

Cryptography is a complex topic. If you have never been exposed to it before, or if you have been exposed to the topic but did not understand it well, seeing it occur with your own eyes often helps. This lab presents you with the opportunity to experiment and visualize the input and output values in several ways.

This lab should make clear the concepts from the lecture of "the key that encrypts must be used to decrypt," "if anything tampers with the ciphertext in route, it will not decrypt properly," and "a good algorithm generates ciphertext that is highly random."

Congratulations on completing this exercise!

SEC301-4.1: Exercise - Wireless Access Point Configuration

Objective

Securely configuring a wireless access point requires insight into many different technologies integrated into these small devices. In this exercise, you'll build familiarity with the common configuration options on wireless access points (APs) through the use of our custom wireless AP product: Linksnips.

Scenario

Virtual Machines

1. SEC301-Win10
2. SEC301-LabServer

Wireless Access Point Configuration

In this exercise, you will configure the administrative settings for a wireless access point (AP) product, Linksnips. The Linksnips product resembles that of many other commercial wireless AP products with similar configuration options available through a web-based interface.

After logging in to the Linksnips administrative console, you configure the required configuration settings to secure the wireless AP, then validate your configuration options.

1. **Open Chrome**

   Launch Chrome by double-clicking on the desktop icon.

2. **Open the WiFi Security Lab Page**

   From the SEC301 lab server website, click on the tab near the top of the browser labeled **WiFi Security Lab.**

3. **Start the WiFi Security Application**

   Read the lab explanation on the WiFi Security Lab page. When you finish, click the **Open Application** button to start the WiFi Security application.

4. **Maximize the Linksnips Window**

   Click the maximize button to fill your screen with the Chrome browser, as shown in the screenshot for this task.

5. **Login to the Linksnips Router Configuration Page**

   From the Linksnips router configuration page, login with the default password ***admin***, then click the **Login** button.

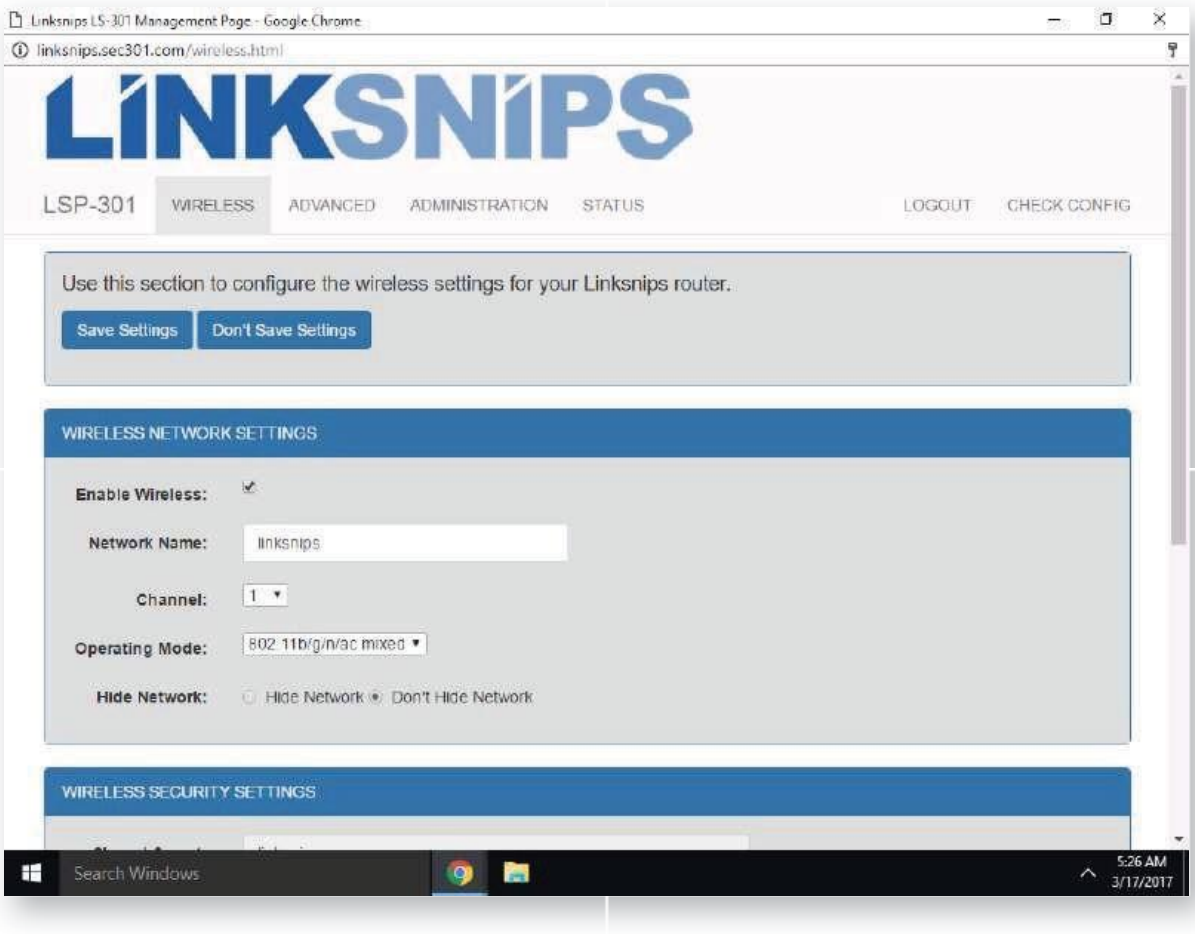
   ⚠ Be sure to take note of the many extra Tips in this lab. They provide significant extra information about why you are doing some of these steps.

6. **Examine Linksnips Configuration Interface**

   Here you configure the Linksnips LSP-301, a wireless access point product. The LSP-301 administrative interface uses three configuration tabs at the top of the screen (Wireless, Advanced, and Administration). After changing a setting, click the **Save Settings** button to apply and save the change.

   The Linksnips LSP-301 also offers a *Check Config* option, which evaluates your configuration settings to determine if they match the recommended security best practices. You can click the Check Config option at any time to check your settings.

   (Note: While many of these settings apply to your home router, it will almost certainly NOT have a Check Config button.)



7. **Set Network Name**

   Change the default network name from *linksnips* to one of your choosing. The network name or SSID can be no longer than 32 characters.

   > Changing the wireless SSID to a non-default, uncommon name will help defeat shared secret guessing attacks.

8. **Set Wireless Security Type**

   The default wireless security for the LSP-301 is disabled (an open WiFi network). Change the security setting to the strongest available type, **WPA2 Personal**.

   > Using WPA2 Personal wireless network security utilizes the strongest security available in consumer-level wireless router products.

9. **Set Shared Secret**

   Change the default shared secret value from *linksnips* to one of your choosing. The shared secret must be between 8 and 63 characters in length.

Changing the wireless shared secret to a complex value helps to prevent an attacker from connecting to your wireless network. See the earlier Haystack Passphrase lab for guidance on the type of shared secret you want here.

10. **Disable WiFi Protected Setup**

    Turn off WiFi Protected Setup (WPS) by clicking the **Disabled** radio button next to the *WPS Status* label.

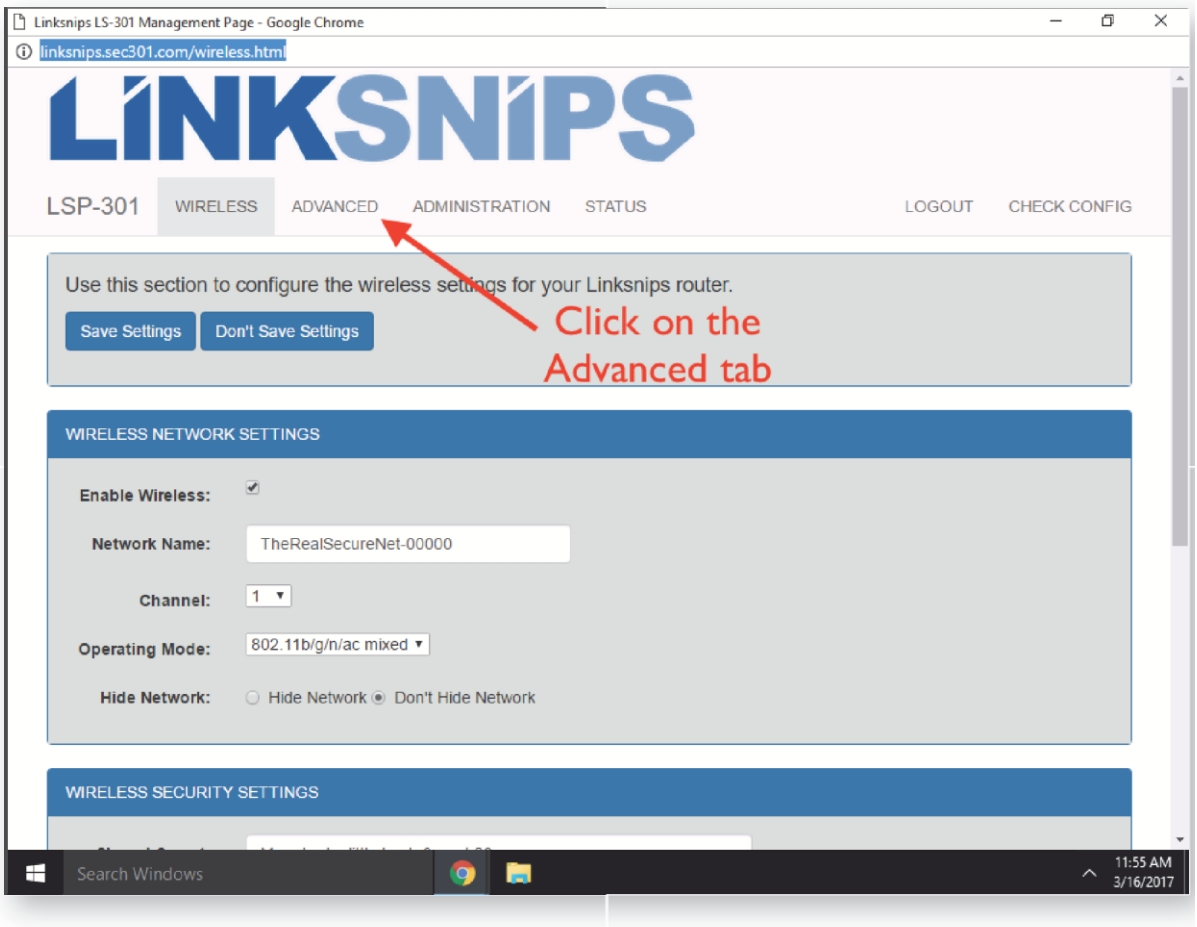
Disabling WiFi Protected Setup in favor of a pre-shared key authentication is the strongest security mechanism available in most consumer wireless router products.

11. **Save the Wireless Settings**

    Scroll back up to the top of the Linksnips wireless settings section and click **Save Settings**. When prompted, click **OK** to acknowledge the message.

12. **Click on the Linksnips Advanced Tab**

    Next, click on the Linksnips **Advanced** configuration tab near the top of the Linksnips application window.



13. **Turn on Firewall Protection**

    In the *Firewall Settings* group, turn on the network firewall by clicking the **On** radio button next to the *Firewall Protection* option.

    Using a network firewall is an essential service to defend against internet attacks.

14. **Turn on ICMP Filtering**

    Next, turn on ICMP filtering by clicking the **On** radio button next to the *Filter ICMP* option.

    Filtering ICMP services prevent an attacker on the internet from interrogating internal hosts through tools such as the ping utility. Remember from our Network discussion that ICMP is the Internet Control Message Protocol and carries the error messages of IP. It also supports ping, which an attacker could use to map a network.

15. **Examine Filter Web Options**

Some firewall devices can filter web application traffic, blocking specific content such as access to proxy servers, Java applets, Flash programs, and web cookies. Blocking these elements can break desired functionality, and should only be applied when you fully understand the implications of these changes in advance.

For now, leave these options unchecked.

16. **Examine DMZ Service Option**

    The default Linksnips configuration is to disable DMZ services. Turning on DMZ services allows an internal system to be publicly accessible from outside the firewall, and should only be used when required for publicly-accessible systems.

    Disabling DMZ services prevent an attacker on the internet from reaching hosts behind the network firewall directly.

17. **Turn on Network Address Translation**

    A critical capability of wireless network routers is the ability to perform Network Address Translation/Port Address Translation (NAT/PAT). Turn on the NAT/PAT feature in the **Network Address Translation** group.

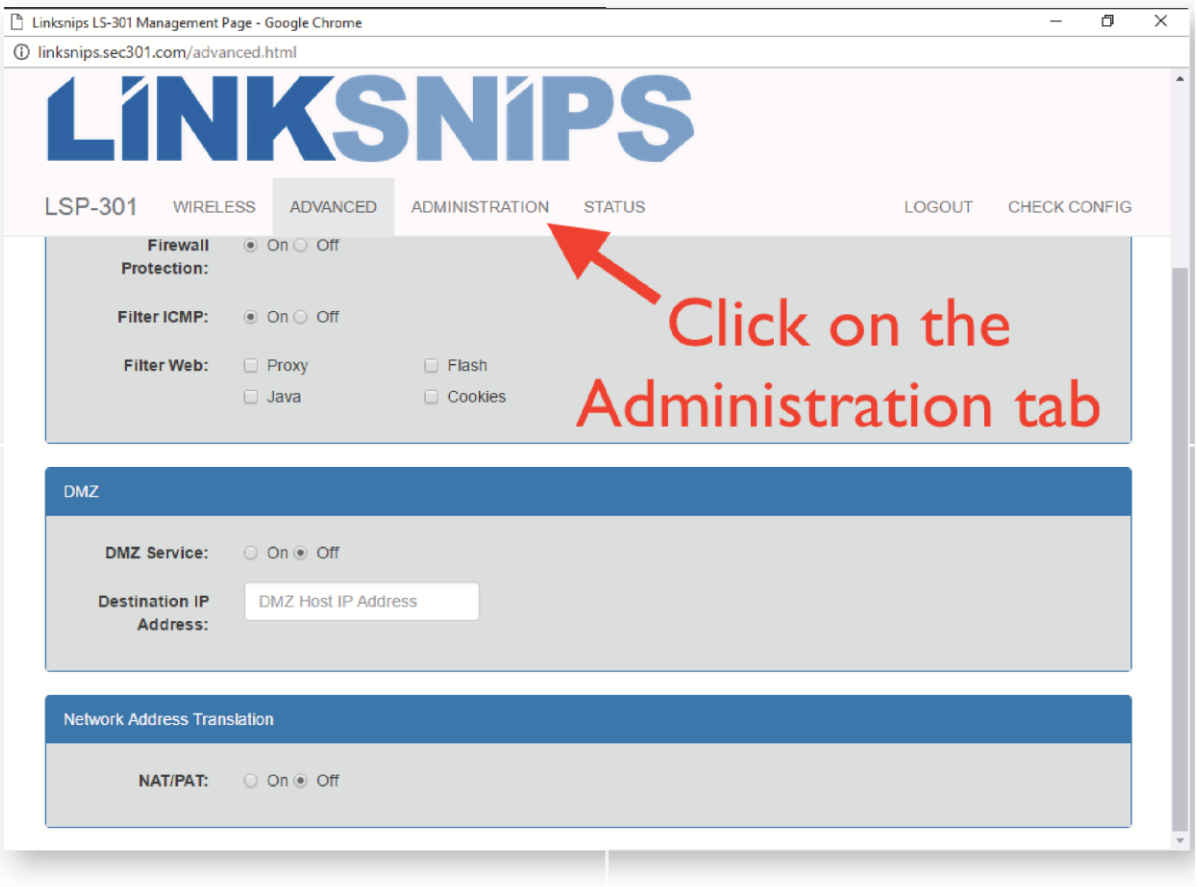
    Using network address translation (NAT) protects multiple hosts behind a single IP address.

18. **Save the Advanced Settings**

    Scroll back up to the top of the Linksnips wireless settings section and click **Save Settings**. When prompted, click **OK** to acknowledge the message.

19. **Click on the Linksnips Administration Tab**

    Next, click on the Linksnips *Administration* tab near the top of the Linksnips application window.



20. **Change the Router Password**

    The default administration password for the Linksnips router is *admin*. Choose and enter a *strong* password in the **Password Management** group, then confirm the password by entering it again.

    Changing the default administrative password stops the most common attacks against wireless routers.

21. **Disable HTTP Web Configuration Access**

    Where possible, administrative access to the wireless router should occur over HTTPS. Some products allow you to enforce this requirement by disabling HTTP access.

    In the **Web Access** group, disable HTTP configuration access, keeping the HTTPS option turned on.

    Disabling HTTP management access prevents an attacker who is monitoring your wireless network from capturing your administrative password.

22. **Disable Web Configuration Access over WiFi**

Some wireless router products can block administrative access over WiFi interfaces, requiring that the administrator configure the router over a wired connection. This feature is excellent for network appliances we do not often configure, requiring that any administrative user also have physical access to the device (or a connected wired network port).

Disable web configuration access over WiFi.

Turning off web configuration access over WiFi stops an attacker from accessing your wireless router unless they can plug into an available port.

23. **Disable Remote Management Access**

Remote management access on wireless router products allows a user on the internet to connect and try to login to the device to manage settings. Disabling remote management access stops an attacker on the internet from accessing the router login page.

Scroll to the **Remote Access** section, then disable remote management.

24. **Disable Remote Web Configuration Access**

Sometimes, web interfaces for networking products aren't always as clear as we'd like, and the Remote Web Configuration Access settings are one example. Although you've disabled Remote Management, the check boxes for Remote Web Configuration Access (HTTP and HTTPS) are still available. It is not clear if the prior option disables HTTP and HTTPS access, or the access persists because of these checked boxes.

When in doubt, disable all the pertinent options. Where possible, you should also consider testing the effect of your changes (possibly working with a second person outside of the network).

Here, disable both HTTP and HTTPS Remote Web Configuration Access to ensure the services are disabled.

Disable remote management HTTP and HTTPS access to prevent an attacker from attempting to login and access the management settings on your wireless router from the internet.

25. **Disable Remote Upgrade**

Like remote access, remote upgrade capability allows an attacker to remotely replace the firmware on a wireless router with an updated (or malicious) version. Disable the remote upgrade capability.

Disable remote upgrade capabilities to stop an attacker from replacing the device firmware over the internet.

26. Disable UPnP Access

In addition to HTTP or HTTPS-based web management applications, many network devices allow administrators to configure settings over the Universal Plug-and-Play (UPnP) protocol. If it is not expressly required, disable UPnP access to restrict management access to the web console.

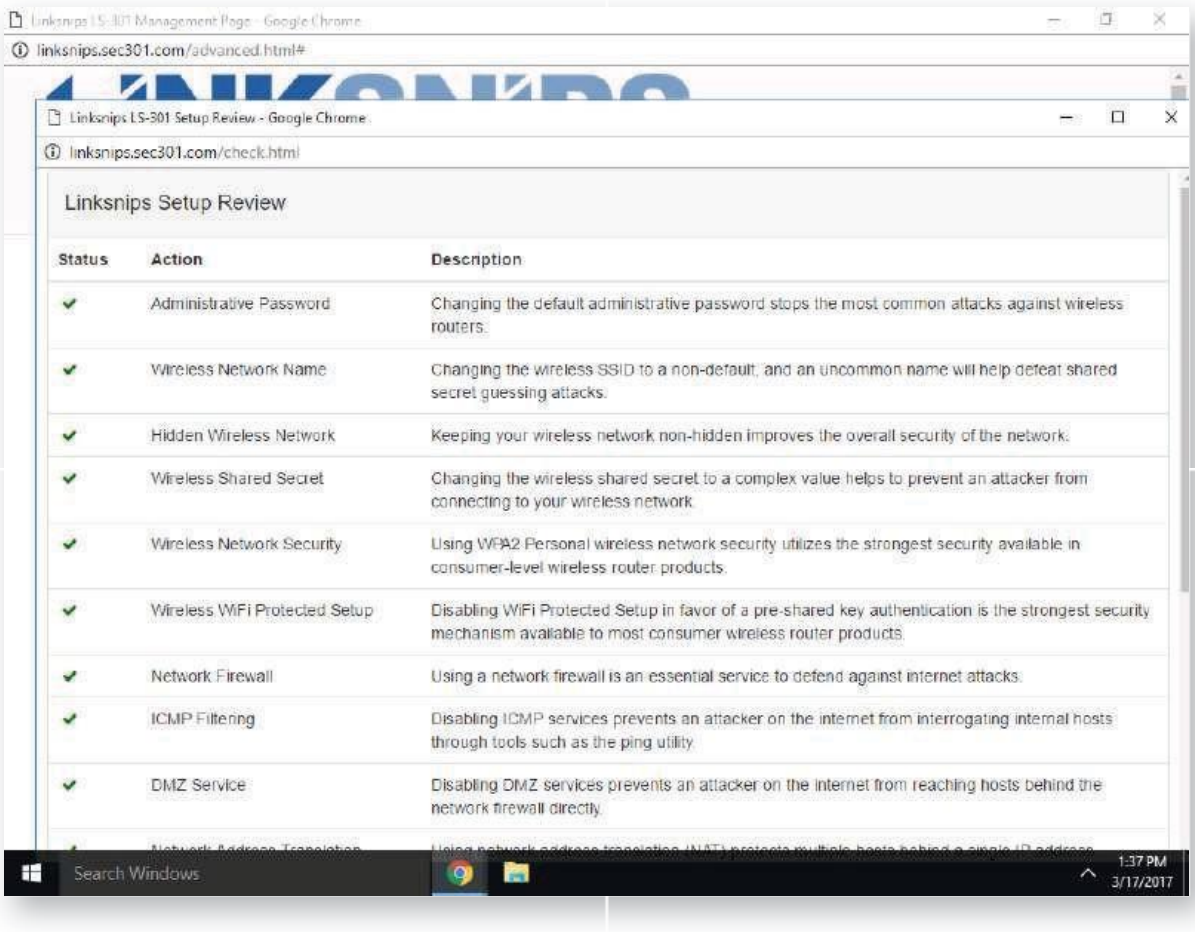
Scroll to the *Universal Plug-and-Play* section and disable all the UPnP configuration options.

27. Save the Administration Settings

Scroll back up to the top of the Linksnips wireless settings section and click **Save Settings**. When prompted, click **OK** to acknowledge the message.

28. Check Your Configuration

Finally, click the *Check Config* link to check your configuration settings against the recommended options. If any of the configuration options include red X marks, return to the associated configuration option and change your configuration until you pass 100% of the tests. Remember to save any of the changes you make to router configuration before rerunning Check Config.



In this exercise, you configured the security settings for the Linksnips wireless AP. In addition to wireless-specific options, you also configured management and access options, network firewall services, network address translation services, and more. Although the Linksnips interface is specific to this product, you will be able to apply these skills to other commercial products as well to securely configure production APs.

Congratulations!

SEC301-4.2: Exercise - Antimalware Scanning

Objective

Use the supplied antimalware tool to identify malware threats to the system. Evaluating the identified threats, quarantine the malware that is a threat to the system. Whitelist the desired software.

Scenario

Virtual Machines

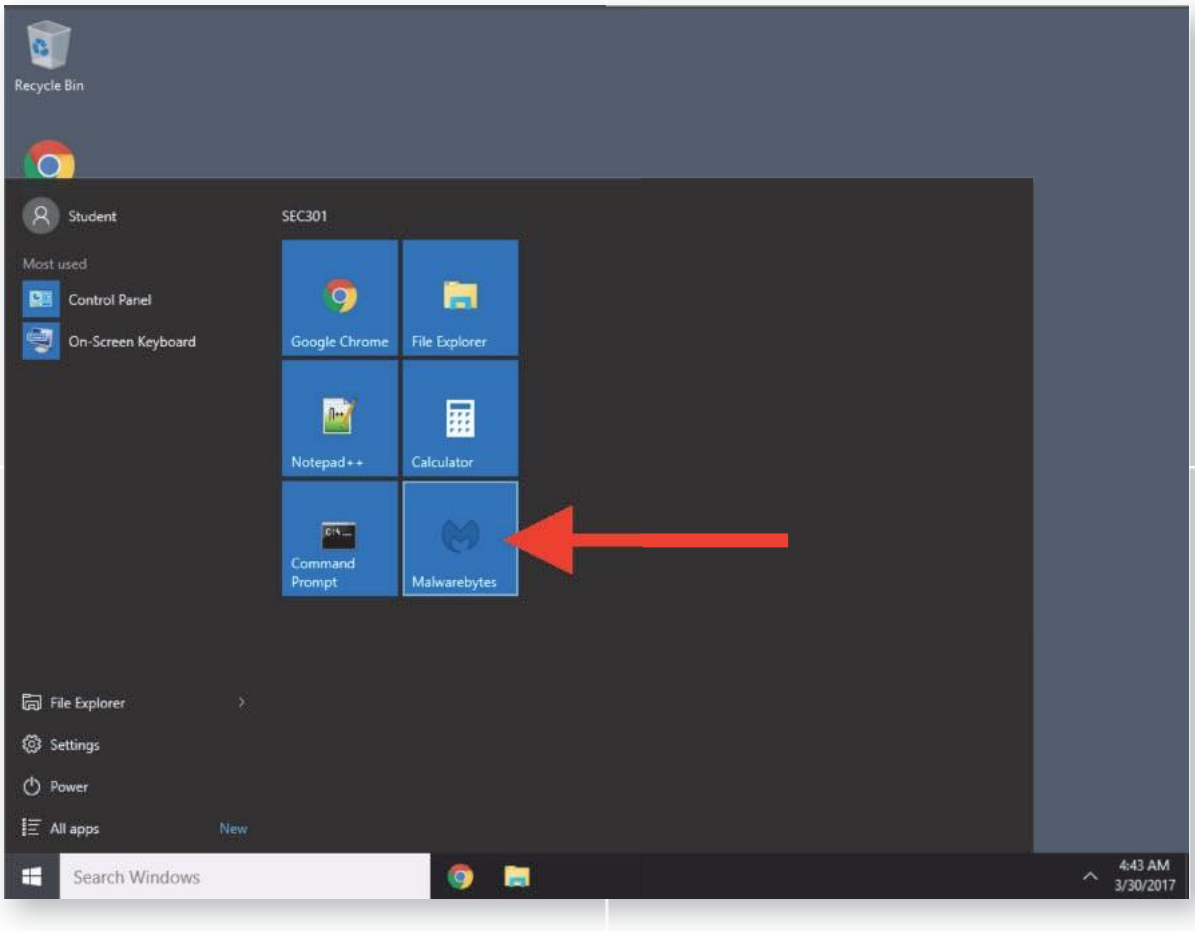
1. SEC301-Win10
2. SEC301-LabServer

Antimalware Scanning

In the exercise, you use the popular antimalware tool Malwarebytes to scan, identify, and quarantine malware samples on a Windows 10 system. The malware included on the system is real but does not pose any threat to your system or data. You also identify Potentially Unwanted Programs (PUP) that are not malicious, so you "whitelist" them to keep them out of future

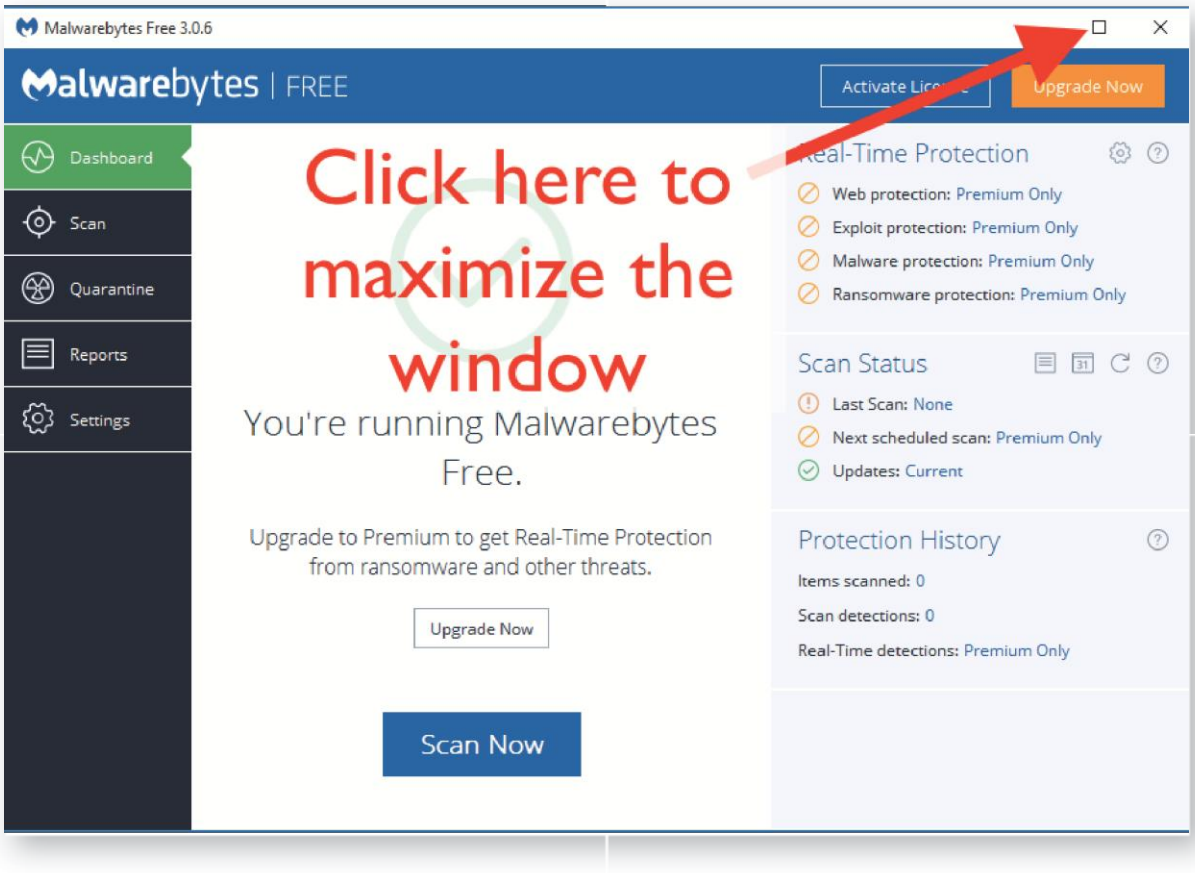
1. **Start Malwarebytes**

   In this exercise, you use the Malwarebytes software to scan the Windows 10 system on LODS for malware threats. Start the Malwarebytes software by clicking on the **Malwarebytes** icon from the Start menu.



2. **Maximize the Malwarebytes Window**

   Maximize the Malwarebytes application window to fill the screen, as shown in the screenshot for this task.



3. **Start the Scan**

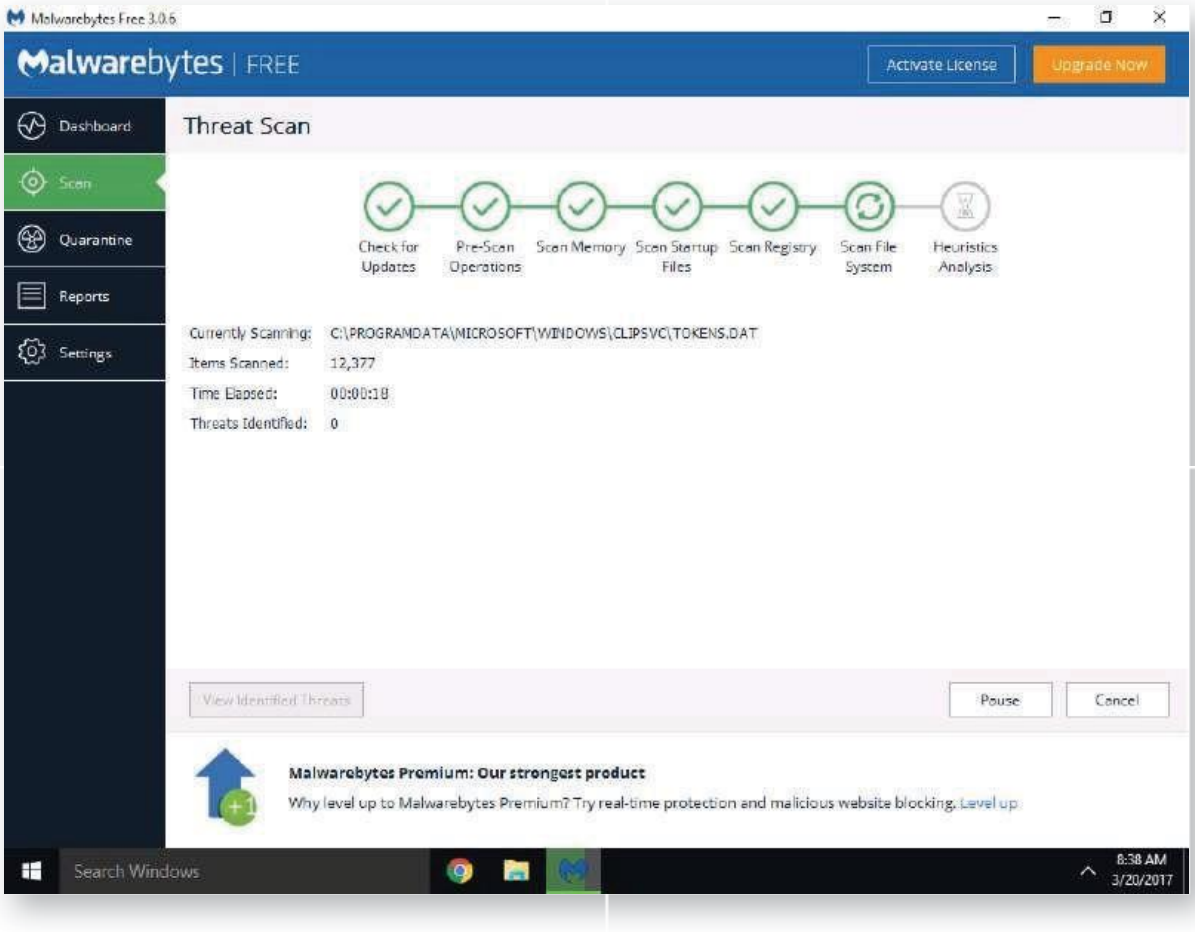
   Click the Malwarebytes **Scan Now** button to start the scan.

   > Depending on the size of the registry, the number of files on the system, and the CPU and hard drive performance of the system, a Malwarebytes scan can take several minutes to several hours to finish scanning. In this exercise, the scan will complete in about four minutes.

4. **Watch the Scan**

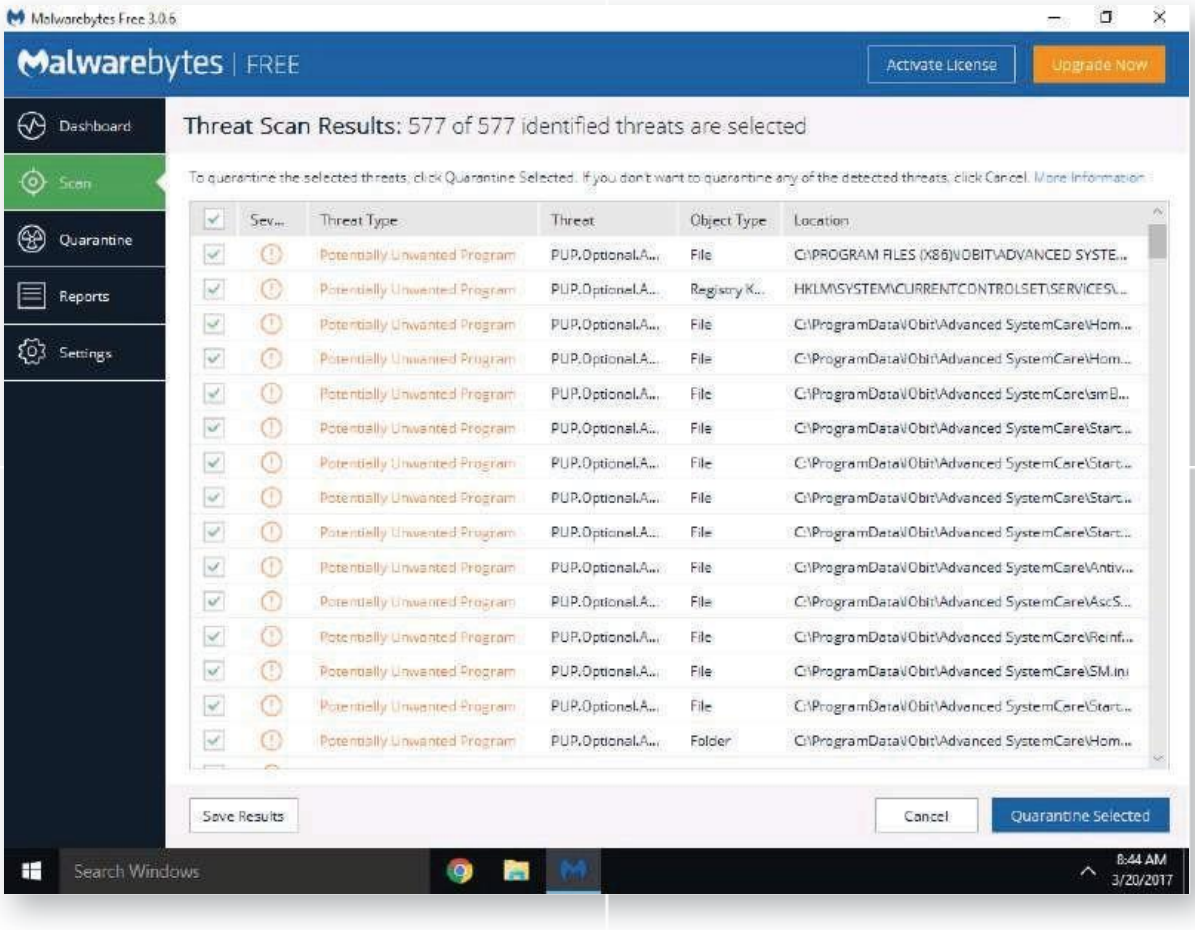
   Malwarebytes uses several techniques to scan for and identify malware threats including in-memory scanning, startup file analysis, registry analysis, and file system scanning. Wait several minutes for the scan to complete. When Malwarebytes finishes the scan, it displays the *Threat Scan Results* window. Move on to the next step when this window appears.

   > ⚠ In this exercise, you scan the Windows 10 computer inside of LODS. You are not scanning your own laptop - so when you find malware, it is not on your computer, it is on ours.



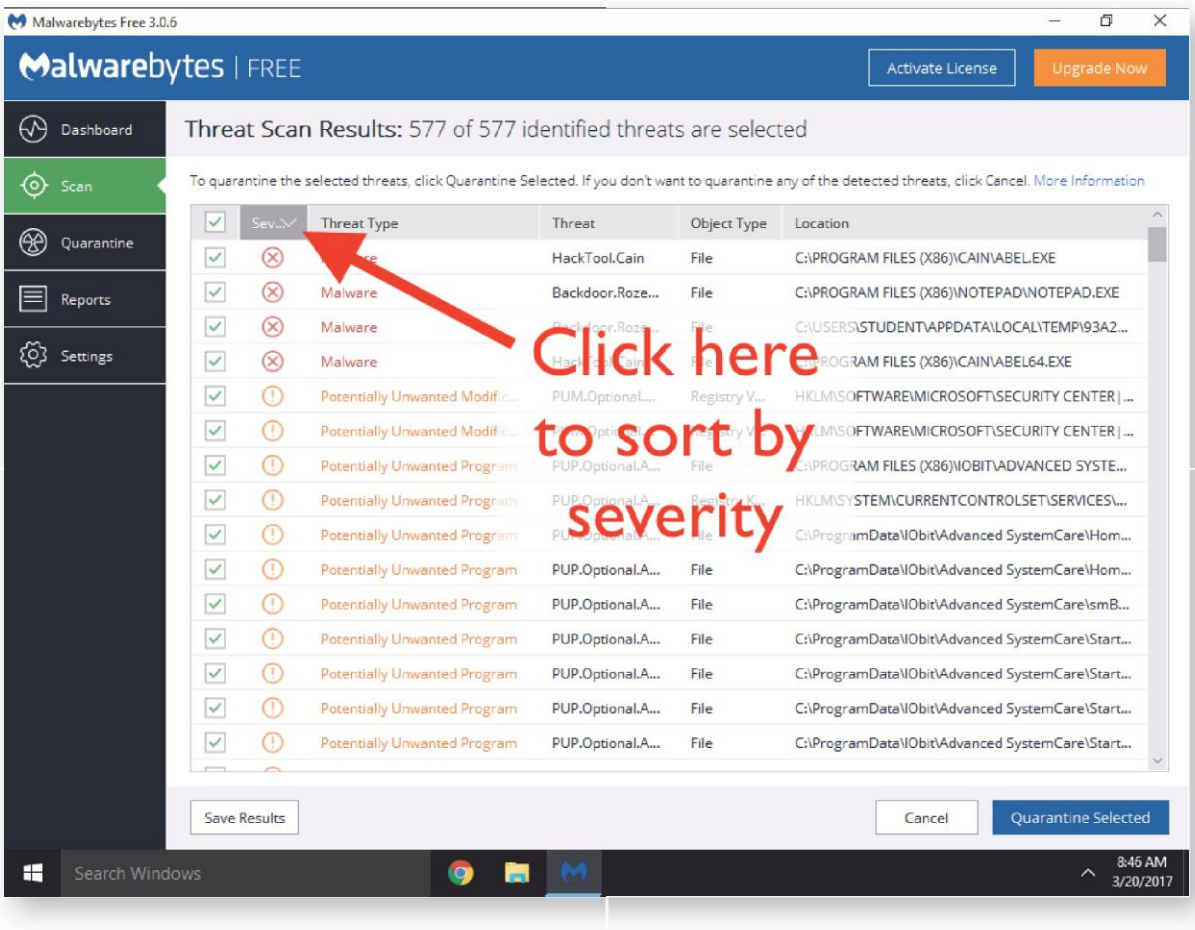
5. **Examine Scan Results**

   The Malwarebytes analysis identifies several hundred system threats. However, not all of them are equal. We need to examine the threats to identify corrective actions.



6. **Sort Threats by Severity**

   Click on the 2nd column header (labeled ***Sev...***) to sort the identified threats by severity. This action shows high-priority malware threats at the top of the list, followed by Potentially Unwanted Modifications (PUMs) and Potentially Unwanted Programs (PUPs).

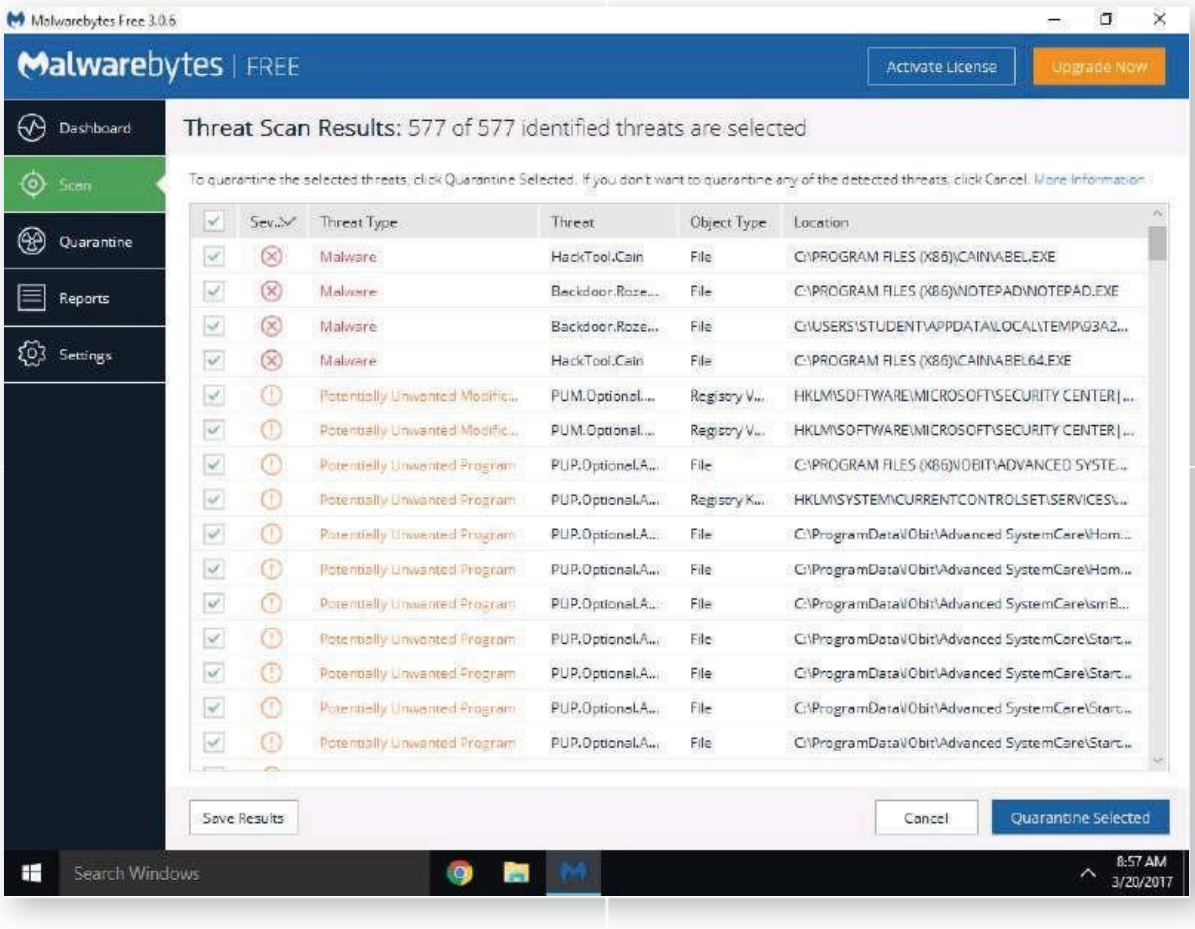


7. **Examine Malware Threat Types**

   The *Threat* column in the scan results provides additional information about the identified Malware, PUM, or PUP. Malwarebytes threats are identified in the 4th column of the threat scan results, and can be one of the following:

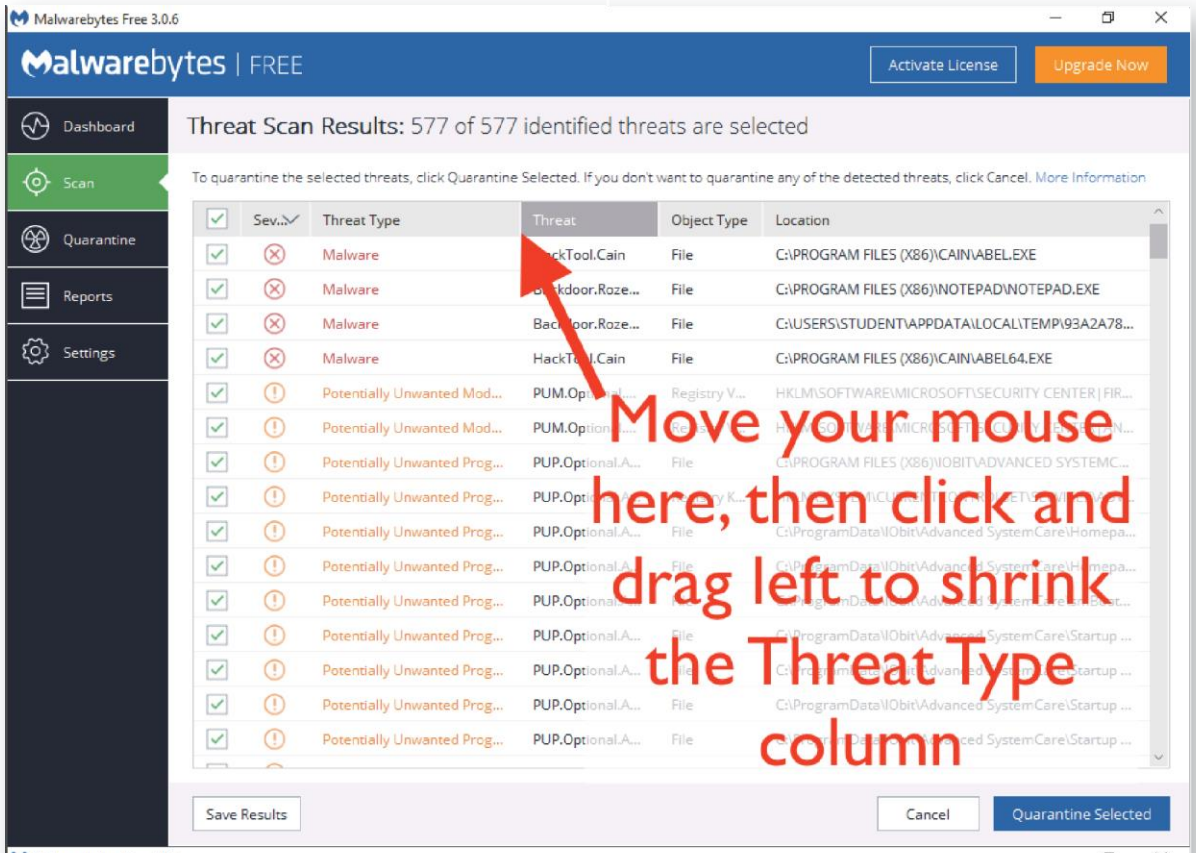
   - HackTool - Software that can be used to conduct hacking attacks
   - Backdoor - Specific trojan horse software that grants an attacker remote access to the system
   - PUM - Potentially Unwanted Modification, with many sub-classifications
   - PUP - Potentially Unwanted Program, with many sub-classifications
   - FraudTool (not pictured) - FraudTools are malware that impersonates legitimate software
   - Trojan (not pictured) - Generic description for any trojan horse software not explicitly categorized
   - RiskWare (not pictured) - Generic description for any other threat not categorized in the other types

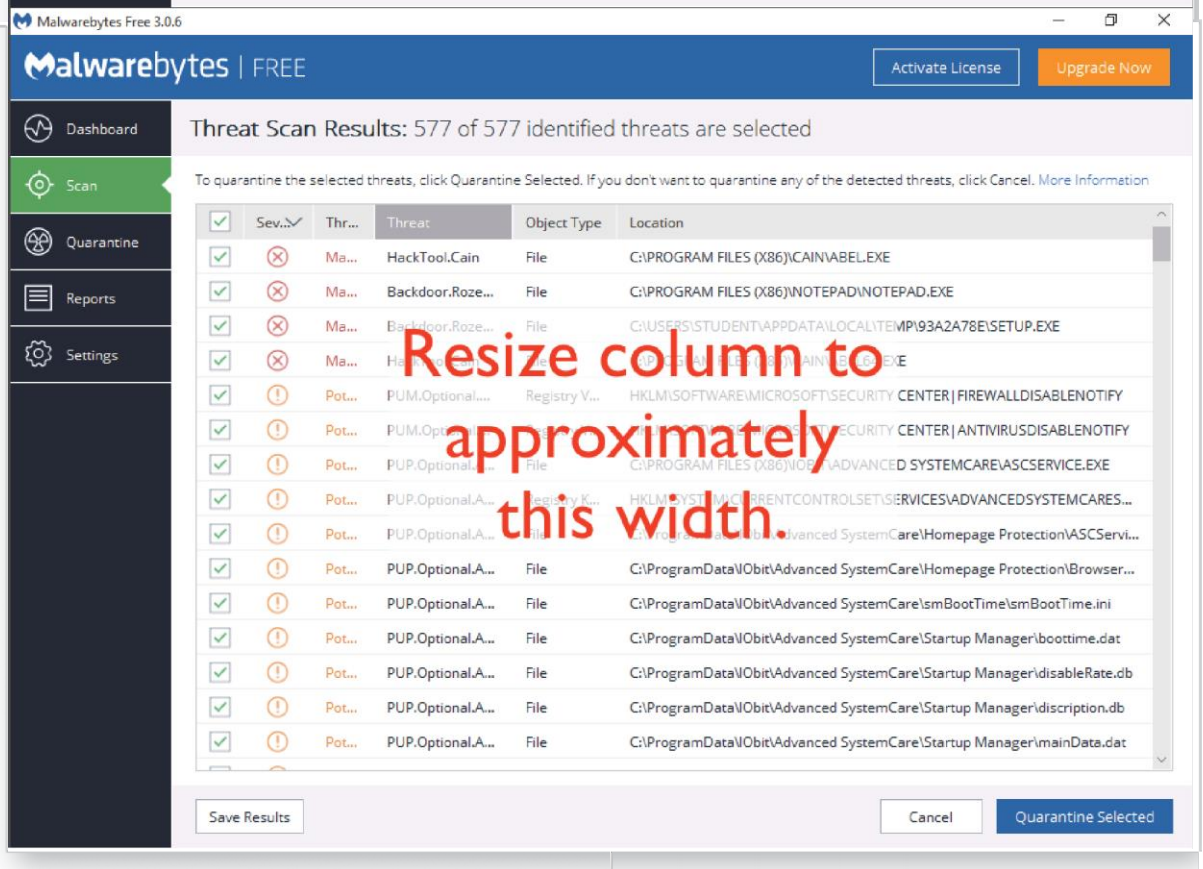
Malwarebytes identified four distinct malware threats: Backdoor, HackTool, Potentially Unwanted Modification, and Potentially Unwanted Program. Next, we continue to investigate these findings in the exercise.



8. **Resize Threat Type Column**

   To see the Location column, we need to resize the Threat Type column width. Move your mouse immediately between the *Threat Type* and *Threat* columns in the header. Click and drag the column left to make it smaller, as shown in the screenshot for this task.





9. **Examine Potentially Unwanted Motifications**

The two scan result entries immediately following the last Malware threat type are Potentially Unwanted Modifications. These entries represent changes to the system that may reduce the effective security of the system.

The *Location* information for this *Registry Value* threat type indicates that the values relate to the Microsoft Security Center, FIREWALLDISABLENOTIFY, and ANTIVIRUSDISABLENOTIFY entries. These changes were applied by the system administrator, disabling notifications for firewall services and anti-virus services on the Windows 10 system.

Applying Malwarebytes fixes to *Potentially Unwanted Modifications* can undo settings implemented by the system administrator. It's a good idea to check with your system administrator before applying the changes recommended by Malwarebytes.

10. **Examine Potentially Unwanted Programs**

    Immediately following the two PUM entries are several hundred *Potentially Unwanted Program* entries. The location information for these entries is all related to the Advanced System Care (ACS) software package by IObit. The ACS software can be used for a malicious purpose but is also for malware removal and other system optimization functions by administrators.

Malwarebytes does not give us any additional information about why the ACS software is categorized as a PUP, though it may fall under the broad category of "obtrusive, misleading, or deceptive advertising, branding, or search practices" (https://www.malwarebytes.com/pup/).

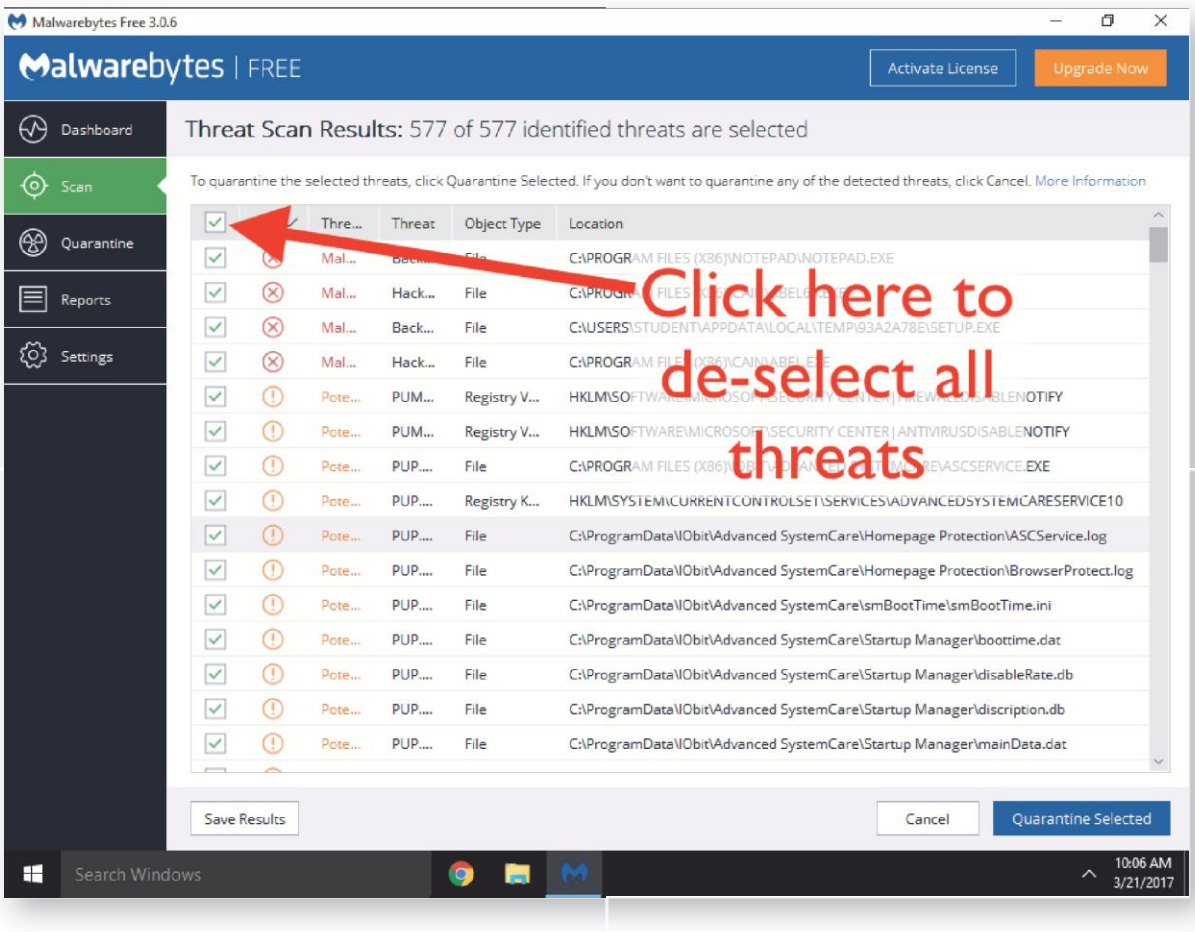
11. **Malwarebytes Analysis Summary**

    For three types of threats identified by Malwarebytes, we want to remove the malware threats, retain the potentially unwanted modifications (per compliance with corporate policy), and whitelist the potentially unwanted program since the administration staff put it in place. We whitelist the PUPs, so they do not show up in future reports. By reducing clutter in the report, we make it easier to identify future malware infection.

    We'll continue the steps in this exercise working from this executive summary of the scan results.

12. **Deselect Scan Results**

    Click on the checkbox icon in the header row of the scan results to de-select all the scan results, as shown in the screenshot for this task.



13. Select the Four Malware Threats

Click the checkboxes next to each of the four malware threats (denoted with the red X in the severity column).

14. Quarantine Selected Items

Click the **Quarantine Selected** button to quarantine the selected malware threats. When prompted *What should Malwarebytes do with the 573 unchecked items*, choose **Ignore Once**.

⚠ Malwarebytes will prompt you to reboot when you complete this step.

DO NOT click Reboot yet!

(If you do, it's OK, just return to step 1 of this exercise)

When choosing to quarantine selected items, Malwarebytes offers you the opportunity to ignore unchecked items, or to quarantine everything identified. In a situation with several hundred threats identified, it is unwise to quarantine everything unless you are sure *every* entry is safe to quarantine. Instead, use the quarantine feature to exclude non-malicious software, then repeat the scan to evaluate the remaining items.

15. **Do Not Reboot When Prompted**

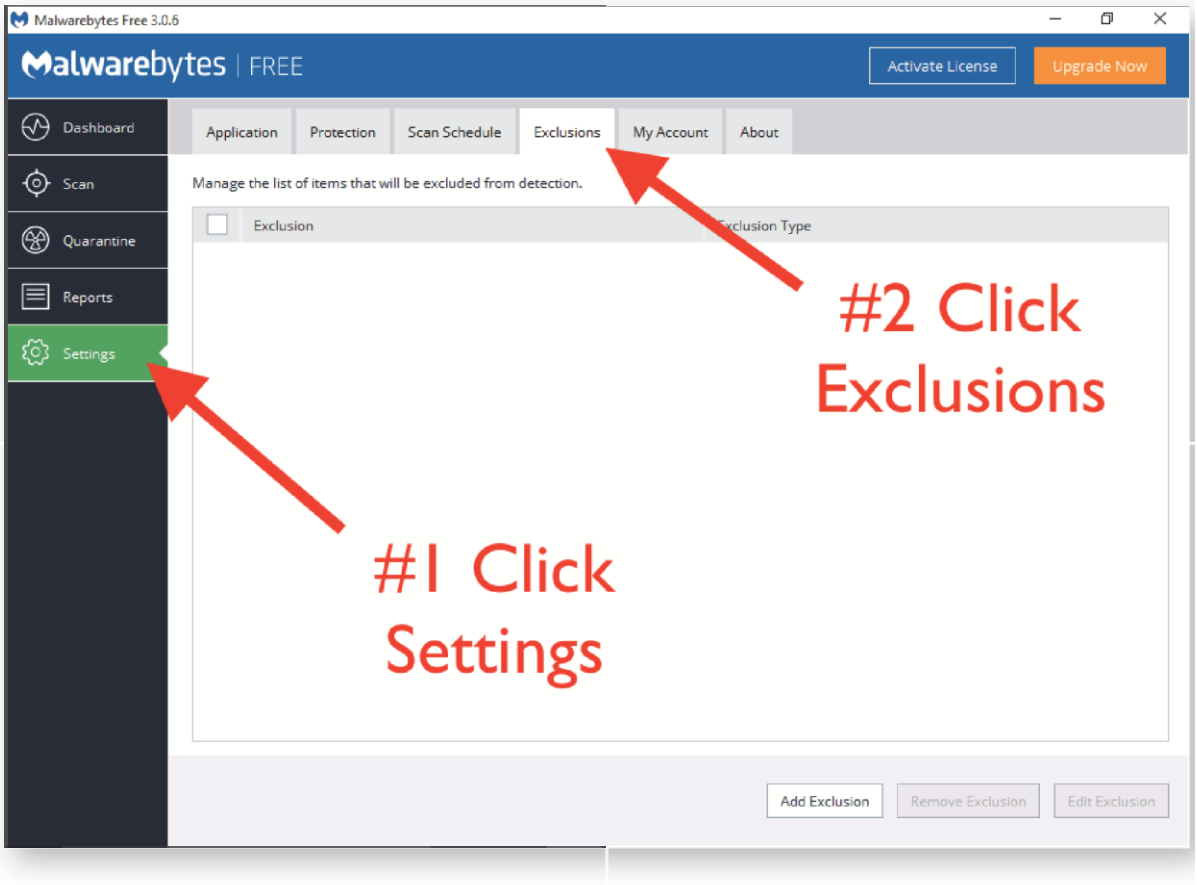
    After quarantining the selected malware threats, Malwarebytes asks if you would like to restart. Restarting after removing malware is a good idea, but for this exercise choose **No** when prompted to reboot.

    ⚠ Notice the instruction below directs you to choose No when prompted to reboot.

    Normally, outside class, you would reboot here, but in the interest of time, please do not do so.

16. **Open Malwarebytes Exclusions Tab**

    Next, you add an exclusion to eliminate the IOBit software from future Malwarebytes scans. Click the Settings button on the left side of the Malwarebytes window, then click the Exclusions tab at the top of the window, as shown in the screenshot for this task.



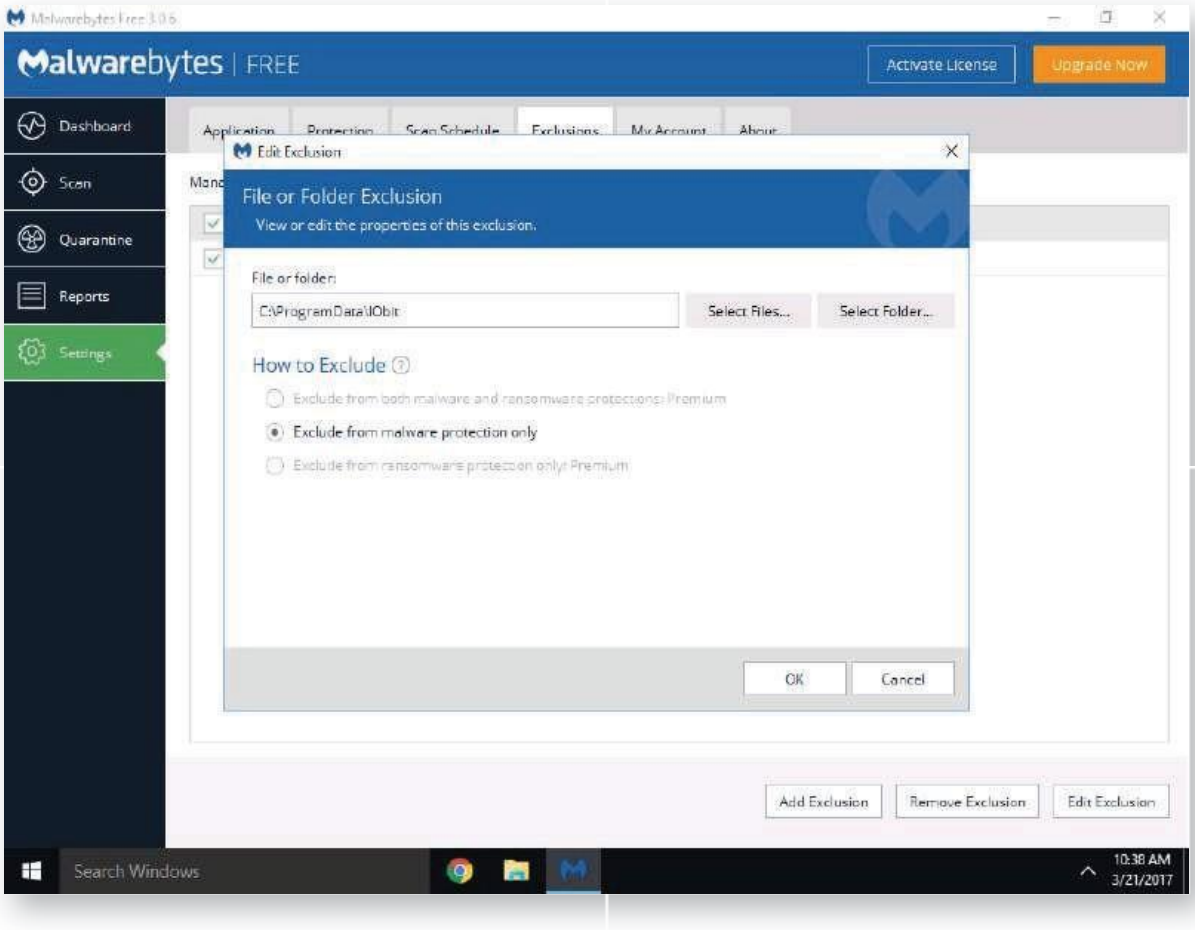
17. Click Add Exclusion

From the Malwarebytes *Settings* window at the *Exclusions* tab, click **Add Exclusion**.

18. Complete the Add Exclusion Wizard

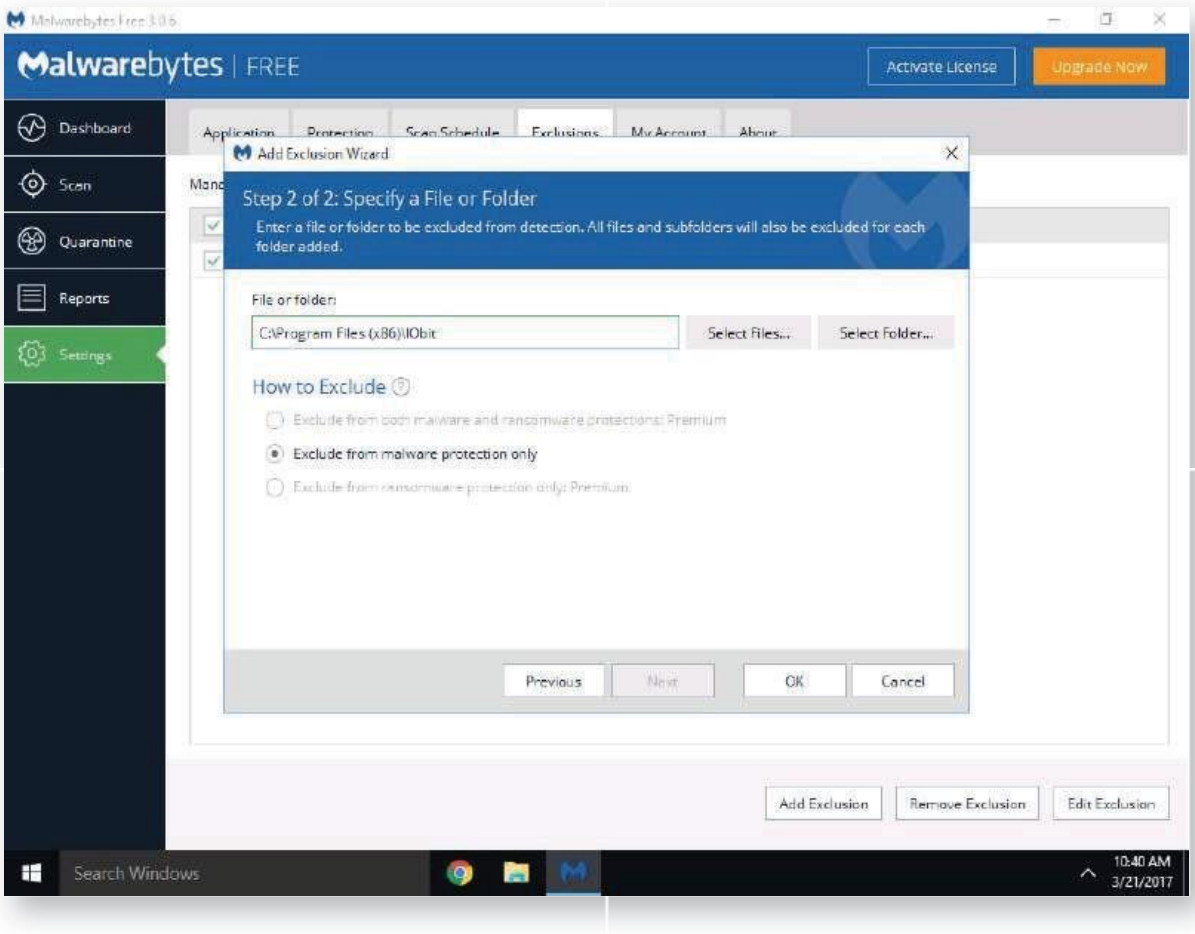
Malwarebytes will walk you through the process for adding an exclusion with the *Add Exclusion Wizard*. Complete the steps as described here:

1. With *Exclude a File or Folder* selected, click **Next**
2. Click **Select Folder**
3. Click on the **Local Disk (C:)** drive on the left of the *Select Folder* dialog
4. Double-click on the **C:\ProgramData** directory
5. Click on the **IOBit** directory (just one click)
6. Click **Select Folder**
7. Click **OK** to complete the wizard



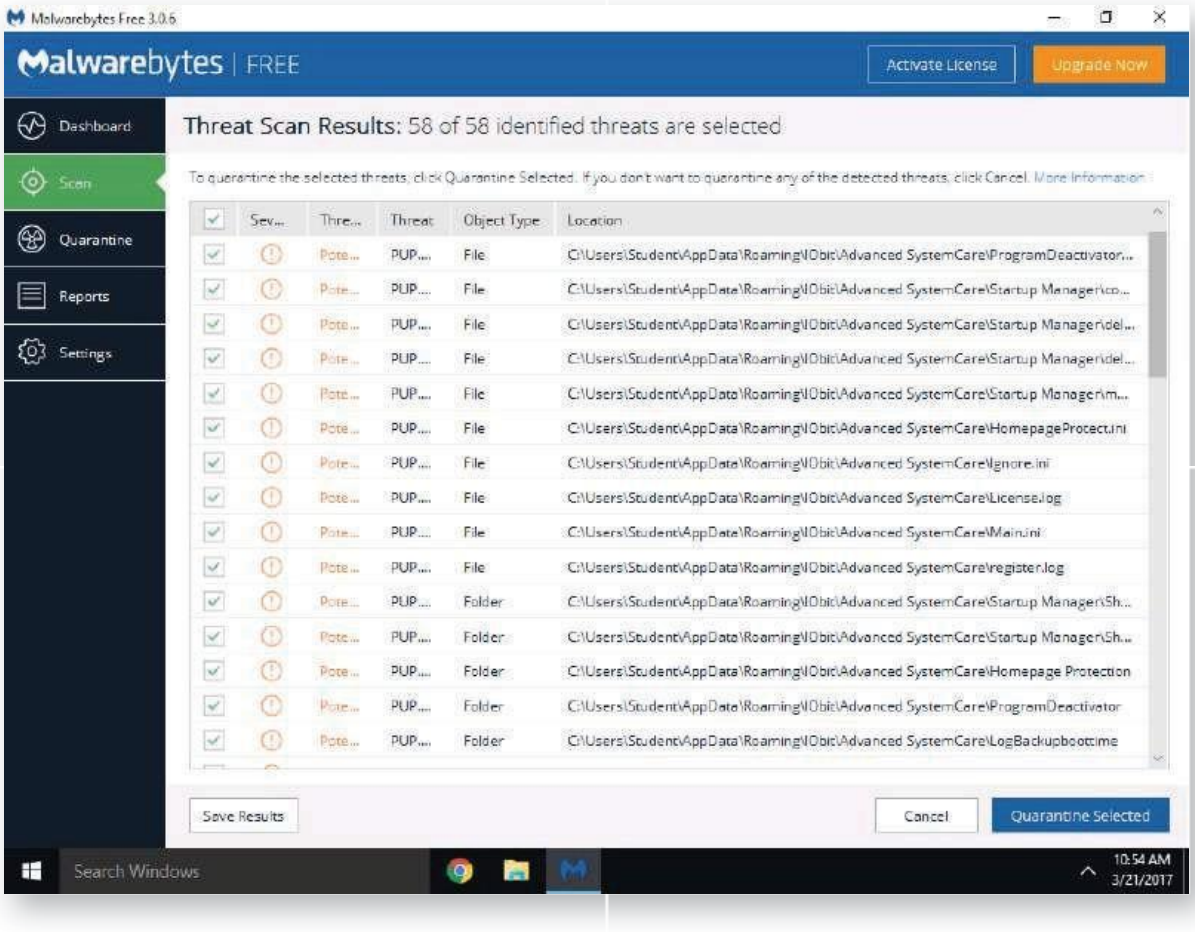
19. **Add a 2nd Exclusion**

    Complete the same process again to add a 2nd exclusion directory. This time, select the **C:\Program Files (x86)\IOBit** directory for the exclusion.



20. **Start a New Scan**

    Click on the **Dashboard** button on the left of Malwarebytes. Then click the **Scan Now** button to repeat the scan.



21. Watch the Scan

Allow Malwarebytes to complete the new scan. This will take about four minutes to complete.

22. Review Revised Scan Results

The results from the 2nd Malwarebytes scan only reports 58 threats, most of which are related to the ACS software installation in directories not included in the exclusion rules. Although these items are still false-positive entries, the exclusion rules eliminated much of the content, allowing you to focus more closely on the remaining issues.

⚠ Note: While whitelisting software to exclude it from the report is often necessary to reduce clutter in those reports, there is a potential problem. In the last several steps, you instructed Malwarebytes to ignore specific directories. If malware ever finds its way into those directories, Malwarebytes cannot find it now.

Security is ALWAYS about tradeoffs...

Malware scanning can require several iterations of the scan/exclude /repeat process. While the initial scan results were too numerous to review quickly, identifying the ACS software and adding it to the exclusion filter allowed us to repeat the scan, this time getting a smaller subset of threats to review.

Using Malwarebytes to scan a system for malware is a straightforward task. However, interpreting the results can be complicated, and requires insight into the use case for the system and the software needed for the user to do their job. The identification of Potentially Unwanted Programs is a crucial feature of antimalware tools. You must carefully consider the scan results when choosing to quarantine or whitelist identified software.

Congratulations on completing this exercise!

SEC301-5.1: Exercise - Firewall Builder

Objective

In this lab, you utilize the open source utility called FirewallBuilder to create a simple set of firewall rules. You look at a network diagram to determine proper addresses and direction of travel. You also look at a policy dictating what traffic should be allowed and denied.

From this information, the required firewall rules should be easy to determine.

Scenario

Virtual Machines

1. SEC301-Win10
2. SEC301-LabServer
3. SEC301-WIMIP
4. SEC301-SmallishRouter

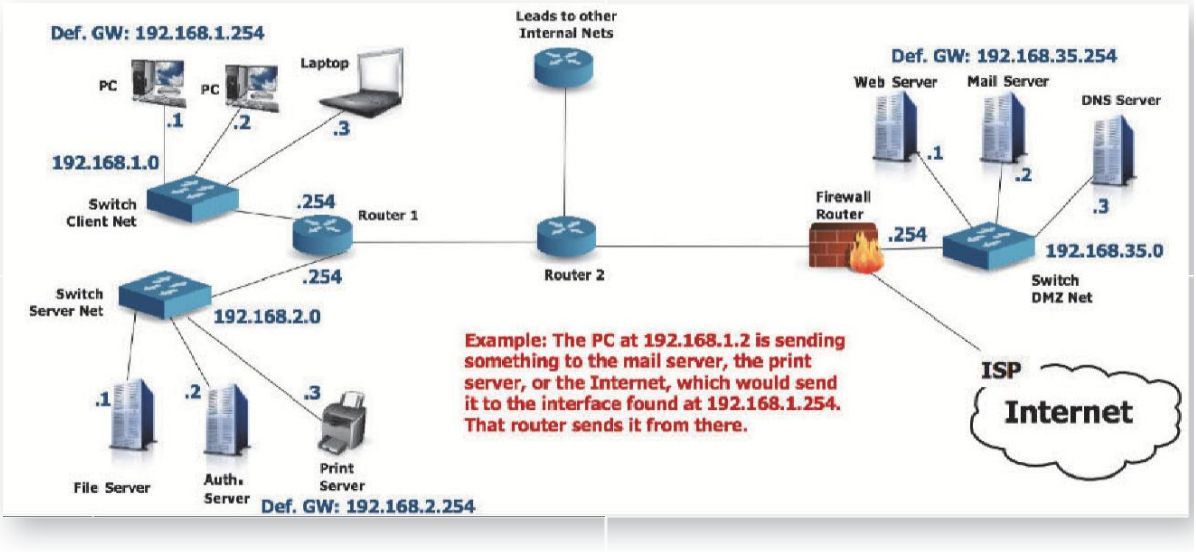
Firewall Builder

In this exercise, you create a simple eight rule firewall policy. To do so, you use the Firewall Builder tool. This free utility helps companies around the world create and manage complex firewall rulesets.

Here we barely scratch the surface of the total functionality of Firewall Builder. There is much more it can do. However, this exercise will give you an excellent introduction to the concept of firewall rules.

1. **Review the network diagram**

   Both in your course book and in this screenshot is the network diagram from our networking discussion earlier in the class. These are the addresses and the network configuration our firewall rules will use.



2. **Open File Explorer**

   Open the File Browser by clicking on the "yellow manilla folder" on the bottom of the Windows 10 screen.

3. **Access the Documents Directory**

   In the File Browser, double-click on **Documents**.

4. **Open the Firewall Config**

   You should see a file called **FW_Config**. This file contains a partial firewall configuration. Double-click this file, so it opens in FirewallBuilder.

5. **Prepare FirewallBuilder**

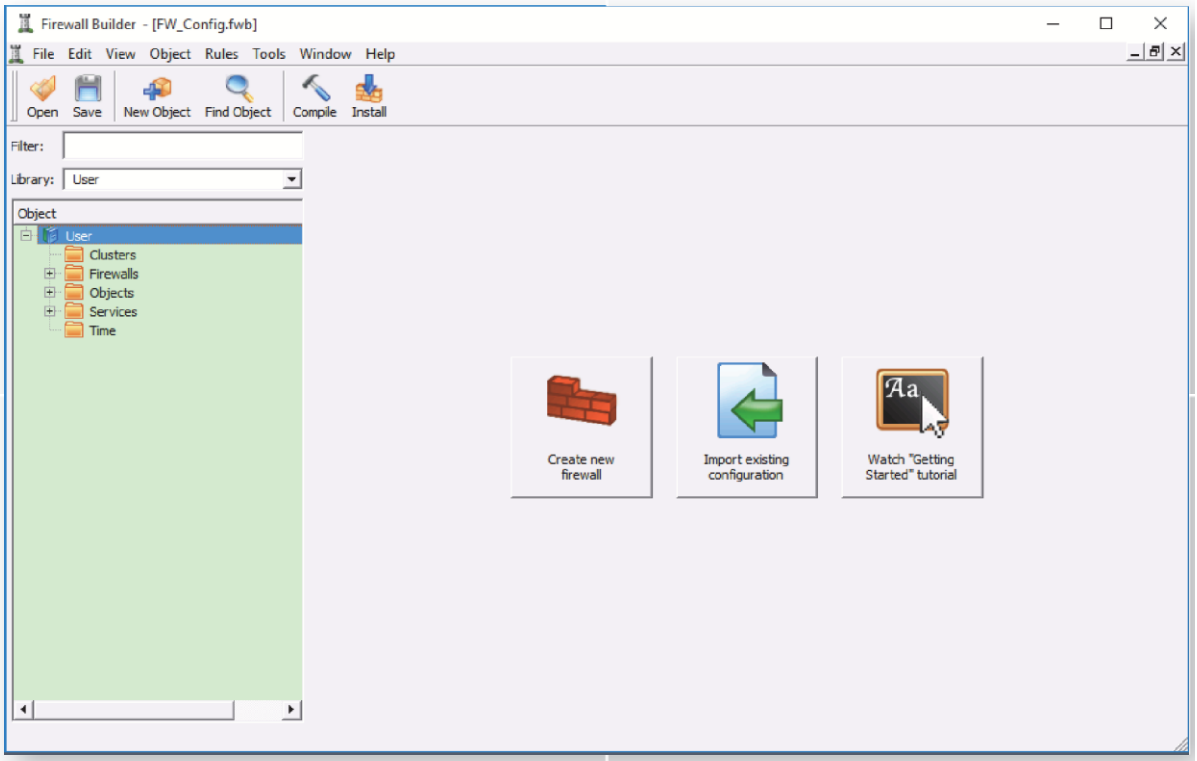
FirewallBuilder should open looking exactly like that in the screenshot. Before we tour the interface, we want to open a few things up so we can work. Please follow these steps exactly:

1. Maximize the FirewallBuilder screen by clicking the button in the upper-right of the Firewall Builder screen.
2. On the left is a green box labeled "Object". In that box, click the plus sign next to "Firewalls." You should see a firewall.
3. Now click the plus sign next to the firewall, you should see "Policy," "NAT," Routing," Int0", "Int1", and "Int2".
4. Double-click on the word "Policy." You should see some things appear in the white window on the right.
5. Click the plus sign next to "Objects." More folders appear including "Hosts" and "Networks."
6. Click the plus sign next to "Hosts".
7. Click the plus sign next to "Networks".

When you have finished all of these steps, you should see the screenshot included in the NEXT Task. We will discuss what all this means there - so go ahead and click "Done" now.

When you work with an interface of this sort, and you see plus signs, those mean to "expand" a section. In other words, some of the elements of the interface are hidden from you. Click the plus sign to expose them.

Likewise, if you see a minus sign, it means you can "collapse" an area to hide some elements of the interface. This phenomenon is common in computing and familiarity with it is necessary.



6. Understand the Interface

Open the screenshot for this task. You can see that the interface of Firewall Builder is fairly straightforward.

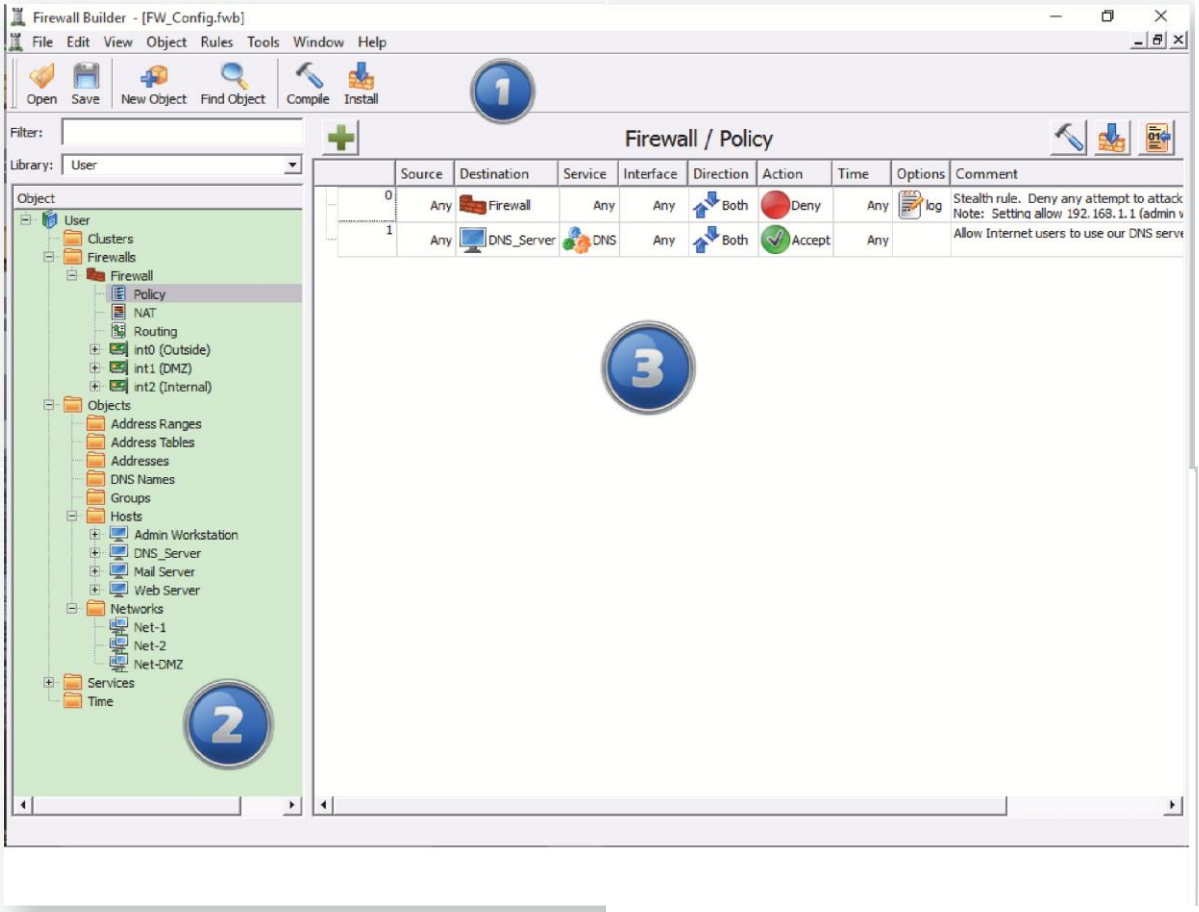
1. You have the area labeled 1 where you find the menus and button bar. Take note of the save button, as we will make extensive use of it.
2. The box on the far left labeled 2 is the object window. Here you find all the objects used in rules. Objects include the firewall itself, hosts, networks, and others used as source and destination of traffic in rules. We also have service objects. For example, there is an HTTP service object that we use in a rule to allow or deny web traffic.
3. In the area labeled 3, we find the actual rules. We have already created two for you and will take a look at those in a moment.

Click Done for an explanation of the objects and rules.

⚠ Firewall Builder uses the same premises almost all firewall products. You have objects for hosts, networks, and services that you will put into rules. At the most basic, a rule consists of:

- A source object
- A destination object
- A service Object
- An action of Allow or Deny

There can be other fields, but those are the key fields.



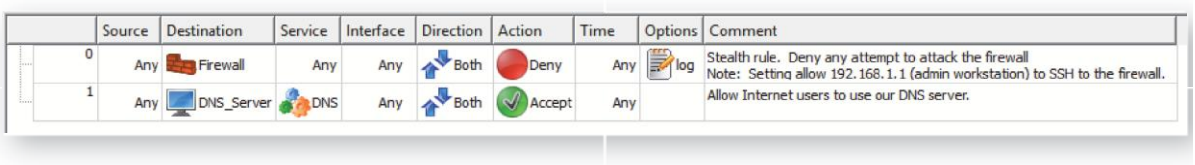
7. **Where we are**

To get things started, we already created the objects you will need for this lab and the first few rules. (You will create another object and several rules.) As you look through the objects and rules, remember that we are using the network diagram from the networking section earlier in the class.

- In the Objects window, you see the Firewall object itself. This object is almost always a requirement.
- You see Host objects for an Admin Workstation (192.168.1.1), the DNS Server, Mail Server, and Web Server.
- You see Network objects for Net 1, Net 2, and the DMZ.

In the Firewall/Policy area, you see two rules are already present.

- A stealth rule with Any source, Firewall destination, Any service Deny & Log. This rule prevents attacks against the firewall itself. (note: there is a configuration setting allowing the Admin Workstation SSH access to the firewall for configuration purposes)
- A rule allowing the world to utilize our DNS server - Source Any, Destination DNS Server, Service DNS, Action Accept



	Source	Destination	Service	Interface	Direction	Action	Time	Options	Comment
0	Any	Firewall	Any	Any	Both	Deny	Any	log	Stealth rule. Deny any attempt to attack the firewall Note: Setting allow 192.168.1.1 (admin workstation) to SSH to the firewall.
1	Any	DNS_Server	DNS	Any	Both	Accept	Any		Allow Internet users to use our DNS server.

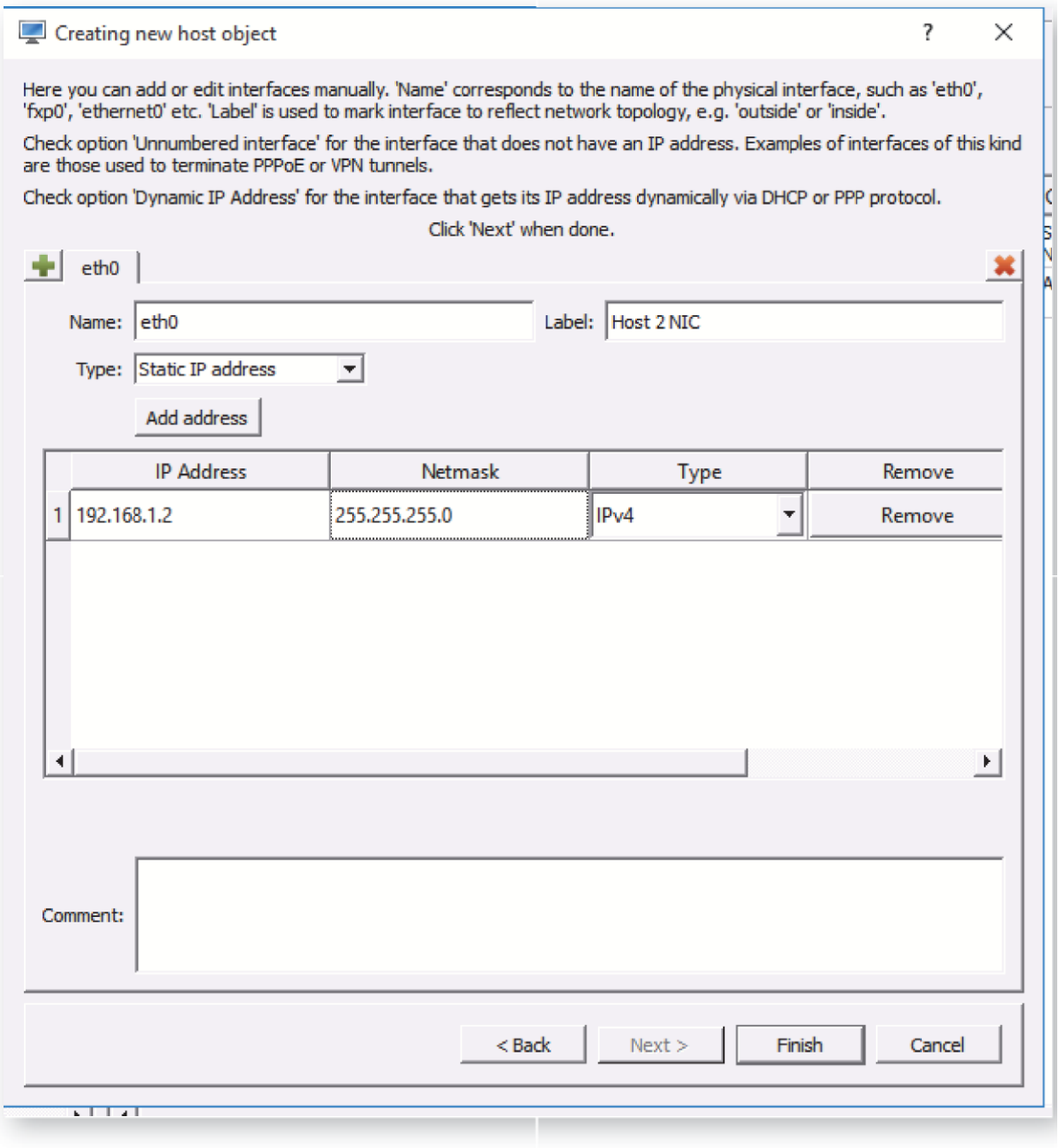
8. **Build an Object**

We have already created the objects you will need to complete the lab, but knowing how to create an object is an important skill. We will create a host object here (creating a network object is very similar).

1. In the button bar, click on the "**New Object"** button
2. A drop-down appears listing all of the types of objects - choose the third on the list "**New Host**"
3. For the name, enter "**Host 2"** and click on "**Next"**
4. Keep **"Configure interfaces manually"** and click **"Next"**
5. In the Name: field, type "**eth0"** and in the Label field, type "**Host 2 NIC**" (for this type of host, these entries do not matter)
6. Click the "**Add Address"** button
7. Click in the IP Address field and type "**192.168.1.2**"
8. Click in the Netmask field and type "**255.255.255.0**"

The end result should look like the screenshot. When you are satisfied, click Finish. When you do, an area at the bottom of the screen opens where you can edit this object. We are done working with it, so we can click the small X about halfway up the right side of the screen to close that Window (double-click an object to re-open that window).

Objects of this nature are just about universal in firewall configuration software. You create objects for hosts, networks, and so on. A nice feature of many firewalls (including Firewall Builder is that you can group objects. So say you need the source of a specific traffic to be three different networks, you can create a group of those three networks and place the group in the source field of a rule. This is just one example of many.



9. **Add a DNS Rule**

We have a rule allowing the world to use our DNS server. Now we need a rule allowing our DNS server to send DNS queries out. Follow these steps to add and configure this rule.

1. Click on rule #1 (the second rule). Then right-click and select "**Add new rule Below**." A new default rule appears with source=any, destination=any, service=any, interface=any, direction=both, action=deny, time=any, options=log. This rule will appear everytime we add a new rule. If we left it unchanged, it would block all traffic.
2. Click the **DNS_Server** object and drag it to the **Source** column. (we will leave the Destination = Any)
3. Right-click on the **Deny** in the **Action** column and select **Accept**.
4. Right-Click on log in the **Options** column and select **Logging Off** (we do not need to log all DNS queries).
5. Above the green **Object Window** is a **Library** drop-down. Click the arrow next to it and select **Standard**
6. In the **Filter** box above Library, type **DNS**. This filters out all other Standard objects and makes it much simpler to find the one you are looking for
7. Click on the **DNS** object and drag it to the Service field of our rule.
8. Double click on the **Comment** field of our rule. A box opens at the bottom of the screen. Type: **Allow our DNS server to send Queries**. Click the black **X** on that bottom window when done
9. There is NO NEED to modify the Interface, Direction, or Time fields on this rule.

Your first rule is done! See the screenshot to see it it matches.

⚠ In Firewall Builder, there are User Objects that we create, and there are Standard objects that are already there. Standard objects include all of the services we might need - for example, there is an HTTP object and an HTTPS object, etc. You cannot edit Standard objects, but you can use them in rules.

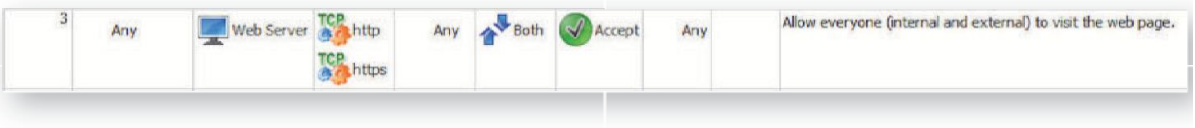
10. **Create a Rule for our Web Server**

Now that we know how to create a basic rule, the instructions will be less detailed. We need a rule allowing everyone in the world to visit our web server in our DMZ.

1. Click on Rule 2, right-click and select **Add New Rule Below**
2. We still have the standard objects open in the objects window but are filtering for DNS. Next to the Filter window is a red X. Click this to clear the DNS filter.
3. Now, in the Filter window, type HTTP. You should see objects for both HTTP and HTTPS.
4. Drag BOTH objects to the Service field of rule 3
5. Again, click the red X next to the Filter field, then use the Library drop-down to select User so we can see our objects.
6. Click and Drag Web Server object into the Destination column (we leave the source=any)
7. Right-Click the Action field and change it to Accept.
8. Right-click on the Options field and change it to Logging Off
9. Double-click on the Comments field and type: Allow everyone (internal and external) to visit the web page.

Check the screenshot to see if it matches.

Some may ask why we don't need a rule specifically allowing our internal computers to access the web page. Actually, you could and you would not be wrong. But since our internal networks are a part of "Any", having this single rule with a source of Any gets the job done.



11. **Create a rule to receive email**

We now need the rule to receive email. By now, you should know how to create a rule, so we will give you rhe criteria, and you can place it into an appropriate rule.

- Add rule # 4 below
- Source = Any
- Destination = Mail Server
- Service = SMTP
- Interface = Any
- Direction = Both
- Action = Accept
- Time = Any
- Options = Logging OFF
- Comment = Allow receiving email from external servers

Screenshot is on the next task...

12. **Create a rule for Sending Outgoing Email**

    Next, we need the rule to send outgoing email to the world. This time, we only give you the key fields:

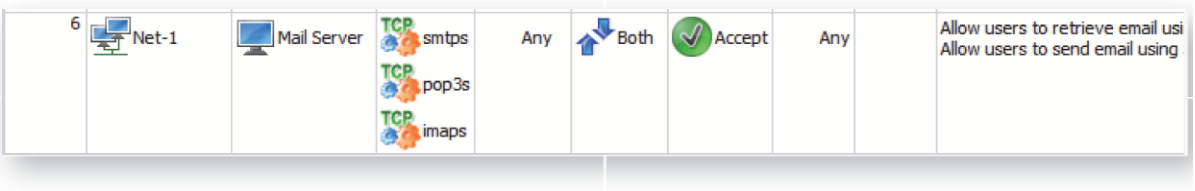
    - New rule # 5
    - Source = Mail Server
    - Destination = Any
    - Service = SMTP
    - Action = Accept
    - Comment = Allow our mail server to send an email out to the world.

    See the screenshot to check the two Email rules.

13. **Rule for our users to send email via Mail Server**

Now we need a rule allowing our users to send email via the mail server. Our policy requires that this is done using encrypting protocols only. Therefore, email is sent to the mail server using SMPTS. Email can be retrieved from the mail server using either POP3S or IMAPS.

- New rule #6
- Source = Net-1
- Destination = Mail Server
- Service = SMTPS, POP3S, IMAPS
- Action = Accept
- Option = Logging OFF
- Comment = Allow users to retrieve email via POP3S or IMAPS. Allow user to send email via SMTPS.



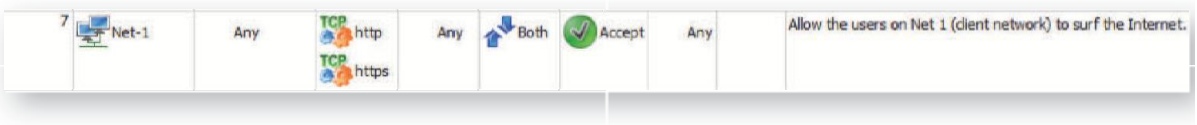
14. **Rule to allow our users to surf the Web**

The policy states that our users can surf the web with HTTP and HTTPS. Therefore, we need a rule allowing this behavior:

- Source Net-1
- Destination Any
- Service HTTP, HTTPS
- Action Accept
- No logging
- Comment Allow the users on Net 1 (client network) to surf the Internet.

(This very abbreviated format is often precisely how rule requirements are communicated to firewall admins. They have to interpret this and create the rule correctly in the software.)

NOTE: If you had multiple internal client networks, you would often create a group containing those networks and use the group in this type of rule.



15. **Create a "Clean Up" Rule**

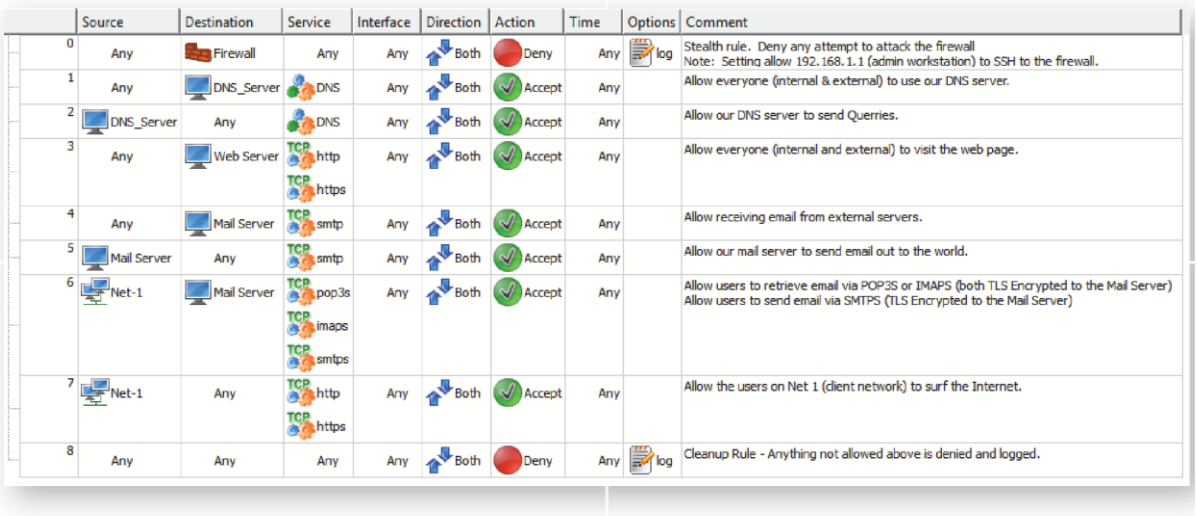
    Almost all firewalls configure for default deny. However, we don't like to leave anything to chance, so we will create our own deny rule. The other problem with default deny rules is that they rarely log the inappropriate traffic they stop.

    - Add a new rule #8 at the bottom
    - Add the comment: Cleanup Rule - Anything not allowed above is denied and logged.

    You are done with that rule.

16. **Review completed Ruleset**

    If you followed all of the instructions exactly, you should have a ruleset that looks just like the screenshot. This same graphic is also in your course book.



	Source	Destination	Service	Interface	Direction	Action	Time	Options	Comment
0	Any	Firewall	Any	Any	Both	Deny	Any	log	Stealth rule. Deny any attempt to attack the firewall Note: Setting allow 192.168.1.1 (admin workstation) to SSH to the firewall.
1	Any	DNS_Server	DNS	Any	Both	Accept	Any		Allow everyone (internal & external) to use our DNS server.
2	DNS_Server	Any	DNS	Any	Both	Accept	Any		Allow our DNS server to send Querries.
3	Any	Web Server	TCP http TCP https	Any	Both	Accept	Any		Allow everyone (internal and external) to visit the web page.
4	Any	Mail Server	TCP smtp	Any	Both	Accept	Any		Allow receiving email from external servers.
5	Mail Server	Any	TCP smtp	Any	Both	Accept	Any		Allow our mail server to send email out to the world.
6	Net-1	Mail Server	TCP pop3s TCP imaps TCP smtps	Any	Both	Accept	Any		Allow users to retrieve email via POP3S or IMAPS (both TLS Encrypted to the Mail Server) Allow users to send email via SMTPS (TLS Encrypted to the Mail Server)
7	Net-1	Any	TCP http TCP https	Any	Both	Accept	Any		Allow the users on Net 1 (client network) to surf the Internet.
8	Any	Any	Any	Any	Both	Deny	Any	log	Cleanup Rule - Anything not allowed above is denied and logged.

17. **View the compiled Rules**

In this configuration, we have an iptables firewall on a Linux computer. You can view the compiled rules by clicking the far-right button on the line above our rules (it has a 01 and an arrow).

You will have to click on "Compile and Inspect files". Click Next - then click Finish. Now to see the compiled rules, click that button one more time. You will see a LOT of things here that you won't understand and explaining them here is beyond the SEC301. But if you scroll down about 3/4 of the way, you will see a line that looks like this:

========================= Table 'filter', rule set policy

Below this line, you see the rules we actually created. Notice you can see both the rule number and the comments to help guide you through which rule is which.

In the screenshot, the author deleted extraneous lines to make things more compact so they can show up on your screen.

```
# ================ Table 'filter', rule set Policy
# Rule 0 (global)
# Stealth rule.  Deny any attempt to attack the firewall
# Note:  Setting allow 192.168.1.1 (admin workstation) to SSH to the firewall.
$IPTABLES -N RULE_0
$IPTABLES -A INPUT  -j RULE_0
$IPTABLES -A RULE_0  -j LOG  --log-level info --log-prefix "RULE 0 -- DENY "
$IPTABLES -A RULE_0  -j DROP
#
# Rule 1 (global)
# Allow everyone (internal & external) to use our DNS server.
$IPTABLES -A FORWARD -p tcp -m tcp  -d 192.168.35.3   --dport 53  -m state --state NEW  -j ACCEPT
$IPTABLES -A FORWARD -p udp -m udp  -d 192.168.35.3   --dport 53  -m state --state NEW  -j ACCEPT
#
# Rule 2 (global)
# Allow our DNS server to send Querries.
$IPTABLES -A FORWARD -p tcp -m tcp  -s 192.168.35.3   --dport 53  -m state --state NEW  -j ACCEPT
$IPTABLES -A FORWARD -p udp -m udp  -s 192.168.35.3   --dport 53  -m state --state NEW  -j ACCEPT
#
# Rule 3 (global)
# Allow everyone (internal and external) to visit the web page.
$IPTABLES -A FORWARD -p tcp -m tcp  -m multiport  -d 192.168.35.1   --dports 80,443  -m state --state NEW  -j ACCEPT
#
# Rule 4 (global)
# Allow receiving email from external servers.
$IPTABLES -A FORWARD -p tcp -m tcp  -d 192.168.35.2   --dport 25  -m state --state NEW  -j ACCEPT
#
# Rule 5 (global)
# Allow our mail server to send email out to the world.
$IPTABLES -A FORWARD -p tcp -m tcp  -s 192.168.35.2   --dport 25  -m state --state NEW  -j ACCEPT
#
# Rule 6 (global)
# Allow users to retrieve email via POP3S or IMAPS (both TLS Encrypted to the Mail Server)
# Allow users to send email via SMTPS (TLS Encrypted to the Mail Server)
$IPTABLES -A FORWARD -p tcp -m tcp  -m multiport  -s 192.168.1.0/24   -d 192.168.35.2   --dports 993,995,465  -m state --state NEW  -j ACCEPT
#
# Rule 7 (global)
# Allow the users on Net 1 (client network) to surf the Internet.
$IPTABLES -A FORWARD -p tcp -m tcp  -m multiport  -s 192.168.1.0/24   --dports 80,443  -m state --state NEW  -j ACCEPT
#
# Rule 8 (global)
# Cleanup Rule - Anything not allowed above is denied and logged.
$IPTABLES -N RULE_8
$IPTABLES -A FORWARD  -j RULE_8
$IPTABLES -A RULE_8  -j LOG  --log-level info --log-prefix "RULE 8 -- DENY "
$IPTABLES -A RULE_8  -j DROP
```

You just completed your first ever firewall configuration. Granted, this is a pretty simple firewall, but those rules are fully functional. From here, we could go on to add Network

Address Translation (NAT), VPN configurations, object and rule groups and the list continues.

Congratulations!