CSCI 6542, Fall 2015
Lab 3, Due 10/21/2015

1. Demonstrate mastery of the tools discussed in class by implementing the tests described in the following exercises.

Deliverables for each test are listed below the test description.

1. Create a Windows 7 Virtual Machine and create a minimum of five (5) users. These user accounts should have passwords of varying strengths. For example, one user should have a very simple password of ~3 characters and additional users should have increasingly more difficult passwords.

2. Use Kali or another method of your choice to extract the user accounts and password hashes from the Windows system that you created in the previous step.
    - **Must submit the methods used, detailed commands issued, and resultant password hashes extracted.**

3. The extracted password hashes must then be cracked through use of john, hashcat, rcrack, or other methods of your choice. The password cracking application must be run against the extracted hashes for a **minimum of 96 hours** (recommend using the free Amazon Elastic Compute Cloud), or until all password hashes are cracked.
    - **Must submit a write up of the methods used, detailed commands issued, and resultant password hashes extracted.**

Some useful commands ran in class include; dmesg, mount, samdump2, john, and hashcat.

The rough process for hash extraction using Kali was as follows:
- On the system that has Windows installed, boot from removable media (or ISO of Kali)
- Once the system has booted into Kali, make the Windows partition available for reading/writing
- Identify the specific location of the Windows SAM and system files
- Break the syskey and extract the hashes from the SAM

2. Demonstrate your understanding of the Open Vulnerability and Assessment Language by creating an OVAL definition file that meets the requirements specified below. It must run successfully using the OVAL interpreter obtainable at http://sourceforge.net/projects/ovaldi/

    2.1. Check that Windows is not storing LM hashes of user passwords.
    2.2. Check that the version of regedit.exe is greater than version 4.0
    2.3. Check that the Print Spooler (Spooler) service is set to start value 'automatic'

    - **Must submit your OVAL .xml file and .html output from Ovaldi. Also write a sentence or two about how your OVAL definitions meet the requirements - describe what objects you are checking and why.**

3. Build a simple Splunk deployment or create a Splunk cloud account - the exact mechanism should not matter. Once you have obtained access to a Splunk deployment, import the netflow_example_from_class.csv as a new data set, and then develop simple Splunk queries as demonstrated in class to determine the following statistics:

    3.1. Determine the sum of bytes, and the count of packets for by prevalent source IP, destination IP and source/destination IP pair.
    3.2. Determine the most common third octet for both source and destination IP addresses (hint, remember the 'rex' command from class)
        3.2.1. The third octet is the third number in the IPv4 address, e.g. in 1.2.3.4, 3 is the third octet - more on this in two weeks.
    3.3. Determine the most common last octet for both source and destination IP addresses (hint, remember the 'rex' command from class - be careful what the end of your regex is anchored to)
        3.3.1. The last octet is the last number in the IPv4 address, e.g. in 1.2.3.4, 4 is the last octet - more on this in two weeks.

    - T**he full queries that generate those statistics and screenshots of the results must be included for full credit toward the assignment.**

If you have specific questions, please feel free to ask in class or send Matthew Norris and Kevin Yasuda an email.