SY201 Name(s): _____

Alpha(s): _____

Assignment Type:	Lab	Collaboration Policy:	**Discuss Only**
Assignment Title:	Lab 9 - Patch or Pwn	Submit Project Name:	lab9
Electronic submission due: 2359 5 Dec Submission instructions: http://courses.cyber.usna.edu/SY201/calendar.php?load=policy			

1. Assignment Overview

In this assignment you will create a program, in which you will use classes and objects to represent hosts (e.g. computers) on a network. As with any hosts in the world, the systems may have known vulnerabilities, and for each vulnerability there may be a patch or an exploit. In the final portion of the lab you will run a series of actions and determine what hosts have been compromised and what hosts don't show signs of compromise.

2. Background

a. Data files that your program will read in (File IO) will be in a format called CSV (Comma Separated Value) where each element on a line is separated by commas (no spaces). This format is commonly used to read/write data from programs. It is also a format that can be edited via convenient editors (e.g. Excel).
   i. Most simply: **`firstField,field2,field3,lastField`**
   ii. If a field is intended to be a list, the list items will be separated by a colon (**`:`**)
   iii. CSV with a list: **`prevField,firstItem:listItem2:lastItem,nextField`**
   iv. Note: if there is a single item in the list, then there will be no colon (no list items to separate)
   v. For data that are name value pairs, the name value pair will be separated by an equals sign (**`=`**) with the name to the left of the equals sign
   vi. CSV with name value pairs: **`prevField,name1=value1,nextField,nameA=valueA:nameB=valueB`**

b. Remember: the exception raised when trying to open a file for reading (file I/O) and the file does not exist is a **`FileNotFoundError.`**

3. Specification.

Don't forget:
- Write your alpha and section number in a comment at the top of your program
- Identify anyone you discuss the assignment with in a comment at the top of your program
- Include comments adjacent to each function you write that describe the function's purpose
- Your program should print out nothing additional to what is called out below

SY201 Name(s): _____

Alpha(s): _____

Part One - create a file lab9-part1.py and implement the following functionality:

A. If your program should run from the command line as follows (take two required arguments via command line, where the second argument is the name of a text file):
   a. **`python3 lab9-part1.py -h hostSetX.txt`**
   b. **`-h`** -> load hosts
   c. Where **`hostSetX.txt`** is a CSV formatted file of hosts on a network
   d. Each line of **`hostSetX.txt`** contains five comma separated values: (1) hostname (2) IP address (3) OS family (4) OS version (5) installed software list
   e. The installed software list will be a list of name value pairs, where the name is the name of the software and the value is the installed version of the software

B. Create a Host class to represent a host on the network
   a. You may use attributes (data and methods) as you see fit
   b. You will build on your Host class in Part Two and Part Three

C. For each line in the **`hostSetX.txt`** file, where **`X`** can be any number of values specified in the command line input, create an instance of your Host class that incorporates the proper data from the file

D. Print out each of your Host objects alphabetically by host name

E. Each Host object should be printed out in the following manner:
   a. name IP Address OS_Family-OS_Version
   b. e.g. **`yog 10.1.83.30 Linux-Ubuntu 14.04`**

Part Two - create a file lab9-part2.py that contains <u>everything</u> from Part One and implement the following <u>additional</u> functionality:

A. If your program should run from the command line as follows (take three required arguments via command line, where the arguments are the flag and the names of text files):
   a. **`python3 lab9-part2.py -v hostSetX.txt vulnSetY.txt`**
   b. Where **`hostSetX.txt`** is a CSV formatted file of hosts on a network formatted as described in Part One
   c. Each line of **`vulnSetY.txt`** contains four comma separated values: (1) CVE ID (2) list of vulnerable platforms (3) list of vulnerable software (4) list of available patches
   d. Note: extra reading about CVEs: https://cve.mitre.org/about/
   e. The vulnerable platforms list will be a list of name value pairs, where the name is the OS Family (match to Host OS Family), and the value is the OS Version (match to Host OS Version)

f. The vulnerable software list will be a list of name value pairs, where the name is the software name (match to Host Software Name), and the value is the software version (match to Host Software Version)
g. The patch list will be a list of name value triples (separated by equal sign), the name will be the software name, the first value will be the vulnerable version, and the second value will be the version to patch the software to when applying a patch (Note: the patch list is only applicable in Part Three)

B. Print out each of your Host objects alphabetically by host name that has at least one vulnerability identified in the **`vulnSetY.txt`** file

C. Each Host object should be printed out in the following manner:
   a. name IP Address OS_Family-OS_Version
   b. e.g. **`yog 10.1.83.30 Linux-Ubuntu 14.04`**

Part Three - create a file lab9-part3.py that contains <u>everything</u> from Part Two and implement the following <u>additional</u> functionality:

A. If your program should run from the command line as follows (take four required arguments via command line, where the arguments are the flag and the names of text files):

```
python3 lab9-part3.py -g hostSetX.txt vulnSetY.txt gameSetZ.txt
```

1) Where **`hostSetX.txt`** is a CSV formatted file of hosts on a network formatted as described in Part One
2) Each line of **`vulnSetY.txt`** contains four comma separated values formatted as described in Part Two
3) Each line of **`gameSetZ.txt`** contains three comma separated values: (1) target host (either hostname or IP address) (2) CVE ID (3) action
4) Your program should process each line of **`gameSetZ.txt`** as an action to take sequentially. If the action is **`exploit`** and the target host is vulnerable to that CVE, then it is now exploited. If the action is **`patch`** and the target host is vulnerable to that CVE, then it should be patched (software version upgraded).
5) Hint: add a **`patch`** method and an **`exploit`** method to your Host class
6) Play out each set of actions in the game file (**`gameSetZ.txt`**) in order:
   i. All Hosts start out in a not exploited state
   ii. Once a vulnerable host is properly exploited, it remains exploited, even if it gets patched later
   iii. Hint: you may want to make more methods and data attributes in your Host class

B. Print out each of your Host objects alphabetically by host name that are exploited after each of the actions in the game file (**`gameSetZ.txt`**) are taken

C. Each Host object should be printed out in the following manner:

c. name IP Address OS_Family-OS_Version

d. e.g. **`yog 10.1.83.30 Linux-Ubuntu 14.04`**

Submit all three files as so:

```
submit -p lab9 lab9-part1.py lab9-part2.py lab9-part3.py
```