Combatting Challenges in Cyber Security Education

Arati Banerjee
Department of Computer Science
University of Central Florida
banerjee@knights.ucf.edu

Salih Safa Bacanli
Department of Computer Science
University of Central Florida
sbacanli@cs.ucf.edu

Dr. Cliff Zou & Dr. Damla Turgut
Department of Computer Science
University of Central Florida
{czou, turgut}@cs.ucf.edu

ABSTRACT

Currently, cybersecurity education relies on a limited toolset that does not accurately reflect the reality of the challenges in the field. A virtual lab that had a finer grain of secure operating systems much better echoes the ever-changing levels of threats and security in the world. Currently, security educators are only able to run penetration testing on systems with a few levels of security. With Windows XP, for example, which Microsoft has only recently stopped support for, we can only simulate vulnerabilities before and after service packs 2 and 3. This inevitably misrepresents the gradual change of security over time.

In this project, we sought to create a tool able to produce virtual machines simulating different points in the Windows operating system life cycle when major security patches were implemented. The tool used features already provided by the operating system to remove security updates and leave the system vulnerable to attacks that later updates were able to stop.

MOTIVATION

- In 2013, across 24 countries, the cost of cybercrime was estimated to be about $144 billion USD [1].
  - The need for cybersecurity professionals is increasing [2].
  - The need for better, more robust tools for educating cybersecurity students is increasing.

- Learning how to create secure systems is key, and cannot be done without learning how to exploit vulnerabilities.
  - There are a vast number of tools available to run penetration testing - a controlled attack to test the security of a system.
  - But what about systems to experimentally exploit?

- Educators use virtual machines to teach exploitation, as the main vulnerability in a system is generally its own operating system [3].
  - The security of an operating system changes with each update.
  - It would better imitate the evolution of security if there was a tool to roll back a virtual machine to states before security patches and follow the timeline of an active operating system.

The **objective** of this project was to create a simple, adaptable tool that would be able to revert Windows virtual machines back to vulnerable states for penetration testing in cybersecurity education.

IMPLEMENTATION

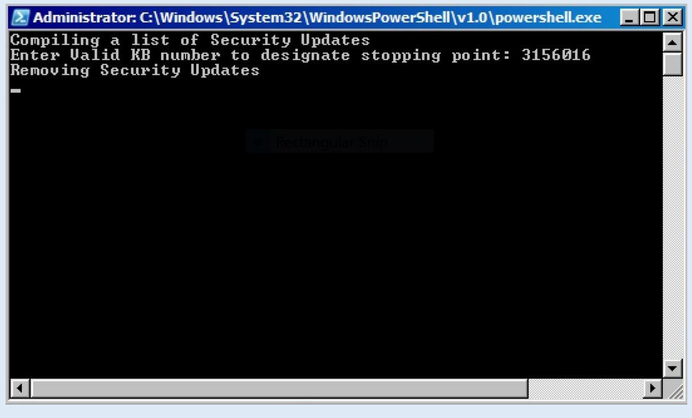
The script created for this project creates the temporary files (final.txt, next.txt), asks for input of KB (Knowledge Base) numbers – which identifies the updates chronologically, and then removes the updates to the given point without requiring further user interaction.

Part of the prototype PowerShell script:

```
1  echo "Compiling a list of Security Updates"
2
3  wmic qfe get | where {$_-match "Security"} | %{$_.split('[= ]')[1]} > final.txt
4
5  $file = gc "final.txt"
6
7  if ($file.length -gt 4){ [array]::reverse($file) }
8
9  $file > next.txt
10
11 Remove-Item final.txt
12
13 $KBNum = Read-Host -Prompt "Enter Valid KB number to designate stopping point"
14
15 echo "Removing Security Updates"
16
17 foreach ($line in Get-Content .\next.txt)
18 {
19    cmd /c "wusa /quiet /uninstall /kb:$line /norestart"
20    if ($line -eq $KBNum) { break }
21 }
22
23 Remove-Item next.txt
```

- Line 3 isolates the KB numbers of security update and stores them in a temporary file
- Line 7 reverses the list
  - takes care of any dependence - KB numbers are assigned in increasing, chronological order
- Lines 17-21 removes updates using the command prompt
  - if a valid KB number is provided at the prompt, the script will stop at the defined number – otherwise it will remove all updates

The running script:



This script is

- intuitive to use, with clear prompts and progress updates at each step
- robust enough to handle all provided Windows 7 and 8.1 virtual machines, and possibly more
- functional on all Windows systems in which the *wmic* and *wusa* commands are available
  - Windows 7 and Windows 8.1
  - Possibly Windows 10 as well

This tool fulfills the objective of this project – to ease the way for cybersecurity educators and students to run penetration testing on Windows virtual machines.

FUTURE WORK

Future work may include implementing a way to include commonly-tested vulnerabilities for pre-set options for experienced cybersecurity educators.

Additional work may include testing on Windows 10 and other systems – this tool has been successfully tested on Windows 7 and 8.1 systems provided at the Microsoft website for developers - https://developer.microsoft.com/en-us/microsoft-edge/tools/vms/

REFERENCES

[1] A. McGettrick, "Toward effective cybersecurity education," *IEEE Security Privacy*, vol. 11, no. 6, pp. 66–68, Nov 2013.

[2] W. A. Conklin, R. E. Cline, and T. Roosa, "Re-engineering cybersecurity education in the us: An analysis of the critical factors," in *2014 47th Hawaii International Conference on System Sciences*, Jan 2014, pp. 2006–2014.

[3] M. Denis, C. Zena, and T. Hayajneh, "Penetration testing: Concepts, attack methods, and defense strategies," in *2016 IEEE Long Island Systems, Applications and Technology Conference* (LISAT), April 2016, pp. 1–6.

ACKNOWLEDGMENT

The support for this work was provided by the National Science Foundation REU program under Award No. 1560302. Any opinions, findings, and conclusions and recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.