HACS408T Penetration Testing **Spring 2019**

Homework 9 - Powershell Toolkit

Introduction

Your task is to write an all-powershell offensive toolkit to conduct an objective on a foreign network: locating a sensitive file, backdooring the target computer, and exfiltrating the file off the computer to a server you control. You have gained a foothold on the network, but your account is unprivileged and cannot log in to any other machines on the domain. As a result, you must brute force the logins of other accounts on the network until you can log in to other machines the domain, find the file, and exfiltrate it over TCP. You should log all successful logins. Once you've gained access to a machine, you should set the registry key `HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Policies\System\LocalAccountTokenFilterPolicy` (or create it if it does not exist) and set it's value to 1. This will serve as our backdoor in the future.

Additionally, your implant needs to take into account the lockout threshold: group policy will have an account lockout and appropriate observation window set. If your implant finds account that is locked out, move on to the next user in your list and attempt to use the locked-out account after the threshold has passed. We will import your script as a module and run the function `BeginImplant`. It is your responsibility to create that function.

Your program needs to take the following arguments:

- PassList: File containing a newline-delineated list of passwords
- FileName: The name of the file to locate
- Target: The IP:Port combination of a listening server, waiting for the file
- LogFile: File to log any successful username/password combinations
    - If your program is not run with this parameter, simply log to the screen rather than a file.

After parsing the above arguments, you should then:

- Find all computers on the domain
- Attempt to brute force the logins of all user/password combinations given by the username and password list, logging those to the LogFile.
- Attempt to find a file matching the Filename argument on any system where you can log in with the credentials given by your brute force attempt
- Send the file over TCP to the waiting server

Requirements

- Implant must be written entirely in Powershell.
- Implant must be self-contained.

- You are not permitted to use any outside libraries (or copy code the Internet). You are allowed (and encouraged) to break the problem down and research what is required to solve each component, but you cannot copy/paste code.
- You must create the above registry key if it does not exist, and if it does exist, it's value must be set to 1.

Important Notes

You may assume that:

- All computers on the network are running Windows Server 2016 with the Active Directory Users and Computers role installed
    - In practice, this would not happen, but we want to limit the number of version issues
- WinRM is working on all machines with the correct firewall rules added
- Each computer is correctly on the domain, including yours
- The target file is actually on one computer on the domain
- At least one username/password combination can log in to a computer with the proper credentials to find and open the file
    - That is, there is a solution every time
- There is only one forest on the domain

Testing

You will be tested on an unknown environment. It is also up to you to create a test environment for this. We recommend creating a network with three computers, all Windows 2016 (GUI):

- A Domain Controller
- An attacker
- A target

You will need to install the correct roles for each by using the server manager. On the Domain Controller, make sure to run promote it to a Domain Controller after installing Active Directory. On the clients, make sure to set the DNS server to the Domain controller before adding that machine to the domain.
To add users to the Domain, use the Domain Controller's Active Directory Users and Computers plugin.

Your implant will be tested with the following commands:

```
Import-Module .\implant_kbock.ps1
BeginImplant -PassList <name_of_file> -FileName <name_of_file>
-Target <IP>:<Port> -LogFile <name_of_file>
```

As usual, please write to the screen if it is helpful (but don't go so crazy we don't know what's going on!)

Extra Credit

For extra credit, multithread your brute force logic so that each user is brute forced in parallel. If you do so, include a `report.pdf` and state that you have done so with how you did it.

Submission

Please submit your code in a .zip file to the ELMS submit link. Your code should be in one `.ps1` file named `implant_<directory_id>.ps1`.