|  |  |  |
| --- | --- | --- |
| Course # | James Riley Dorough | Covertly Capturing and Using Hashes Lab Report |
| CSIS 462 | **01APR22** | **Semester Week 13, Lab 8** |

Title

* Collect, exfiltrate, and implement blue team hashes in an attack against their systems

Introduction

In this lab I use Mimikatz to collect hashes on blue team systems. I then work with various tools to decide how to use the exfiltrated hashes. I end up using hashcat to show how hashes can be cracked. I passed up on a pass the hash attack as I heard other students would be doing that and decided to learn a different tool. Once the password is cracked using a simple dictionary attack or a hybrid attack using a rule set. You can continue to attack a target range using the newly gained credentials.

Method

* The first step is to install Mimikatz on the target system
  + This means you either need a low-level account on the target system, an outdated and exploitable service running for RCE, or some form of implant or C2 running

Text

Description automatically generated

Text

Description automatically generatedText

Description automatically generated

* Once you have your hashes (I used the NTLM hashes) you need to exfiltrate them to a system where you can operate with them
* For longer hashes or keys, you will need to transfer files instead of simply copy (covertly) the short hashes used in this report
* Once the hashes are on your system, you can either use them directly in a pass the hash attack or attempt to crack them to retrieve the password
* Using the Hashcat tool you can use a variety of wordlists, dictionaries, rulesets, lists of encryption algorithms, and various modes of operation
* I first combined the hashes I collected into a single file to pass as an argument

Text

Description automatically generated

* The command I ran was “ Hashcat -m 1000 -a 0 2-3.NTLM.Hashes /rockyou.txt /rockyou-3000.rules –show
  + The -m 1000 tells Hashcat that you are attempting to break a Windows NTLM hash
  + The -a 0 specifies that you want to use a simple dictionary attack
  + 2-3.NTLM.Hashes is pretty self-explanatory, these are your hashes to crack
  + Next you specify the dictionary you want to use to test for passwords; I used the rockyou.txt password list locate downstream from /etc
  + Lastly, Hashcat’s home directory contains sets of rules you attempt various modes of cracking passwords by appending and inserting extra symbols, characters, or splicing together different sets of passwords
* Running the command presents a live view of the attempts being made to crack the hash with information about the process

Text

Description automatically generated

* Hashcat is weird in the sense that it won’t let you run an attack against a hash you have already cracked
* You can use the --show flag to show the results of an already cracked hash
  + I have not discovered a way to clear out the storage where Hashcat keeps the already cracked hashes to show them completing normally



* The importance of using dictionaries and special rule sets or knowing the makeup of what types of characters may be found in the password is critical to have a successful attack

Table

Description automatically generated

* This image shows benchmarks of using AWS cloud computing to crack passwords of various length and complexity. This is using computers with guaranteed power supply and run time with a speed of 632 GH/s! No VM running on a student’s laptop can directly brute force an 8 character password if it includes letters and symbols. You need to know as much about the requirements of passwords or possible nomenclature your target may use, such as including a rule to append or insert the target’s birth year in the cracking attempts

Conclusion

This lab focused on a different side of penetration. Instead of just using vulnerabilities to execute code on a system. This lab builds on persistence by providing the attackers valid credentials of users on blue systems. If you can extract admin credentials from a system covertly as a low-level user, you can study the access habits of the privileged user to know when the most reasonable times to operate to remain hidden are. You can make changes to the system carefully and extract useful information without drawing attention to your actions.