Updated: Sep 27, 2022

Lab 5.1 - Password Guessing

💡Last week, during an exploit demonstration, we performed a hydra attack against the samwise account on cupcake. This week, we will extend this sort of informed guessing attack against another target. Trying to acquire easy to guess usernames/passwords is often the first step in the exploitation stage. This is different from brute-force attempts which can be very processor-intensive, take a very long time and may not yield results.

Hints

If you need them, they will appear at the end of this lab

Scenario

10.0.5.21 has a listing of bios that can in turn be used in information gathering. If successful you can gain access to hidden resources and possibly a login or two.

NSLookup

Remember, that trick of using a specific DNS server (the one on shire), to lookup a hostname or reverse lookup an IP address? The host we are interested in has the IP of 10.0.5.21. Use nslookup or another comment to determine the actual hostname of the system as provided by DNS.

`Deliverable 1. Provide a screenshot that shows the lookup and reported hostname.`

Password Guessing

Running some quicker tests to try and uncover a few accounts that may be using easy-to-guess passwords may provide a simple initial foothold. We will come back to other password-cracking techniques later in the exploit stage - as many of them are more beneficial once you have some access to the systems.

Class Lab: Hacking the Shire Staff

Wordlists and Dictionaries

Updated: Sep 27, 2022

Password-guessing tools use wordlists/dictionaries to test combinations of usernames and passwords. Many files are available to download but crawling websites is a useful way to gather words/usernames that may be uniquely associated with a particular organization.

cewl

The Custom Wordlist Generator(cewl) included in Kali, crawls websites and creates a custom wordlist. Spend some time reading the man page. You will need to play with the depth flag to single out a member's biography. Cewl has some issues and we will work through them as they appear.

Use cewl to create a custom wordlist for the shire bio pages of Frodo, Pippin, Bilbo and Samwise from http://10.0.5.21/bios. Make sure you crawl the full page as opposed to the shortened version.

In the interest of time, we will be making a small list. To do so:

- Pay attention to unique/interesting words
- Passwords will come from Proper Nouns on the bio pages
- Create a separate wordlist for each staff member member
- Create a smaller variant of the original where you've trimmed out words that will not likely be a password. (The, and, is, from….)

Mangling the Wordlist

Making variants of your list of plaintext words is useful. Cagey users want an easy to remember password but they also want to obfuscate it just a bit to feel secure.

rsmangler

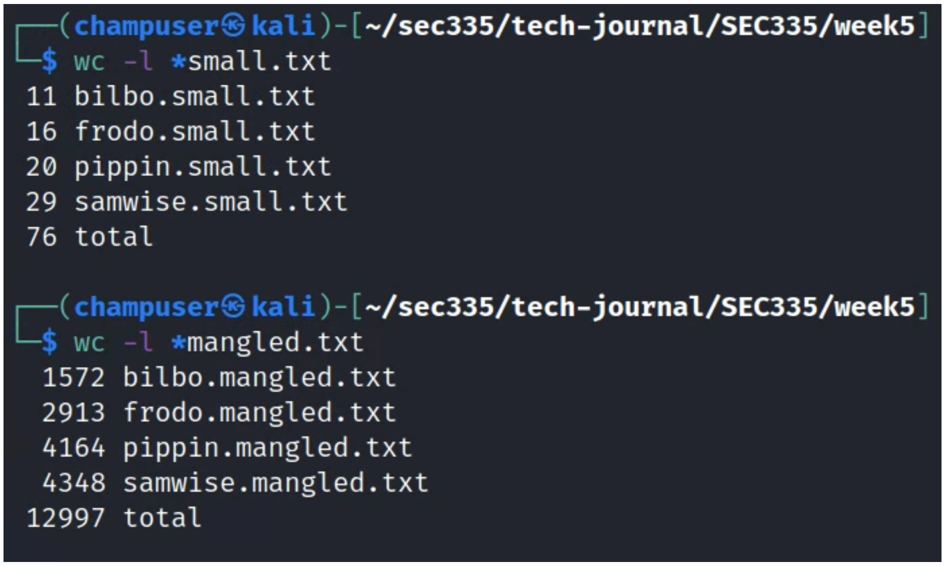
rsmangler is a tool that takes wordlists and "mangles" them by adding, leet speak, numbers, years, mix case, special characters and various other flags.

Read the man page for rsmangler: Note that switches disable certain "mangles"
Select the 4-5 most unique words for each of the target users (so, create 4 lists total)
Run rsmangler on each list - in the interest of time

- password sizes will range from 9-12 characters (use min/max)
- Ideally, you should have a list under 1,500 entries
- The linux "wc" command (aka word count) can show you how many lines are in a file
- Record the command you used

Here's the instructor's source and mangled wordlist counts, you want to keep yours of similar size or smaller.

Updated: Sep 27, 2022



```
┌──(champuser㉿kali)-[~/sec335/tech-journal/SEC335/week5]
└─$ wc -l *small.txt
11 bilbo.small.txt
16 frodo.small.txt
20 pippin.small.txt
29 samwise.small.txt
76 total

┌──(champuser㉿kali)-[~/sec335/tech-journal/SEC335/week5]
└─$ wc -l *mangled.txt
  1572 bilbo.mangled.txt
  2913 frodo.mangled.txt
  4164 pippin.mangled.txt
  4348 samwise.mangled.txt
 12997 total
```

- note: the list to be mangled needs to have a newline at the bottom of it. When you look at it in nano, make sure there is a line after the last word.

Password Guessing

Target Server: 10.0.5.21

```
Deliverable 2.  Using what you have during the reconnaissance
modules, run a scan to determine any listening tcp services to
include the service versions.  Provide a screenshot of both your
command and results.

Deliverable 3.  Let's see if we can get a little more information on
your website such as hidden directories.  Research the dirb command
and run it against the webserver.  Turn off recursion.   Provide a
screenshot of anything secret you've found.

Deliverable 4.  Provide a screenshot that displays a prompt when
attempting to access a protected directory.
```

💡Kali has multiple tools to script password guessing logins. Medusa, ncrack and hydra are all options. Take a few minutes to assess all three tools and pick one. You've already used hydra so feel free to try another tool, though hydra should work for this lab.

Updated: Sep 27, 2022

Cracking the http password

The usernames for our http password protected case will be just the first names for our characters. Such as samwise, pippen, bilbo and frodo.

Start with one of the group member's output lists from rsmangler to try and guess passwords on the target server.

💣hydra will give you a false positive if you fail to terminate the protected directory with a /
Make sure to test positives with a browser navigating to the the actual site.

Repeat to get at least 3 out of 4 of the listed staff!
Deliverable 5. For at least 3 shire staff, bypass authentication on the protected directory using the tool of your choice. Provide screenshot(s) showing the tool execution and the guessed password. (Make sure you validate all of these work)

Brute Forcing System Accounts

Brute-forcing web logins can be much faster than some other services. SSH, for example, will often tear-down sessions after 3 failed logins which can slow down password guessing attacks.

The target server also has SSH login accounts for the same staff members though their names are formatted first.lastname. (pippin is a nickname (not a first name) and wouldn't be used)

Deliverable 6. Submit screenshots of your tool of choice reporting successful ssh password guesses of at least 3 member's linux accounts.

Deliverable 7 - Journal. Create a content page on password guessing and cracking. Provide a link to the page. Make sure to include

- Notes on using CEWL
- Notes on using rsmanglier with parameters
- make note of any issues and how they were resolved
- Notes on medusa, hydra, ncrack - whatever you used

💡Take great notes on these techniques as well as those user credentials, they will likely crop up on an targets later in the semester.

Updated: Sep 27, 2022

Deliverable 8 - Reflection. This is a good time to reflect on your own passwords. Answer the following and provide a link (if your wiki is public, just answer in the submission rather than posting your weaknesses to the internet).

- Are your own passwords guessable?
- Are they repeated over multiple systems and services?
- Are they included in lists such as rockyou?
- How can you improve your password tradecraft?
- What are you doing right?

Hints on the next page (you need to ask for them, but here are a few to start)

Updated: Sep 27, 2022

- cewl

```
cewl -d 1 http://10.0.5.21/bios/frodo -w frodo.txt
```

- rsmangler

```
rsmangler --file samwise.small.txt -x 12 -m 9  -l -s -e -i -p -u  -a
--output samwise.mangled.txt
```

- hydra (http-auth)

```
hydra -l pippin -P pippin.mangled.txt -s 80 -f 10.0.5.21 http-get /admin/
```

Fall 22 Hints
Samwise's http password starts with an 'R'
Bilbo's http password also starts with an 'R'
bilbo.baggins ssh password starts with an 'F'
frodo.baggins ssh password starts with an 'S'
samwise.gamgee ssh password starts with an 'M'