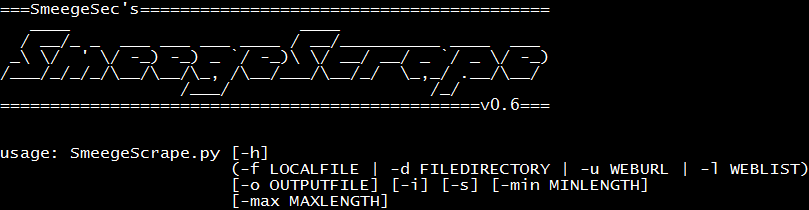
SmeegeScrape.py

Text Scraper and Word List Generator

SmeegeScrape.py is a simple python script to scrape text from various sources including local files and web pages, and turn the text into a custom word list. A customized word list has many uses, from web application testing to password cracking, having a specific set of words to use against a target can increase efficiency and effectiveness during a penetration test. I realize there are other text scrapers publicly available however I feel this script is simple, efficient, and specific enough to warrant its own release. This script is able to read almost any file which has cleartext in it that python can open. I have also included support for file formats such as pdf, html, docx, and pptx.



**Usage:**

One of the following input types is required: Local File (-f filename), Directory (-d directory), URL (-u url), URL List (-l list)

|  |  |
| --- | --- |
| -h, --help | Show this help message and exit |
| -f LOCALFILE, --localFile LOCALFILE | Specify a local file to scrape |
| -d DIRECTORY, --fileDirectory DIRECTORY | Specify a directory to scrape the inside files |
| -u URL, --webUrl URL | Specify a url to scrape page content (correct format: http(s)://smeegesec.com) |
| -l URL\_LIST\_FILE, --webList URL\_LIST\_FILE | Specify a text file with a list of URLs to scrape (separated by newline) |
| -o FILENAME, --outputFile FILENAME | Specify output filename (default: smeegescrape\_out.txt) |
| -i, --integers | Remove integers [0-9] from all output |
| -s, --specials | Remove special characters from all output |
| -min # | Specify the minimum length for all words in output |
| -max # | Specify the maximum length for all words in output |

**Examples:**

Using the following test file (“Test-File.txt”) as input:

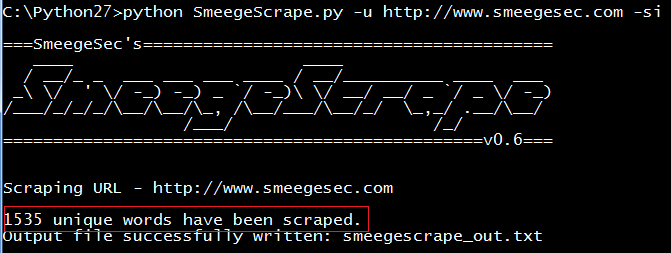
This is a sample text file with different text.

This file could be many different filetypes including html, pdf, powerpoint, docx, etc. Anything which can be read in as cleartext can be scraped.

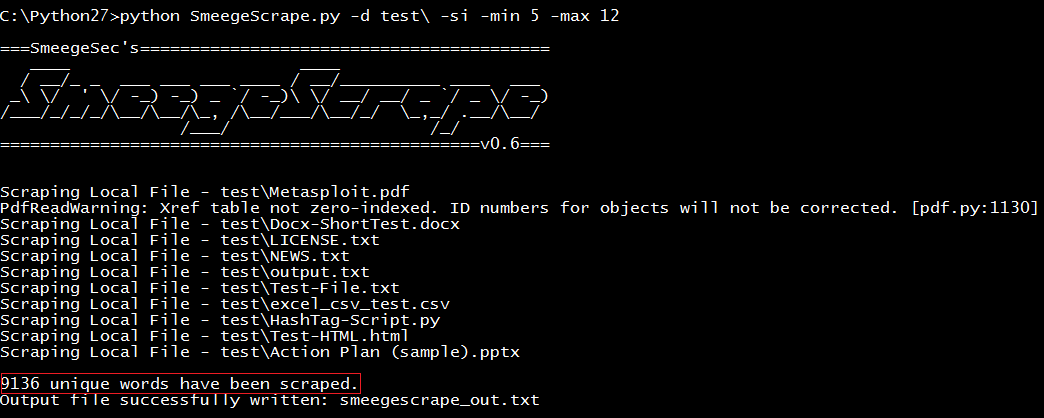
I hope you enjoy SmeegeScrape, feel free to comment if you like it!

A simple example of this scraper would be the command **SmeegeScrape.py -f Test-File.txt** which produces an output text file with each found word separated by a newline. The options -i and -s can be used to remove any integers or special characters found. Also, the -min and -max arguments can be used to specify desired word length.

SmeegeScrape.py can also scrape web pages using the python urllib2 module. An example command would be **SmeegeScrape.py -u** [**http://www.smeegesec.com**](http://www.smeegesec.com) **-si.** The format of the url is checked via regex and it must be in the correct format (e.g. http(s)://smeegesec.com)

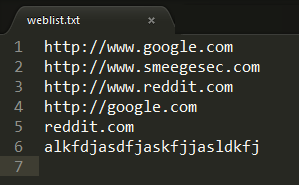


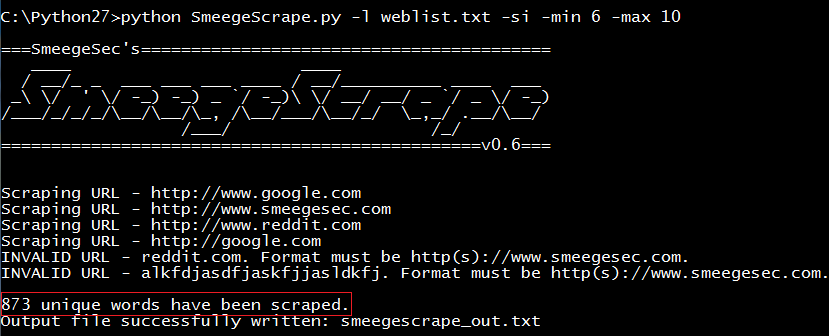
Scraping multiple local files inside of a single directory is also possible via the **-d** option. Running **SmeegeScrape.py -d test\_dir\files\_dir\ -si -min 5 -max 12**:



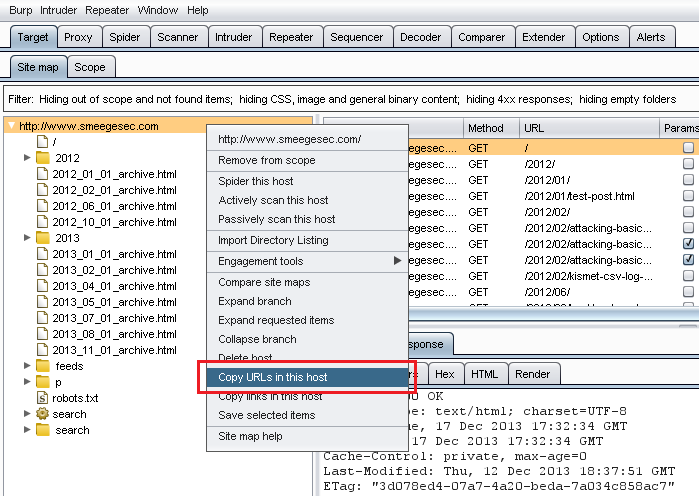
The screen output shows each file which was scraped, the total number of unique words found based on the user’s desired options, and the output filename.

Lastly, the -l option takes in a list of web urls from a text file and scrapes each url. Assuming we have weblist.txt as our input file we could run the following command: **SmeegeScrape.py -l weblist.txt -si -min 6 -max 10**

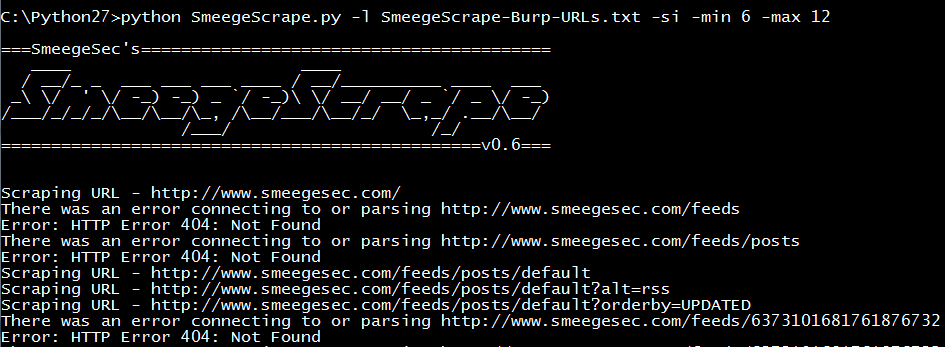


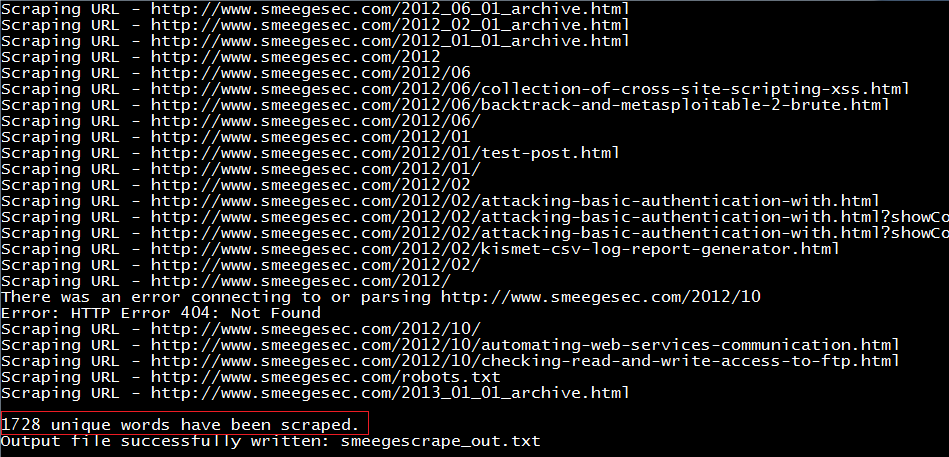


This weblist option is excellent to use with [Burp Suite](http://portswigger.net/burp/) to scrape an entire site. Firstly, discover as much content on the target site as you can (spidering, manual discovery, dictionary attack on directories/files, etc.) . After the discovery phase, right click on the target in the site map and select the option “Copy URLs in this host” from the drop down list. In this instance for even a small blog like mine over 300 URLs were copied. Depending on the size of the site the scraping could take a little while, be patient!



Paste the URLs into a text file and run that as input. **SmeegeScrape.py -l SmeegeScrape-Burp-URLs.txt -si -min 6 -max 12**:



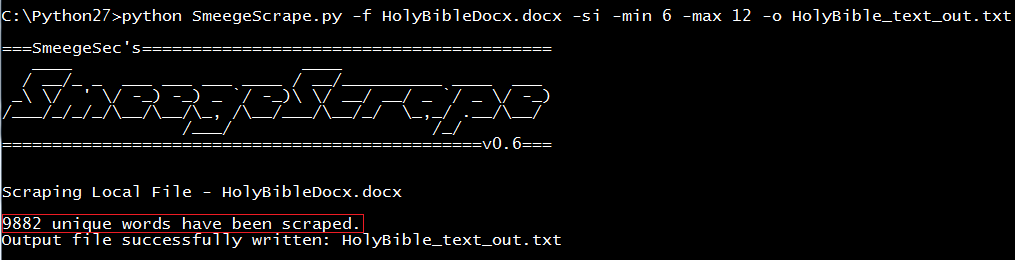


So very easily we just scraped an entire site for words with specific attributes (length and character set) that we want.

As you can see there are many different possibilities with this script. I tried to make it as accurate as possible however sometimes the script depends on modules such as nltk, docx, etc. which may not always work correctly. In situations like this where the script is unable to read a certain file format, I would suggest trying to convert it to a more readable file type or copy/paste the text to a text file which can always be scraped.

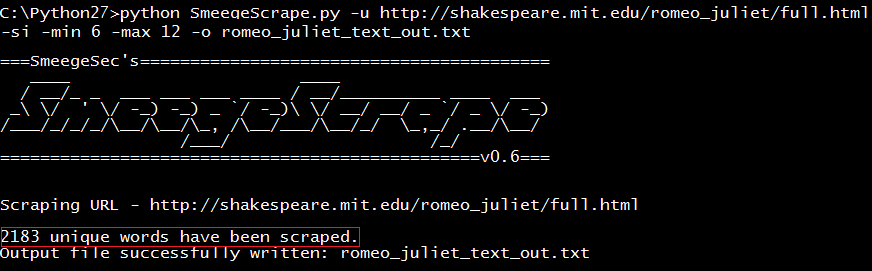
The custom word list dictionaries you create are up to your imagination so have fun with it! This script could also be easily modified to extract phrases or sentences which could be used with password cracking passphrases. **Here are a couple examples I made:**

Holy Bible - King James Version of 1611:



Output examples: Joshua,Loammi,shamefully,allure,Ruhamah,poplars,Whoredom,slideth,spouses,reject,committest,rebuker,profound,howled,testifieth,raising,spiritual,cornfloor,miscarrying,hemlock,reaped,repentings,similitudes,chimney,repentance

Shakespeare’s Romeo and Juliet:



Output examples: Juliet,Entire,Shakespeare,homepage,households,dignity,Verona,ancient,grudge,mutiny,unclean,starcross,lovers,piteous

Comments and suggestions always welcome - enjoy!