Cyber_suite

Motivation/Introduction

I got really interested in hacking and ctfs after the Zense team sent us a mail way back in 1st sem , talking about bandit(https://overthewire.org/wargames/) and ctfs in general. I tried to do them when I got some free time. They were hard ,to be honest, but in the process I noticed that there were similarities and mundane tasks, in challenges that could possibly be automated. So I thought I would build a suite of tools that would help in these hacking challenges like convertion from one form to another, encryption and decryption of basic ciphers and maybe try and detect vulnerabilities in code. Now I know that there are some amazing tools(mostly free!) which do these tasks super efficiently and there is no way that a 1st year cse student who has just started out with hacking and making large projects could match those projects. But I really enjoy these challenges and I thought building tools which might help in them, would be fun. And it was!

The tools I was finally able to accomplish was not as vast I had imagined but I think they are decent:

1. Converter: Converts between hex,decimal,binary and ascii.
2. Encrypter: Given a file or string it encrypts the data using the specified cipher(Ceasar,vignere with specified keys)
3. Decryptor: Given an encrypted file and type of encryption(ceasar,vignere,monoalphabetic random substitution) it decrypts the data to the best possible extent.
4. Static Code Analysis: Given python files it outputs lines which might be vulnerable to sql injection and command line injection.

Modules used

1. sys: For command line arguements
2. ast: Abstract Syntax Tree used for static code analysis

- No special installation of modules required

Usage

Converter

python3 converter.py conversion value_to_be_converted

Eg: `python3 converter.py hex-dec ff`
conversion arguement is of the form datatype1-datatype2(asc,hex,dec,bin)

Eg: hex-asc, dec-bin, asc-bin.... All 12 combination are allowed

Note: Hex values cannot have upper case letters. So A3 is not allowed. Instead use a3.

Encrypter

python3 encrypt.py option cipher key data

Eg: `python3 encrypt.py -s rot 13 sooraj python3 encrypt.py -f vignere gold testfiles/test2.txt`
option: -s or -f to input direct string or input file contaning text

cipher: rot(caesar) or vignere

key: For rot it has to be a number and for vignere it has to be a small word.

https://www.geeksforgeeks.org/vigenere-cipher/

https://en.wikipedia.org/wiki/Caesar_cipher

data : if -s then a string has to be given. If -f then name of file has to be given

Decrypter

Broadly speaking there are 2 types of decryption here: where key is known and where key is not known. In the latter case we will try to guess using some user based input and some brute forcing. Examples will hopefully make it clearer.

python3 decrypt.py option cipher data

`python3 decrypt.py -sk rot 13 fbbenw`
option:

1. -s : if key is unknown and data is a string given directly
2. -sk: key is known and data is a string given directly. for rot key is a number and vignere it is a word/string

3. -sK: specifically for vignere where only keyword length is known but key itself is unknown. Works best for small keywords and large data. Use -fK instead.

4. Similar where input is a file except change s to f. So -f,-fk,-fK.

cipher:

1. rot: Caesar cipher. If key is unknown then all 26 possible rotations are tried and displayed. Left to user to take the text that makes sense.

```
python3 decrypt.py -s rot fbbenw
```

2. monoalph: Monoalphabetic substitution cipher where each letter is mpped to some other random letter. There is no standard offset like in caesar cipher. For this frequency analysis is used . This needs a bitmore interaction and attention from user using their knowledge of English and using some context.Lets take an example...

In test2.txt there is some text I had written in the start of sem1. It is composed of only English alphabets. test7.txt has this text but in encrypted form.

```
python3 decrypt.py -f monoalph testfiles/test7.txt
```

We first see the original file contents and then *s and then the first iteration of decrypted text, This still seems weird but upon closer inspection we can see the word 'tre' and can make an educated guess that it has to be 'the' . So we then change the mapping of 'g' which initially mapped to 'r' to now map to 'h'.

```
Mapping used:

 {'s': 'e', 'q': 't', 'b': 'a', 'n': 'o', 'u': 'i', 'c': 'n', 'j': 's', 'd': 'h', 'g': 'r', 'w': 'd', 'z': 'l', 'm': 'c'
, 'v': 'u', 'y': 'm', 't': 'w', 'x': 'f', 'e': 'g', 'a': 'y', 'k': 'p', 'l': 'b', 'f': 'v', 'i': 'k', 'h': 'j', 'p': 'x'
, 'r': 'q', 'o': 'z'}
Would you like to change the mapping(y/n): y
enter letter whose mapping u want to change: g
new mapping of previously mentioned letter: h
```

We then see the second iteration of decrypted
text.

```
lce ti the diriosvarcn bsolemad mint aontatctaion hsve pioe vartcsu sol everg ntcleot’n uesroaop an iouaoe.
 mg ejberaeode hsn yeeo oi laffereot suthicph a ntsrtel stteolaop iouaoe dusnnen ioug verg redeotug.
 mg farnt few iouaoe nennaion were the iraeotstaio nennaion yefire slmannaion diolcdtel yg aaat h sol y io xiim.
 thsokfcuug mg nanter hsl sureslg yeeo cnaop xiim fir her dusnnen naode msg sol thcn nhe dicul heub me osvapste at effedtaveug.
  ioe snbedt a ficol qcate ncrbranaop wsn the bcodtcsuatg ejhayatel yg yith aontatctaion wheo ntsrtaop the nennaion.
```

We see word 'the' appearing correctly. Now there are more words that are wrong by just 1 letter like 'hsve'->'have' , 'everg':'every', etc. Changing the mapping of 'j' to 'a' instead of 'j' mapping to 's' the text becomes even more understandable.

Keep proceeding in this way to decrypt it fully.

After a few iterations of this we get this:

```
dce to the uoronavircs bandemiu most institctions have pone virtcal and every stcdent’s learninp is online.
 my ejberienue has geen no different althocph i started attendinp online ulasses only very reuently.
 my first few online sessions were the orientation sessions gefore admissions uondcuted gy iiit h and g on xoom.
 thankfclly my sister had already geen csinp xoom for her ulasses sinue may and thcs she uocld helb me navipate it effeutively.
  one asbeut i focnd qcite scrbrisinp was the bcnutcality ejhigited gy goth institctions when startinp the sessions.
i was very imbressed with goth sessions gct i deuided to zoin iiitg in the end.
uominp to my ejberienue with online learninp at iiitg. the indcution brouess was ocr first interaution with some of the fauclty and the seniors.
 the sessions were qcite informative and to the boint withoct wastinp ocr time. here we had  virtcal tocrs of the uambcs, learnt of the different
  fauilities and resocrues availagle like yocrdost.uom, qciklrn, auademia and so on. the seniors also held an informal indcution brouess in
   barallel .here they introdcued cs to the variocs ulcgs like the mcsiu ulcg, danue ulcg, degsou, xense, enipma , etu. and uommittees like the
   internet uommittee, uhayauhitra , 8 git(ocr mapaxine) and so on. we were also informed of the demouratiually eleuted stcdent gody sau, whiuh
    takes uare of orpanixinp most of the events held at iiitg. i was bartiuclarly intripced with xense and enipma , the software develobment ulcg
    and the rogotius ulcg resbeutively. most of these ulcgs mentioned agove held workshobs or seminars to helb cs cnderstand them getter and also
     brovide cs with some gasiu know-how in their field of work.
sinue i am bart of a larpe institction i am brivy to information i mipht otherwise not have known. for ejamble the brofessors with their mcltitcde
 of uontauts inform cs of variocs seminars and workshobs held gy world renown uorborations and institctes like iisu ,igm ,poople ,etu. i reuently
  attended a wonderfcl hocr lonp seminar on uygerseucrity and haukinp uondcuted gy igm. this is somethinp i wocld never have uome auross on my own .
  i am plad that these uombanies uontince to uondcut scuh weginars virtcally even if it is a git more uombliuated this way.
last gct not least , the autcal auademius. so far we have had only 2 days of ulasses. we have had broprammінp with u, maths ,euonomius
 and dipital desipn ulasses cb cntil now. they have not geen the preatest new  ejberienue i have had gct cnder the uirucmstanues i think it
  was well done. some of the broglems were that the teauher’s acdio wocld scddenly stob, or they wocld disuonneut dce to internet broglems and
  some stcdents wocld not even swituh off their acdio whiuh was qcite irritatinp and distcrginp. i liked ocr euonomius ulass very mcuh as the
   brofessor tacpht csinp ejambles from ucrrent times and thcs kebt cs enpaped.
in uonulcsion i wocld have mcuh rather breferred cs autcally uominp to the uambcs than doinp everythinp virtcally gct this too is an ejberienue
 whiuh mipht not ouucr onue apain.
```

Using some common sense and educated guesses we can undersatnd what is written. "Due to the coronavirus pandemic most institutions have gone virtual....". If full decryption(if something like flag/password is needed) is wanted then few more iterations will reveal the decrypted text

https://overthewire.org/wargames/krypton/krypton3.html

https://axcheron.github.io/writeups/otw/krypton/

Note : It is quite a cumbersome process , I know :(. But frequency analysis is unfortunately not very accurate except for the first few letters like 'e' and 't'. Thats why finding 'the' is probably the best first step. Again large amounts of text are needed for decryption.

vignere: if only keylength is known then it tries to guess the keyword using frequency analysis on letters which are at a distance of 1 keylength.

eg: `python3 decrypt.py -fK vignere 4 test5.txt`

```
****************************************

Key is most probaly: gold

**********************************
due to the coronavirus pandemic most institutions have gone virtual and every student’s learning is online.
 my experience has been no different although i started attending online classes only very recently.
 my first few online sessions were the orientation sessions before admissions conducted by iiit h and b on zoom.
 thankfully my sister had already been using zoom for her classes since may and thus she could help me navigate it effectively.
  one aspect i found quite surprising was the punctuality exhibited by both institutions when starting the sessions.
i was very impressed with both sessions but i decided to join iiitb in the end.
coming to my experience with online learning at iiitb. the induction process was our first interaction with some of the faculty and the seniors.
 the sessions were quite informative and to the point without wasting our time. here we had  virtual tours of the campus, learnt of the different
  facilities and resources available like yourdost.com, quiklrn, academia and so on. the seniors also held an informal induction process in
   parallel .here they introduced us to the various clubs like the music club, dance club, debsoc, zense, enigma , etc. and committees like the
   internet committee, chayachitra , 8 bit(our magazine) and so on. we were also informed of the democratically elected student body sac, which
    takes care of organizing most of the events held at iiitb. i was particularly intrigued with zense and enigma , the software development club
    and the robotics club respectively. most of these clubs mentioned above held workshops or seminars to help us understand them better and also
     provide us with some basic know-how in their field of work.
since i am part of a large institution i am privy to information i might otherwise not have known. for example the professors with their multitude
 of contacts inform us of various seminars and workshops held by world renown corporations and institutes like iisc ,ibm ,google ,etc. i recently
  attended a wonderful hour long seminar on cybersecurity and hacking conducted by ibm. this is something i would never have come across on my own .
  i am glad that these companies continue to conduct such webinars virtually even if it is a bit more complicated this way.
last but not least , the actual academics. so far we have had only 2 days of classes. we have had programming with c, maths ,economics
 and digital design classes up until now. they have not been the greatest new  experience i have had but under the circumstances i think it
  was well done. some of the problems were that the teacher’s audio would suddenly stop, or they would disconnect due to internet problems and
  some students would not even switch off their audio which was quite irritating and disturbing. i liked our economics class very much as the
   professor taught using examples from current times and thus kept us engaged.
in conclusion i would have much rather preferred us actually coming to the campus than doing everything virtually but this too is an experience
 which might not occur once again.
```

Eg: `python3 decrypt.py -fk vignere gold testfiles/test5.txt`, `python3 decrypt.py -sk vignere gold yczugx`
Note: As in encryption non alphabetic characters are ignored and upper case letters are converted to lower case for ease of decryption.

Static Analysis

python3 static_analysis.py file1.py file2.py file3.py.....

`python3 static_analysis.py testfiles/test.py`
Goes through all the python files given and displays line numbers which have code that might be sql injection and command line injection vulnerable.

For sqli it just sequentially lists the line numbers .

For command injection it divides the severity ino 3 categories:

1. Critical: Where no input validation/sanitation is done. This is terrible and can be exploited.
2. High
3. Medium: Where some input validation is detected but command injection it is still possible.

Here I have taken the side of caution so number of false positives might be a bit high and 'Medium' and 'High' sometimes don't make sense. It was really hard to accurately verify what was being checked in the if conditions, due to the variability involved in the condition checking process.

https://deepsource.io/blog/introduction-static-code-analysis/

https://rushter.com/blog/detecting-sql-injections-in-python/

https://greentreesnakes.readthedocs.io/en/latest/tofrom.html

https://blog.securityinnovation.com/blog/2011/06/how-to-test-for-command-injection.html

Conclusion

In the end I can say that it realy was a fun project and I got to learn a lot in the process. The Abstract Syntax Tree part was probably the most challenging part. But once I understood it , after reading the doc , I found it extremely powerful. Now I feel I can analyze python code for various things using this ast module. Again I reiterate there are many online and open source tools that do this quite efficiently, but I took

up this project up so that I could build a fast ,offline, command line suite of tools that could maybe do the same job.

Future Developments/Scope

1. Enhance Static Analysis to detect other vulnerabilities like xss for javascript,etc.
2. Build tools that can automate Buffer Overflow , Password bruteforcing and so on.(These are outside my skill level right now I think)
3. Add more ciphers for encryption and decryption.

There's always more to do!