

Forum:[Hack Forums](#)

Thread: [A forensic toolkit in Python](#)

CaptainBlack - May 15, 2017, 6:05 pm

A project I been working on:

<https://github.com/MonroCoury/Forensic-Tools>

Features:

- Document metadata extraction.
- Image EXIF metadata extraction.
- Firefox database parsing, including extracting cookies, history, form history, Google searches, and downloads. Can limit results to a certain time range.
- Skype database parsing, including Account details, contacts with full details, call log, and messages. Ability to look for messages/calls within a given time range and/or from/to a specific partner.
- What'sapp android database parsing, with the ability to extract full details.
- Results are saved to html tables with row background highlighting for easier reading.
- I'm trying to make it as simple and easy to use as possible. Firefox scanner attempts to find the default databases across different platforms on its own should the user forget to point it.

Still a work in progress.. Planned features:

- Chrome browsing data extraction
- Internet Explorer browsing data extraction
- Windows registry parsing
- PDF, zip, and rar password cracking

Lambasoft - May 16, 2017, 2:43 am

Cool project. I hope this wont be used for malicious content.

One suggestion, it would be better if you went Object Oriented.

CaptainBlack - May 16, 2017, 4:18 am

^I sure hope so myself! I mean it's the easiest thing to modify them and make them send the resulting html code to a server or even an email address before it's saved to the local drive. But that's why I chose Python instead of say Java; anyone who has basic understanding of Python shouldn't have a problem telling where the source code is sending the results. Whereas you would need a decompiler to convert Java byte code into readable source code.

Very interesting suggestion! I'm writing some of the upcoming scripts oop way, like the password cracker script. I'll change some of the existing ones from procedural to oop soon as I get time.

Ensky - May 16, 2017, 5:37 am

Really cool project my man, I'll take a look!

Dazzee - May 16, 2017, 5:40 am

Good luck.

Perhaps you could add the ability to scan for deleted files like in Autopsy.

????Kosdff - May 16, 2017, 5:44 am

Nice project, I will take a look at this :)

SIM-1866423 - May 16, 2017, 6:57 am

I'll test them soon, let me know if you need help. Welcome the section ..

Goodies - May 17, 2017, 12:35 pm

It'd be nice to see these be modular and actually exist inside of an idiomatically-designed framework.

Aside from that, after looking it over for a few minutes, I'd be surprised if all of these scripts ran. I see some indentation issue like here:

https://github.com/MonroCoury/Forensic-Tools/blob/master/common_methods.py#L39

Also, in that same function, do not call `sys.exit` from inside your function. Functions should often return statuses if they're not going to return values. If the file exists, raise the exception! The caller should be performing the error handling.

You should also be using context managers more than you are. Some of this code, if it fails, will never close certain objects properly as they are not in context managers and have no finally clause on their try/except statements.

This was just a couple things I saw in the first script I looked at. I'm guessing the hundreds of other SLOC were done in a similar fashion. I'm also guessing that you primarily just want to get pen to paper and get a working POC, not trying to perfect your source.

All in all, I really like the idea of a modular forensics framework/toolkit that will perform commonly employed forensic techniques (file signature analysis, integrity checking, etc), but the presentation of it is just as important if you want it in widespread use. I'd recommend making a nice CLI interface. You can use metasploit or w3af as examples. Something where you can load modules and execute them with parameters.

Ideally, you'd be able to provide your own modules fairly simply.

Pylon - May 17, 2017, 4:35 pm

You should definitely go the Object Oriented route. It makes life so much easier and readable in the long run. You should also turn this in to a proper library with an `__init__.py` file. So far so good, keep us updated!

Good luck!

If you want to learn Python (or just about any language), [Join Me!](#)
How do you like my redneck signature?

CaptainBlack - May 17, 2017, 5:01 pm

(May 17, 2017 12:35 PM)Goodies Wrote: [\[CITE\]](#) It'd be nice to see these be modular and actually exist inside of an idiomatically-designed framework.

Aside from that, after looking it over for a few minutes, I'd be surprised if all of these scripts ran. I see some indentation issue like here:

https://github.com/MonroCoury/Forensic-Tools/blob/master/common_methods.py#L39

Also, in that same function, do not call `sys.exit` from inside your function. Functions should often return statuses if they're not going to return values. If the file exists, raise the exception! The caller should be performing the error handling.

You should also be using context managers more than you are. Some of this code, if it fails, will never close certain objects properly as they are not in context managers and have no finally clause on their `try/except` statements.

This was just a couple things I saw in the first script I looked at. I'm guessing the hundreds of other SLOC were done in a similar fashion. I'm also guessing that you primarily just want to get pen to paper and get a working POC, not trying to perfect your source.

All in all, I really like the idea of a modular forensics framework/toolkit that will perform commonly employed forensic techniques (file signature analysis, integrity checking, etc), but the presentation of it is just as important if you want it in widespread use. I'd recommend making a nice CLI interface. You can use `metasploit` or `w3af` as examples. Something where you can load modules and execute them with parameters.

Ideally, you'd be able to provide your own modules fairly simply.
I test every script thoroughly before uploading it, they all run. The indentation error you mentioned isn't really an indentation error. The return statement isn't meant to be part of the if block.

I use `with` statement to handle streams. No need to use `finally` with that as it closes the stream automatically.

I have thought about making this oop and posting it to pypi only recently.. Guess it'll take me a while to modify everything. Originally, I meant for the toolkit to be just a bunch of scripts runnable from a console, hence using `sys.exit` instead of raising errors.

Appreciate taking the time to review it! I'll definitely keep your suggestions in mind

Pylon - May 17, 2017, 5:07 pm

Your doing great, keep up the good work!

If you want to learn Python (or just about any language), [Join Me!](#)
How do you like my redneck signature?

CaptainBlack - May 18, 2017, 3:00 pm

Everyone, big thanks for the nice words and the rep!

Added chrome browser database parsing.

Pylon - May 18, 2017, 10:23 pm

Dope! I'll check it out soon!

If you want to learn Python (or just about any language), [Join Me!](#)

How do you like my redneck signature?

CaptainBlack - *May 25, 2017, 7:00 am*

Project updated, several new features added

CaptainBlack - *January 3, 2018, 9:12 am*

Update: Fixed a couple bugs and added Facebook database parsing.