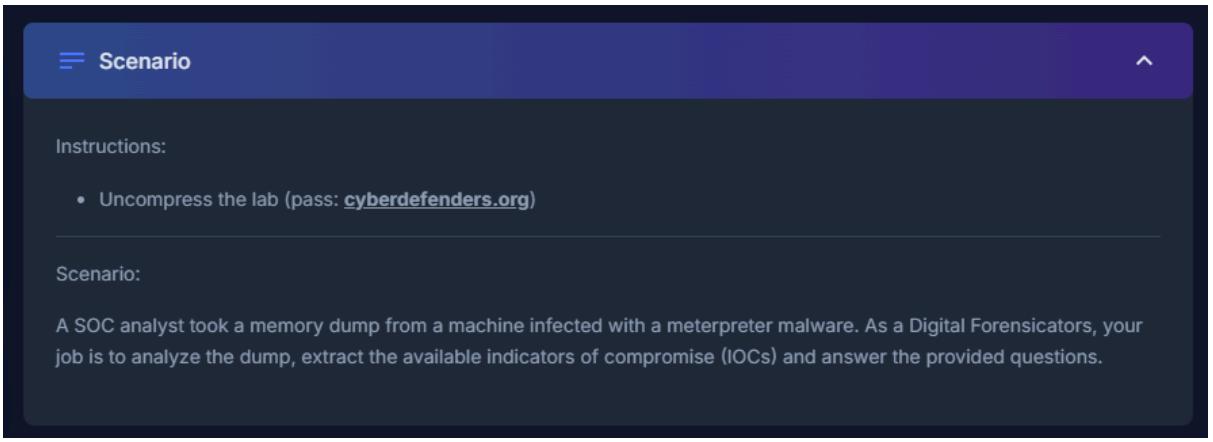
Projekt nr. 3
KRYCY
Autorzy: Piotr Szewczyk, Paweł Murdzek

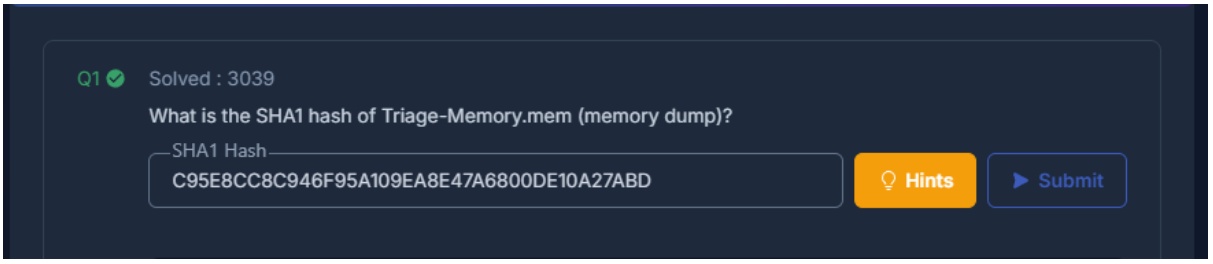
Wybrany projekt: DumpMe

Scenariusz zadania:



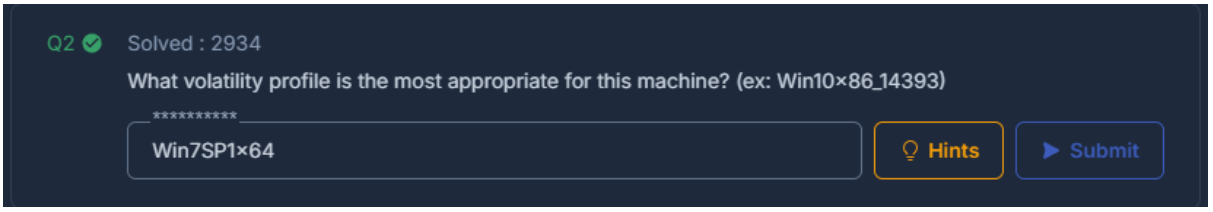
Pytania dotyczące zadania:

Pytanie nr. 1

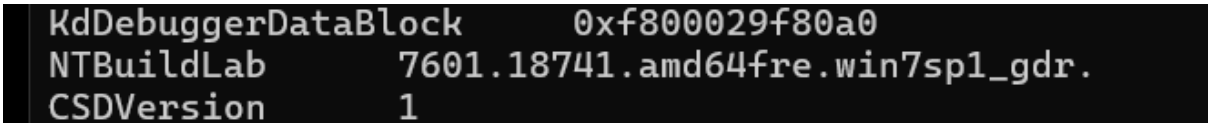


Na początku pytają się o SHA1 pobranego pliku zgodnie z praktyką. Po uruchomieniu komendy w powershellu “Get-FileHash -Algorithm SHA1” dostajemy powyższy hash i wykonujemy zadanie.

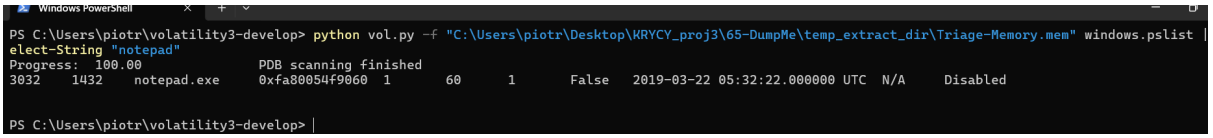
Pytanie nr. 2



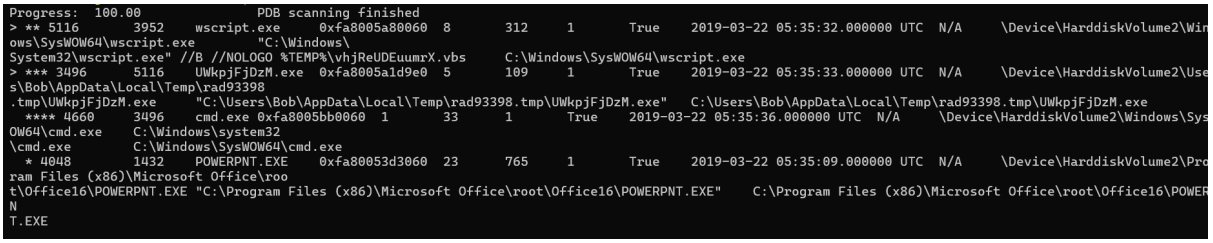
Z jakim systemem mamy doczynienia? Używając Volatility3 uruchamiając komendę “windows.info” otrzymujemy informacje o systemie. Dopasowywujemy informację do formatu odpowiedzi.



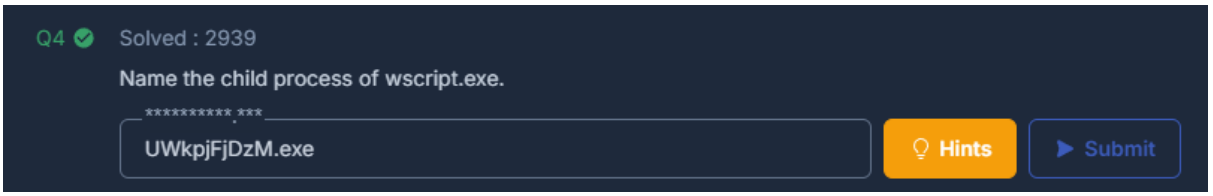
Pytanie nr. 3
Kolejne pytanie dotyczy PID notatnika, filtrując wyszukiwania komendą “windows.pslist | Select-String "notepad"” znajdujemy wynik:



Pytanie nr. 4
W kolejnym pytaniu dostajemy pytanie o child process wscript.exe. Używając funkcji “windows.pstree”, dowiadujemy się, że PID wscript.exe = 5116. Znów używamy komendy pstree, tym razem filtrując procesy 5116.

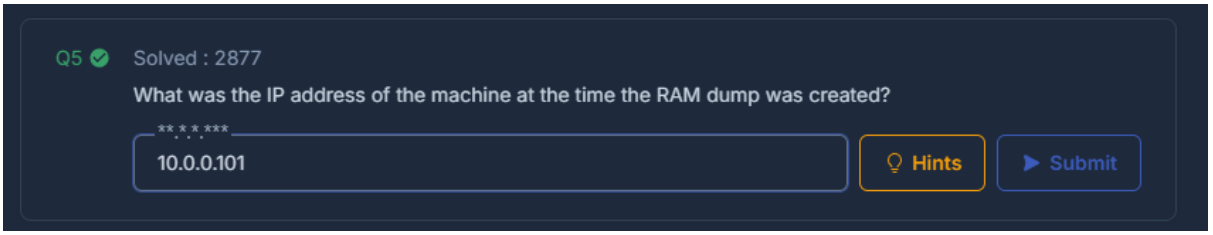


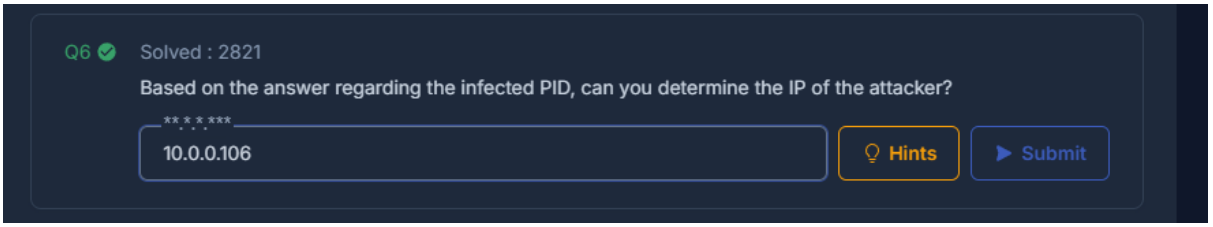
Bezpośrednim child process jest:

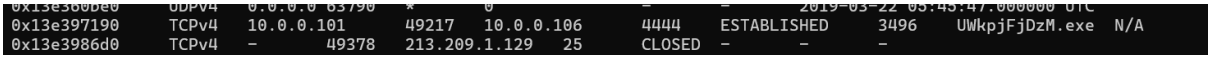


Pytanie nr. 5 oraz Pytanie nr. 6

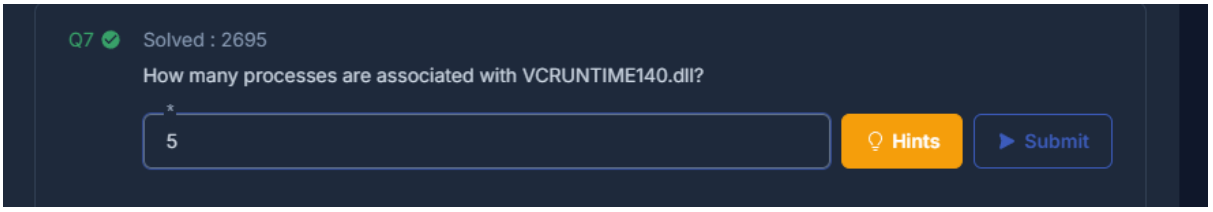
W tym zadaniu trzeba znaleźć adres IP ofiary oraz adres atakującego, komenda “netscan” pozwala na wyświetlenie wszystkich połączeń sieciowych. Po wyniku od razu widać, że tym adresami są:

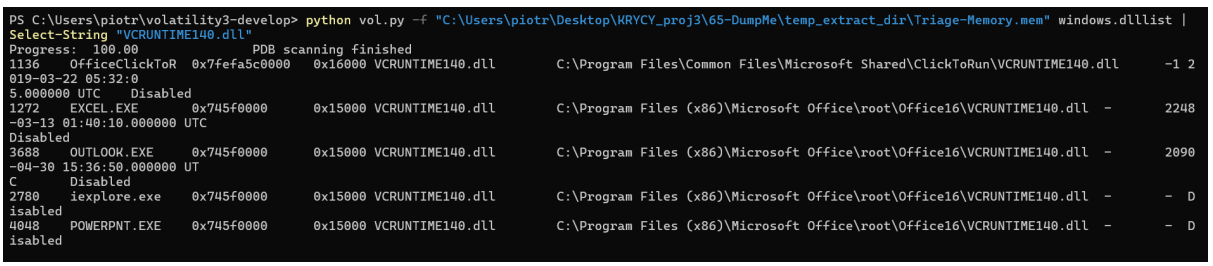






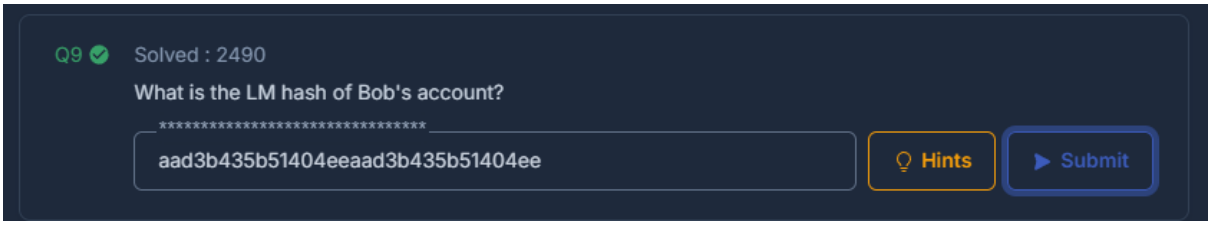
Pytanie nr. 7
W tym pytaniu mamy za zadanie odnaleźć ile procesów jest związanych z VCRUNTIME140.dll. Do sprawdzania powiązań z dll służy komenda “windows.dlllist”. filtrujemy po VCRUNTIME140.dll i liczymy ilość wystąpień.



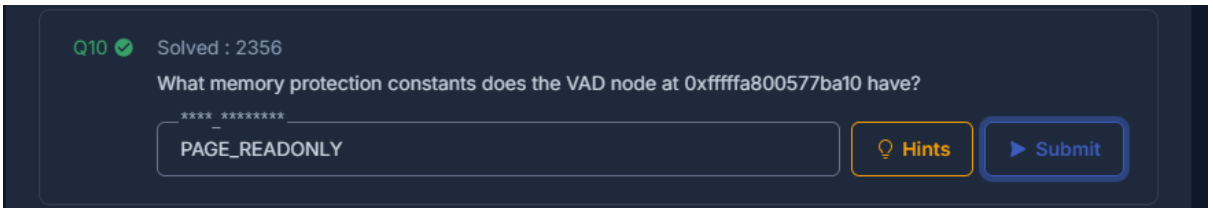


Pytanie nr. 8
Pytanie dotyczy o hash md5 zainfekowanego procesu. Z wcześniejszych logów wynika, że tym procesem jest PID = 3496, czyli UWkpjFjDzM.exe. Zatem trzeba odzyskać ten plik. Wtyczka dumpfiles pozwala na zapisanie takiego pliku. Używając komendy “windows.dumpfiles --pid 3496” zapisujemy pliki związane z tym procesem. Wyszukujemy plik z fragmentem w/w .exe w nazwie i odczytujemy hash’a md5. Niestety uzyskany hash nie pasował do klucza odpowiedzi. Wyczytaliśmy, że Volatility2 inaczej podchodzi do zapisu plików niż wersja 3. Niemniej jednak podejście jest wydaje się ok.

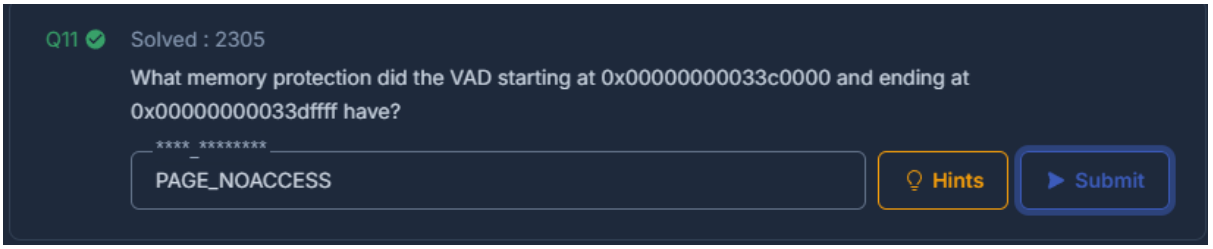
Pytanie nr. 9
Pytanie dotyczące LM hash'a Bob'a. Użycie komendy "hashdump" pozwoliło na odczytanie zawartości rejestru (gałęzi SAM i SYSTEM) z pamięci. Zidentyfikowano konto użytkownika Bob, dla którego hash LM ma wartość **aad3b435b51404eeaad3b435b51404ee**, czyli nie istnieje - przestarzała metoda hashowania haseł.



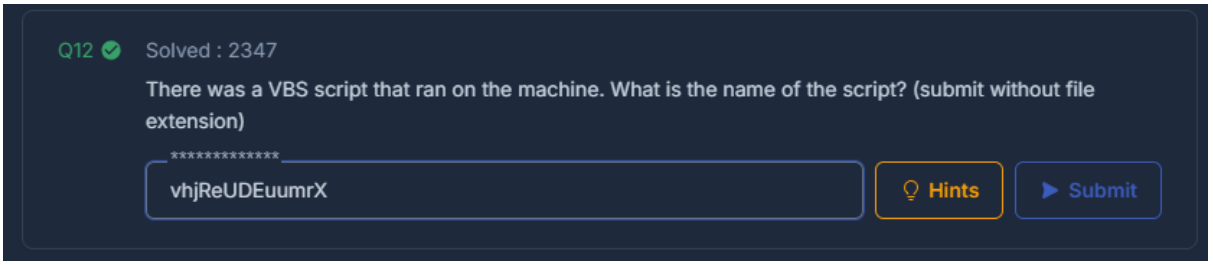
Pytanie nr. 10
W tym pytaniu zadaniem jest sprawdzenie jakie uprawnienia ma konkretny węzeł VAD. VAD - Virtual Address Descriptior - część jądra windowsa mówiąca o pozwoleniach procesów w konkretnym zakresie pamięci. Do znajdowania informacji związanych z VAD wykorzystujemy komendę "vadinfo".



Pytanie nr. 11
Pytanie dotyczyło kolejnego node'a tym razem zaczynającego i kończącego się zgodnie z pytaniem. Podobne zadanie do potrzebnego, trzeba było zacząć od tego adresu.

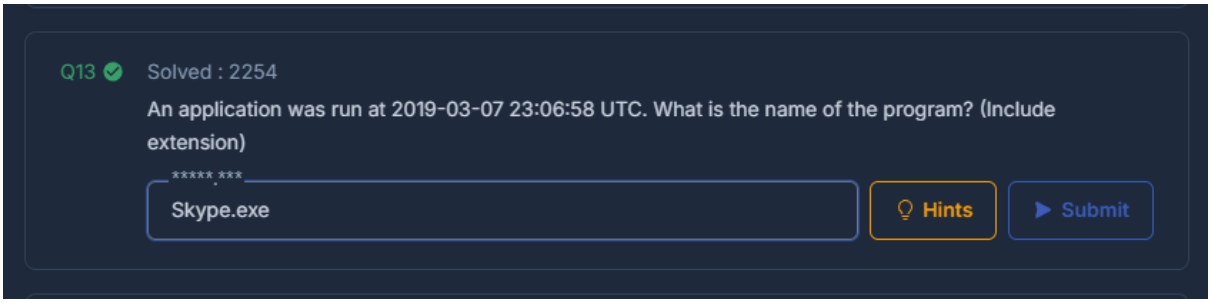


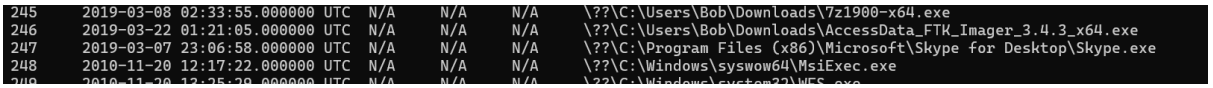
Pytanie nr. 12
Zadanie dotyczy nazwy skryptu VBS uruchomionego na maszynie. We wcześniejszych zrzutach ekranu widać przy procesie wscript.exe (pytanie nr. 4) widać od razu nazwę skryptu w folderze TEMP.



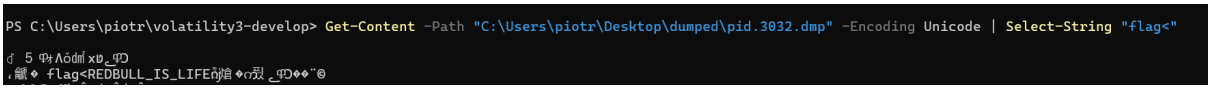
Pełni funkcję loader'a.

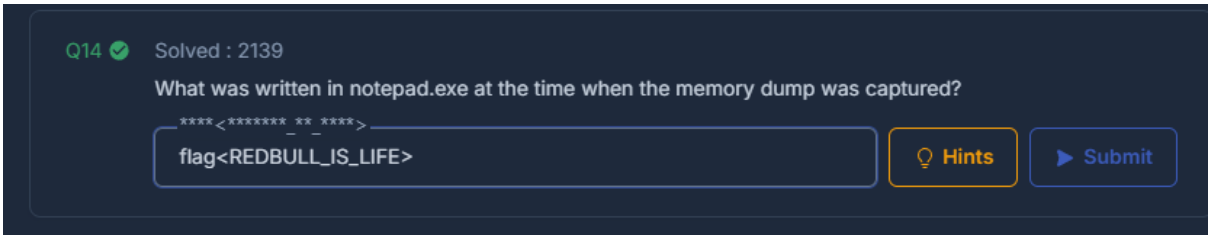
Pytanie nr. 13
Zadanie polegało na znalezieniu uruchomionej aplikacji w konkretnym czasie. Użyto pluginu shimcache i przejrzano wynik pod kątem danego wystąpienia. Shimcache zawiera metadane wszystkich uruchomionych, nawet historycznych programów .exe



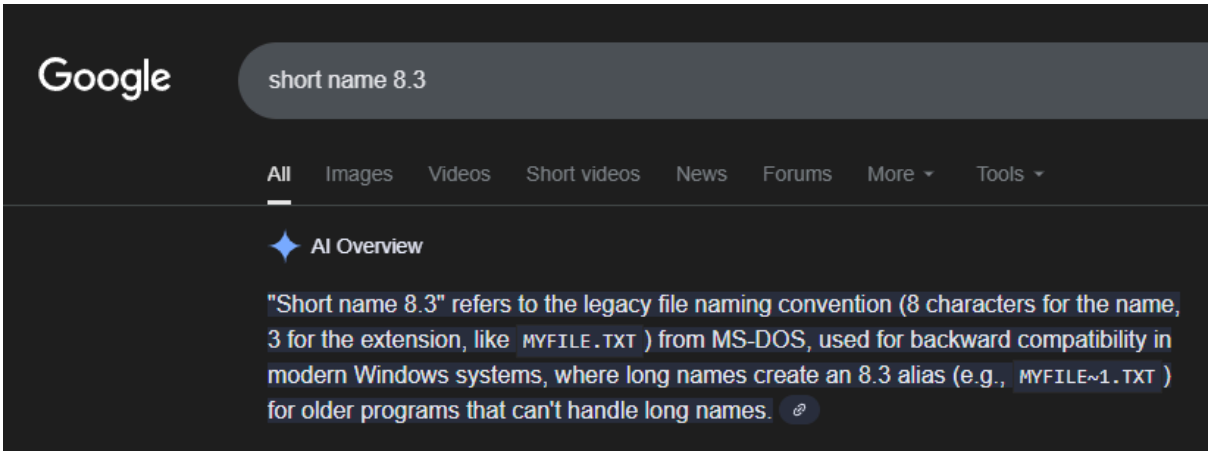


Pytanie nr. 14
W tym pytaniu pytają się nas o to co zostało napisane w notatniku bez jego zapisu. Z wcześniejszego zadania znamy już PID notatnika = 3032. Zadanie było ułatwione o tyle ze w masce w zadaniu widać było znaki <> i że to właśnie chodzi o flag<>. Bez tego trzeba by było filtrować bardziej złożenie, żeby móc coś wyłapać (key, password itp.).Udało się to rozszyfrować za pomocą komendy Get-Content.



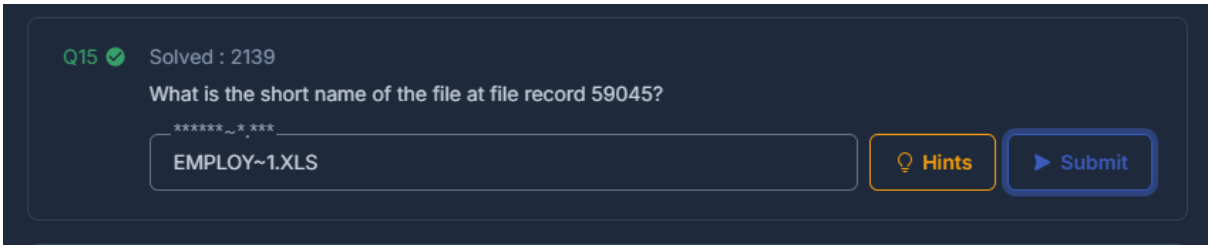


Pytanie nr. 15
W pytaniu 15 zadaniem jest znalezienie krótkiej nazwy pliku z rekordu 59045. Czyli trzeba znaleźć konkretny rekord z Master File Table (MFT). Definicja krótkiej nazwy:

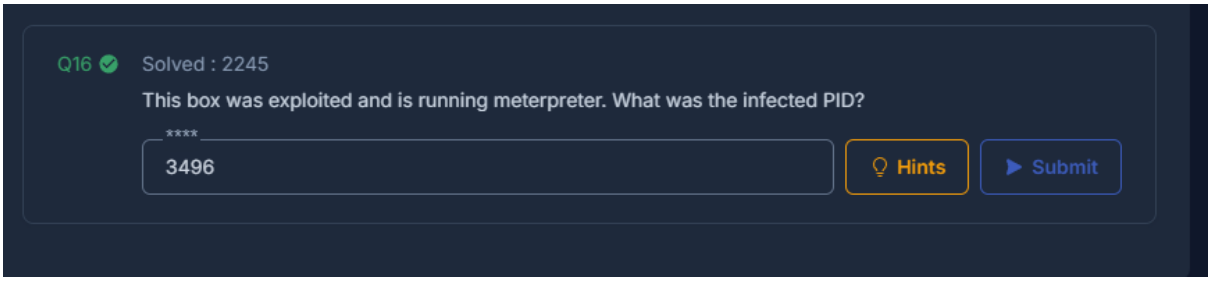


Używając komendy “windows.mftscan.MFTScan” udało dostać się do szukanej danej.

```
PS C:\Users\piotr\volatility3-develop> python vol.py -f "C:\Users\piotr\Desktop\KRYCY_proj3\65-DumpMe\temp_extract_dir\Triage-Memory.mem" windows.mftscan |
Select-String "59045"
usage: vol.py [-h] [-c CONFIG] [--parallelism [{processes,threads,off}]]
              [-e EXTEND] [-p PLUGIN_DIRS] [-s SYMBOL_DIRS] [-v] [-l LOG]
              [-o OUTPUT_DIR] [-q] [-f FILE] [--write-config]
              [--save-config SAVE_CONFIG] [--clear-cache]
              [--cache-path CACHE_PATH] [--offline | -u URL]
              [--filters FILTERS] [--hide-columns [HIDE_COLUMNS ...]]
              [-r RENDERER] [--single-location SINGLE_LOCATION]
              [--stackers [STACKERS ...]]
              [--single-swap-locations [SINGLE_SWAP_LOCATIONS ...]]
              PLUGIN ...
vol.py: error: argument PLUGIN: plugin windows.mftscan matches multiple plugins (windows.mftscan.ADS, windows.mftscan.MFTScan, windows.mftscan.ResidentData)
PS C:\Users\piotr\volatility3-develop>
```



Pytanie nr. 16
Ostatnie pytanie dotyczy zainfekowanego procesu, już wcześniej było wiadomo, że jest to PID = 3496.



Podsumowanie:
Analizowany incydent to wieloetapowy atak typu LoTL, który rozpoczął się od otwarcia złośliwego arkusza `EMPLOY~1.XLS`, co umożliwiło wykonanie skryptu VBS `vhjReUDEuumrX` i pełne przejęcie kontroli nad systemem przez proces `UWkpjFjDzM.exe` (PID 3496) działający z agentem Meterpreter, prowadząc ostatecznie do kradzieży danych uwierzytelniających użytkownika Bob oraz zaawansowanej manipulacji strukturami pamięci VAD.