# Useful Python Scripts for PE File Analysis and Threat Hunting

## Introduction

Digital certificates play a crucial role in ensuring the legitimacy and security of Portable Executable (PE) files. In this guide, we'll explore the importance of digital certificates, how to analyze PE files, what to do when receiving a malicious hash, the process of threat hunting, formulating hypotheses, identifying related threats, and automating correlation rules to monitor anomalies.

## 1. Extract Compilation Date from a PE File

This script uses the pefile library to extract the compilation date of a PE file.

import pefile  
from datetime import datetime  
  
def get\_pe\_compilation\_date(file\_path):  
 pe = pefile.PE(file\_path)  
 timestamp = pe.FILE\_HEADER.TimeDateStamp  
 compilation\_date = datetime.utcfromtimestamp(timestamp)  
 return compilation\_date  
  
# Example usage  
file\_path = "path/to/suspicious/file.exe"  
compilation\_date = get\_pe\_compilation\_date(file\_path)  
print(f"Compilation Date: {compilation\_date}")

## 2. Extract Certificate Information from a PE File

This script extracts certificate details, including the issuer, serial number, and expiration date.

import pefile  
  
def extract\_certificate\_info(file\_path):  
 pe = pefile.PE(file\_path)  
 cert\_info = []  
 if hasattr(pe, 'DIRECTORY\_ENTRY\_SECURITY'):  
 for security\_entry in pe.DIRECTORY\_ENTRY\_SECURITY:  
 if hasattr(security\_entry, 'Certificate'):  
 cert = security\_entry.Certificate  
 cert\_info.append({  
 'Issuer': cert.Issuer,  
 'Serial Number': cert.SerialNumber,  
 'Expiration Date': cert.ValidTo  
 })  
 return cert\_info  
  
# Example usage  
file\_path = "path/to/suspicious/file.exe"  
cert\_info = extract\_certificate\_info(file\_path)  
print(cert\_info)

## 3. Fetch Detailed Certificate Information from VirusTotal

This script uses the VirusTotal API to fetch detailed certificate information, including the expiration date.

import requests  
  
def get\_certificate\_expiry(api\_key, file\_hash):  
 url = f"https://www.virustotal.com/api/v3/files/{file\_hash}"  
 headers = {"x-apikey": api\_key}  
 response = requests.get(url, headers=headers)  
 cert\_info = response.json().get('data', {}).get('attributes', {}).get('pe\_info', {}).get('digital\_signatures', [])  
 cert\_details = []  
 for cert in cert\_info:  
 issuer = cert.get('issuer')  
 serial\_number = cert.get('serial\_number')  
 expiration\_date = cert.get('valid\_to')  
 cert\_details.append({  
 'Issuer': issuer,  
 'Serial Number': serial\_number,  
 'Expiration Date': expiration\_date  
 })  
 return cert\_details  
  
# Example usage  
api\_key = "your\_virustotal\_api\_key"  
file\_hash = "known\_hash"  
cert\_details = get\_certificate\_expiry(api\_key, file\_hash)  
print(cert\_details)

## 4. Correlate Compilation Date and Certificate Expiry Date

This script correlates the PE file's compilation date with the certificate's expiration date to determine if the file was compiled before the certificate expired.

from datetime import datetime  
  
def correlate\_dates(compilation\_date, cert\_details):  
 for cert in cert\_details:  
 cert\_expiry\_date = datetime.strptime(cert['expiration\_date'], "%Y-%m-%dT%H:%M:%SZ")  
 if compilation\_date < cert\_expiry\_date:  
 print(f"Compilation Date: {compilation\_date}")  
 print(f"Issuer: {cert['issuer']}")  
 print(f"Serial Number: {cert['serial\_number']}")  
 print(f"Expiration Date: {cert\_expiry\_date}")  
 print("The PE file was compiled before the certificate expired.\n")  
 else:  
 print(f"Compilation Date: {compilation\_date}")  
 print(f"Issuer: {cert['issuer']}")  
 print(f"Serial Number: {cert['serial\_number']}")  
 print(f"Expiration Date: {cert\_expiry\_date}")  
 print("The PE file was compiled after the certificate expired.\n")  
  
# Example usage  
compilation\_date = datetime.strptime("2023-03-01", "%Y-%m-%d")  
cert\_details = [  
 {  
 'issuer': 'Example Issuer',  
 'serial\_number': '1234567890',  
 'expiration\_date': '2023-04-01T00:00:00Z'  
 }  
]  
correlate\_dates(compilation\_date, cert\_details)