Integriography: A Journal of Broken Locks, Ethics, and Computer Forensics

Musings about ediscovery, computer forensics, cyber security, and the state of play in all …..

Using Python to parse and present Windows 64 bit timestamps

January 16, 2010 Integriography Leave a comment Go to comments

I'm working on learning Python since Perl, even after 20 years, still doesn't stick in my head. The phrase "like a duck to water" doesn't quite apply to my experience with Python, but I'm certainly swimming along nicely.

Since I learn languages and tools more effectively when I have a real problem to work on, I set out to parse the NTFS MFT record using Python. This was going swimmingly until I hit the file MAC times. According to Microsoft:

"A file time is a 64-bit value that represents the number of 100-nanosecond intervals that have elapsed since 12:00 A.M. January 1, 1601 Coordinated Universal Time (UTC)."

So, two problems: 1) It is a 64 bit value and python only has 32 bit longs and b) it doesn't use the Unix definition of epoch.

After a lot of false starts, I learned the power of dictionaries and classes and came up with the following:

```
from datetime import date, datetime

class WindowsTime:
    def __init__(self, low, high):
        self.low = long(low)
        self.high = long(high)

        self.unixtime = self.GetUnixTime()
        self.utc = datetime.utcfromtimestamp(self.unixtime)
        self.utcstr = self.utc.strftime("%Y/%m/%d %H:%M:%S.%f")

# Windows NT time is specified as the number of 100 nanosecond intervals since January 1, 1601.
# UNIX time is specified as the number of seconds since January 1, 1970.
# There are 134,774 days (or 11,644,473,600 seconds) between these dates.
    def GetUnixTime(self):
        t=float(self.high)*2**32 + self.low
        return (t*1e-7 - 11644473600)
```

I have a module that defines a dictionary and parses chunks of data read from the MFT:

```
# Decodes the Standard Information attribute in a MFT r
def decodeSIrecord(s):

 d = {}
 d['crtime'] = WindowsTime(struct.un
```

```
                                    struct.unp
  ...
```

Then just do:

```
SIobject = decodeSIrecord(data)
```

And you can print the times out directly from the

```
print "CRTime: %s" % (SIobject['crtir
```

This isn't rocket science, and there's probabaly ; ıg to use Python to work with Windows filesystems and metadata, this could come in han

Related

Dissecting a Blackhole 2 PDF (mostly) with peepdf.
In "malware"

analyzeMFT - a Python tool to deconstruct the Windows NTFS $MFT file
In "Software Tools"

Updated analyzeMFT, $MFT sequence numbers, and NTFS documentation
In "Software Tools"

Categories: Writing code Tags: forensics, python, windows 64 bit timestamps
Comments (1) Trackbacks (0) Leave a comment Trackback

1. 
   Syd P
   February 19, 2010 at 12:37 am
   Reply

Try using the unsigned long long 'Q' for 64-bit values.

i.e. ft = struct.unpack('<Q',fh.read(8))[0]

Cheers,
Syd

1. No trackbacks yet.

Leave a Reply

Enter your comment here...

analyzeMFT – a Python tool to deconstruct the Windows NTFS $MFT file CRU DataPort (Wiebetech) USB Writeblocker Quick Review
RSS feed

analyzeMFT

If you are looking for analyzeMFT, it can be found on GitHub: https://github.com/dkovar/analyzeMFT

Recent Posts

- SANS DFIR Summit Prague – Blue Team Perspectives slides
- Patents in the DFIR community space
- IRcollect – collect incident response information via raw disk reads and $MFT parsing
- analyzeMFT – ADS support added
- Adventures in Powershell for IR

forensics

- Cyber Crime 101
- Didier Stevens
- Forensic Focus
- Into The Boxes
- Windows Incident Response

Archives

- October 2013
- August 2013
- July 2013
- June 2013
- April 2013
- January 2013
- November 2012
- August 2012
- June 2012
- July 2011
- June 2011
- May 2011
- April 2011
- March 2011
- November 2010
- August 2010
- June 2010
- February 2010
- January 2010
- December 2009
- November 2009

Email Subscription

Enter your email address to subscribe to this blog and receive notifications of new posts by email.

Join 35 other followers

Sign me up!

Search

Top