Analysis and Evaluation of the Windows Event Log for Forensic Purposes

Introduction

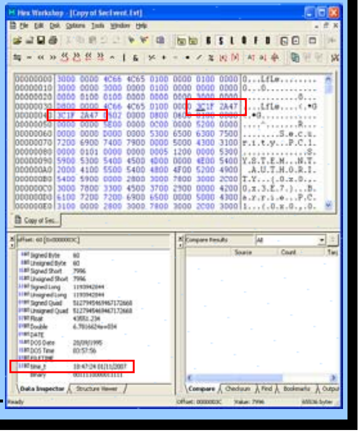
The windows event log is used in digital forensic cases, but unfortunately it is flawed in many ways, and often cannot be seen as a verifiable method of determining events.

In the past few years there have been a few highly publicised cases where the data that is contained within the event log is used to successfully secure a conviction.

The aim of this project is to develop a solution that addresses the flaws in the Windows event logging service.

Windows Event Log Weaknesses

To adjust the time by 1 second, the left most bit needed to be incremented by one. For example '3C1F 2A47' is equal to '18:47:24 01/11/2007' and to increase this by 1 second the new hexadecimal value would have to be '3D1F 2A47'.


Hex Workshop - [Copy of SecEvent.Evt]
File Edit Disk Options Tools Window Help
Copy of Sec...

```
00000000 3000 0000 4C66 4C65 0100 0000 0100 0000 0...LfLe........
00000010 3000 0000 3000 0000 0100 0000 0000 0000 0...0...........
00000020 0000 0100 0100 0000 0000 0000 3000 0000 ............0...
00000030 0800 0000 4C66 4C65 0100 0000 3C1F 2A47 ....LfLe....<.*G
00000040 3C1F 2A47 0502 0000 0800 0600 0100 0000 <.*G............
00000050 0000 0000 5E00 0000 0C00 0000 5200 0000 ....^.......R...
00000060 0000 0000 D000 0000 5300 6500 6300 7500 ........S.e.c.u.
00000070 7200 6900 7400 7900 0000 5000 4300 3100 r.i.t.y...P.C.1.
00000080 0000 0101 0000 0000 0005 1200 0000 5300 ..............S.
00000090 5900 5300 5400 4500 4D00 0000 4E00 5400 Y.S.T.E.M...N.T.
000000A0 2000 4100 5500 5400 4800 4F00 5200 4900  .A.U.T.H.O.R.I.
000000B0 5400 5900 0000 2800 3000 7800 3000 2C00 T.Y...(.0.x.0.,.
000000C0 3000 7800 3300 4500 3700 2900 0000 4200 0.x.3.E.7.)...B.
000000D0 6100 7200 7200 6900 6500 0000 5000 4300 a.r.r.i.e...P.C.
000000E0 3100 0000 2800 3000 7800 3000 2C00 3000 1...(.0.x.0.,.0.
```

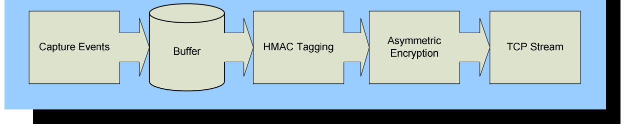
Offset: 60 [0x0000003C]

Signed Byte	60
Unsigned Byte	60
Signed Short	7996
Unsigned Short	7996
Signed Long	1193942844
Unsigned Long	1193942844
Signed Quad	5127945469467172668
Unsigned Quad	5127945469467172668
Float	43551.234
Double	6.7816624e+034
DATE	
DOS Date	28/09/1995
DOS Time	03:57:56
FILETIME	
time_t	18:47:24 01/11/2007
Binary	0011110000011111

Design

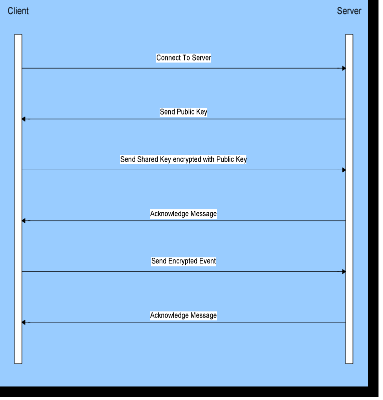
The .NET Framework provides the 'FileSystemWatcher' class which is able to monitor if files have been accessed, modified, deleted or renamed. It can be set to monitor specific files within a specified directory or the entire system.

To authenticate the messages HMAC will be used. RFC 2104 (1997) describes HMAC as a mechanism for message authentication using cryptographic hash functions.



Implementation

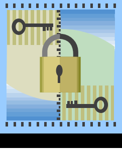
All the components of the system will be created using C# and the .NET framework, as they provide Event logging, Performance monitoring and encryption classes, which this application makes extensive use of.



Evaluation

It had been found that the symmetric encryption was 800% faster than asymmetric encryption.

HMAC hash signatures were tested to see how long it would take to do a brute force attack on them. It was discovered that approximately 21,093 keys were processed every second, this was then compared to the key entropy and how a longer random key would be harder to break.



Conclusions

It was found that the Windows event log lacked any form of security, and that it was possible to make changes to the data contained within it.

A combination of encryption and message hashing can be used to improve the integrity of the event log data.

References

RFC 2104. (1997). HMAC: Keyed-Hashing for Message Authentication. From URL: http://www.ietf.org/rfc/rfc2104.txt [Accessed 07/05/08]

Barrie Codona BSc (Hons) Network Computing 06007743@napier.ac.uk. Supervisor: Professor W. Buchanan. Second Marker: Dr G. Russell.