Edith Cowan University
Research Online

Australian Digital Forensics Conference | Security Research Institute Conferences

2008

The Malware Analysis Body of Knowledge (MABOK)

Craig Valli
Edith Cowan University

Originally published in the Proceedings of the 6th Australian Digital Forensics Conference, Edith Cowan University, Perth Western Australia, December 3rd 2008.
This Conference Proceeding is posted at Research Online.
http://ro.ecu.edu.au/adf/44

The Malware Analysis Body of Knowledge (MABOK)

Craig Valli and Murray Brand
School of Computer and Information Science
Edith Cowan University
c.valli@ecu.edu.au
mbrand0@student.ecu.edu.au

Abstract

The ability to forensically analyse malicious software (malware) is becoming an increasingly important discipline in the field of Digital Forensics. This is because malware is becoming stealthier, targeted, profit driven, managed by criminal organizations, harder to detect and much harder to analyse. Malware analysis requires a considerable skill set to delve deep into malware internals when it is designed specifically to detect and hinder such attempts. This paper presents a foundation for a Malware Analysis Body of Knowledge (MABOK) that is required to successfully forensically analyse malware. This body of knowledge has been the result of several years of research into malware dissection.

Keywords
Malware Analysis Body of Knowledge, MABOK, Malware, Digital Forensics

INTRODUCTION

Aycock (2006, p.1-12) defines malware as "software whose intent is malicious, or whose effect is malicious". Analysis of malicious software is essential for computer security professionals and digital forensic analysts and is emerging as an important field of research (Masood, 2004). Malware is often targeted at organizations and is increasingly using anti-forensics techniques to prevent detection and analysis. Anti-Forensics is described by Rogers (2006) as "attempts to negatively affect the existence, amount, and/or quality of evidence from a crime scene, or make the examination of evidence difficult or impossible to extract". Kessler (2007, p1) extends this definition in a practical sense by saying "anti-forensics, then, is that set of tools, methods, and processes that hinder such analysis". The movement towards the employment of anti-forensic techniques in malware could be attributed to increases in penalties arising from prosecutions and also from the illicit financial gain that can now be achieved from employing malware nefariously (Larsson, 2007). Commercial Anti-Virus (AV) software is often limited in its ability to detect and remove malware. It is highly unlikely to detect new malware that is unleashed on the internet, corporate intranet or that has been customized to target specific networks. It is also unlikely to detect malware that has been customized to target specific networks (Masood, 2004).

It is undeniable that there is a digital arms race between malware developers and malware researchers. As soon as a technique is developed by one side, the other side implements a counter measure. Two of the major trends discussed by Symantec (2008, p.49) is that attackers are increasingly motivated by financial gain and that there are indications that malware development is becoming increasingly commercialised and developed by professionals with extensive software engineering abilities. Another trend is that malware has an increasing variety of techniques available to hinder the forensic analyst (Falliere, 2007; Ferrie, 2008; Yason, 2007). This can include detection of the tools used by the forensic analyst and prevention of analysis via anti-debugging, anti-disassembly, anti-emulation, anti-memory dumping, incorporation of fake signatures and code obfuscation. Signature based detection of malware is dependent upon an analyst having already analysed the malware and extracted a signature as well as the end user having updated their malware signature file. Heuristics are dependent upon correct implementation. Although these techniques go some way in protecting a system they are far from infallible and only of minor assistance to the forensic analyst, especially if the malware is new or has been customised. The increasing availability of high speed network Internet connections has also enabled the rapid production and dissemination of the malware. All of these factors are contributing to increasing numbers of network borne malware with respect to volume, variety and complexity.

Security professionals in the field need to know how to determine if they are the target of an attack and how to eradicate or mitigate threats from their systems. This process of threat reduction can be assisted if security professionals have up to date methodologies and skill sets at their disposal.

The purpose of this paper is to present a Malware Analysis Body of Knowledge (MABOK) that can be used as a framework for competency development and assessment for the field of malware analysis. This body of

knowledge has been the result of several years of research into malware analysis. Over 1000 samples of malware collected from a honeypot have been successfully analysed and the MABOK is an attempt to map the knowledge needed to accomplish successful dissection of malware.

THE PROBLEM WITH MALWARE ANALYSIS

The spectrum of malware that represents a real threat is expansive. A non exhaustive list includes rootkits, worms, bots, trojans, logic bombs, viruses, phishing, spam, spyware, adware, keyloggers and backdoors. No computing platform or environment is immune to these threats. Traditionally, malware is thought of as a virus or worm that has a single function or payload. The resulting countermeasure for traditional malware has been the employment of a removal tool that was initiated by signature detection or by recognition of heuristics defined by specific behaviours. These tended to be like the malware they were responding to in that they were unitary or singular in purpose.

Modern network borne malware is increasingly multi-partite in nature incorporating several infection vectors and possible payloads in the one instance. Signature based systems that rely on file hashing or similar functions that uniquely identify malware based on file contents are increasingly failing due to the mass customisation allowable with the use of frameworks such as *MetaSploit* (Metasploit LLC, 2008). Furthermore, anti-forensic techniques are widely deployed to obfuscate infection, hinder detection and retard eventual removal of the malware. This increasing complexity and entropy makes modern malware analysis a significant undertaking that takes considerable time, expertise and requires an extensive knowledge domain either in an individual or in coverage provided by a team of analysts.

Two fundamental techniques available to the analyst are static and dynamic analysis. Static analysis does not execute the code and the code is analysed via disassemblies, call graphs, searches for strings, library calls, and reconstruction of data structures, enumerations and unions within the code. This analysis technique is very time consuming and easily hindered by anti-forensics in the form of code obfuscation, packers and protectors which are increasingly being used by malware authors.

Dynamic analysis, in contrast, does run the code and the analyst observes its behaviour and interaction with the host and network via mechanisms such as registry, file and network monitoring tools. This technique is generally much easier to conduct than static analysis but is also easily hindered by malware that can detect the use of an emulation environment such as *VMware* (VMware, 2008) or the use of debugging tools such as *IDA Pro* (Hex-Rays, 2008). By detecting the use of these tools and environments, the malware can change its behaviour. Once detected, the malware can decide not to run its true payload and can run in a deceptive mode that makes it look like much less of a threat. It can delete itself together with any evidence, or if it is running with the appropriate privileges, damage or destroy the system that it is being run on or attached to (Smith & Quist, 2006). Zeltser (2007) uses an iterative and recursive technique that incorporates both the static and dynamic analysis techniques to extract the full functionality of the code in a recursive and iterative technique that spirals into the analysis from the higher level view to the more detailed view. This technique also facilitates the opportunity to discover and mitigate anti forensic techniques as the analysis process proceeds.

ANALYSIS PROCESS

A high level and simplistic view of the malware analysis process is depicted in figure 1 below. It shows malware as one of two inputs to the analysis methodology process which produces a report as an output. The generated results also feedback into the analysis methodology via an assessment process which can be used to adjust the methodology dynamically, or as a process improvement mechanism. Legal and ethical constraints serve as a bounding constraint to the process.

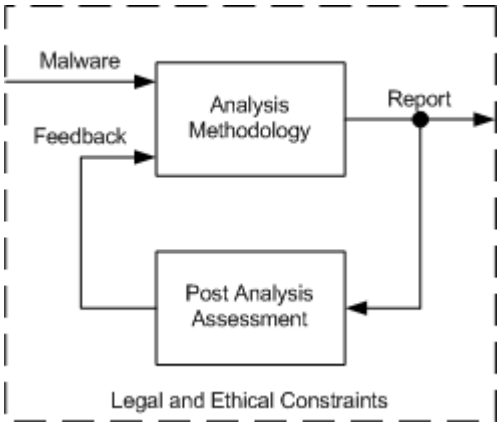

Figure 1 : Simplistic High Level View

An intermediate level view, as depicted in figure 2 below, must take additional constraints and inputs into consideration as well as the influence of a learning taxonomy. The malware can be collected from various sources and may have to be collected in a forensically sound manner. Collection can be influenced from the feedback mechanism of the post analysis assessment as well as from a learning taxonomy. The tools that are applied in the analysis may also have to be forensically acceptable and the analyst needs to understand their use and their limitations. The analyst will also require considerable cognitive abilities in malware ontology, programming, anti-forensics, environmental configurations and malware analysis. The learning taxonomy is central to the analysis process and both influences and is influenced by the collection process, analysis process and post analysis assessment. The MABOK traces directly to the learning taxonomy central to the malware analysis process.

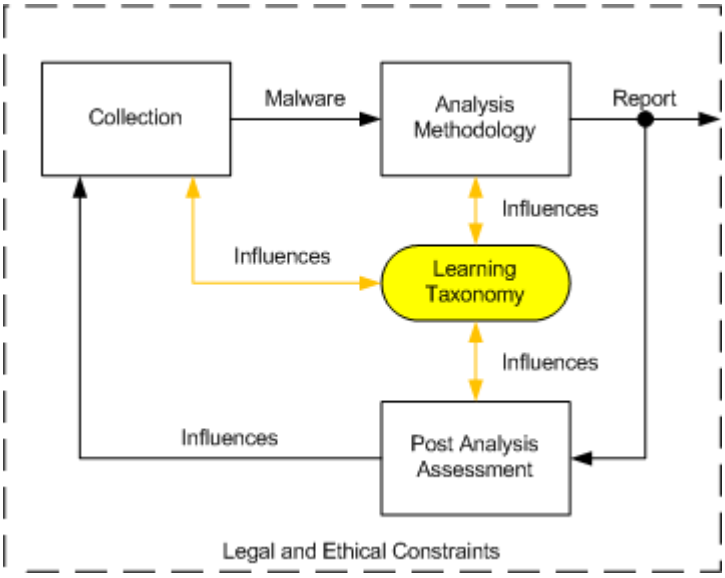

Figure 2 : Intermediate View

KNOWLEDGE DOMAIN

The knowledge domain proposed for the MABOK is presented in figure 3 below. The highest level stratum lists the categories of the MABOK whilst the branches beneath each category list the subject areas of each category in no particular order. The MABOK is a knowledge base required to perform effective malware analysis. The skill set represented in the MABOK may not reside within a single person but maybe reflected in a team.

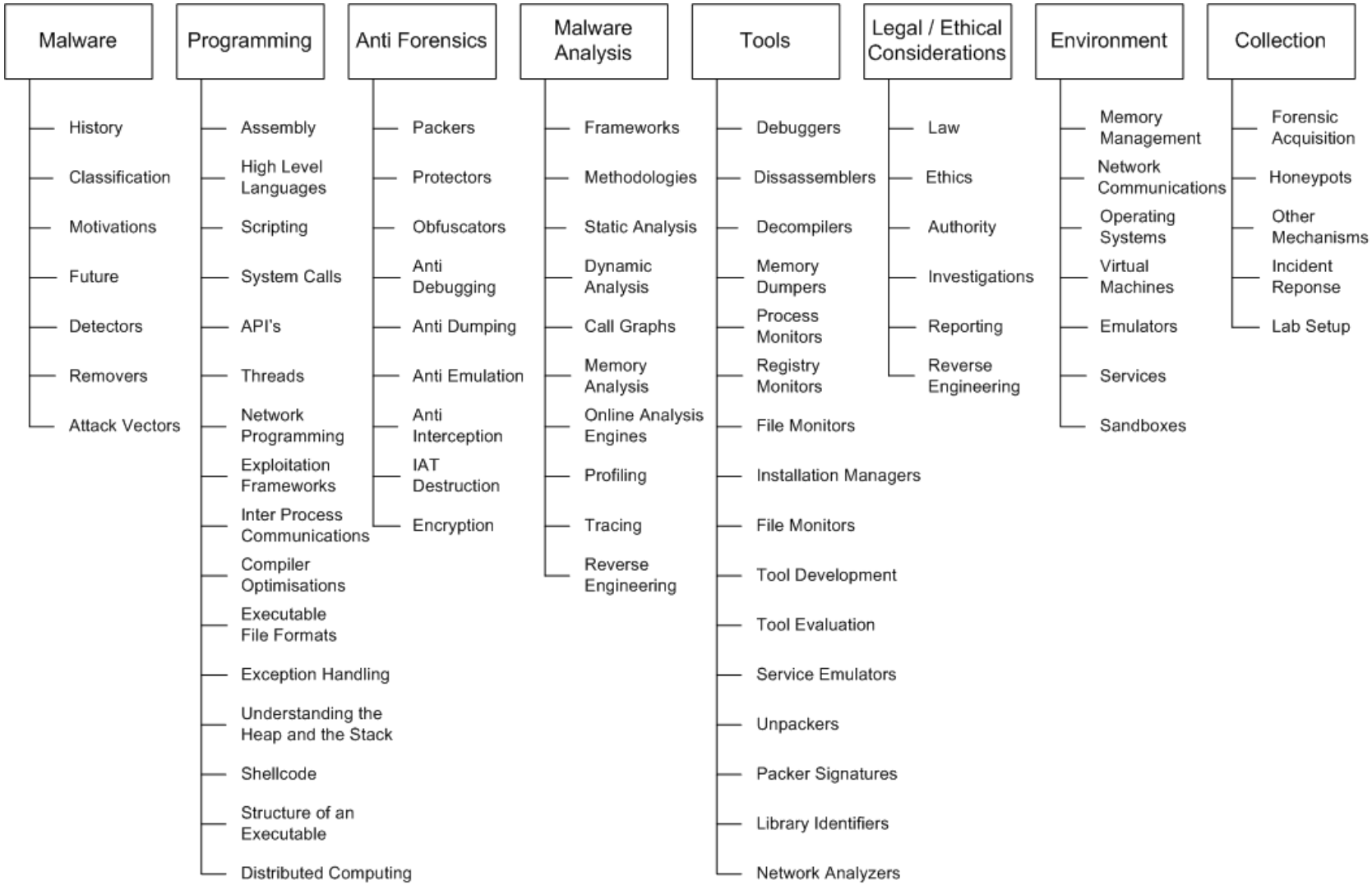

Figure 3 : Malware Analysis Body of Knowledge

Malware

Malware has an extensive history, and can be applied in a wide variety of malicious applications. It is important for the analyst to understand the motivations behind malware, including how it is applied and how it is detected. The effectiveness (or lack of) malware detection is also a very important consideration as well as detection techniques.

Programming

Programming skills are vital for in depth analysis of malware. Systems level programming, high level languages, scripting and even assembly language programming are important skills required to understand how malware is implemented and how it takes advantage of vulnerabilities. It is also an important skill set for the development of customised tools and for scripting disassemblers and debuggers. The poser of being able to script debuggers and disassemblers should not be underestimated in a malware analysis context. Many analysis tools now also allow additional functionality to be added by allowing users to write customised Dynamic Link Library (DLL) plugins or scripting languages such as *IDAPython* (Erdélyi, 2008) which integrates *IDA Pro* scripting with the *Python* (Python Software Foundation, 2008) scripting language.

Producers of malware also develop and utilise advanced programming techniques and technologies such as distributed computing to enable a competitive advantage over detection software and techniques. Therefore, it is imperative that a malware analyst also be well versed in cutting edge technologies and techniques.

Anti-Forensics

Often, static analysis of network based malware such as worms and bots cannot even start until the malware has been unpacked and the Original Entry Point (OEP) has been reached. Until then, the code appears as benign through obfuscation techniques incorporated into the malware by the authors of the code. Malware can also incorporate anti-debugging, anti-memory dumping, Import Address Table (IAT) destruction, anti-emulation and encryption to hinder the malware analyst. These techniques can be mitigated, but a thorough understanding of their use must be mastered to be able to achieve competent analysis of malware.

Malware Analysis

An adaptive, eclectic choice of techniques is required for analysis of malware. Various frameworks and methodologies such as static and dynamic analysis exist for the malware analyst to analyse malware such as *PaiMei* (Amini, 2008). Static analysis is the examination of source code logic and behaviours, whereas dynamic analysis is the monitoring and observation of the code as it executes. Both techniques have strengths. Obfuscation of code may render static analysis null and void. However, dynamic execution of that code segment may reveal the next code sections required for further static analysis. Other common software engineering techniques, such as profiling, tracing and debugging are also available, applicable and have utility in malware analysis. The diversity of malware modus operandi requires a range of approaches and techniques to perform successful dissection and analysis of the malware. The skills needed to perform competent analysis are profound, highly technical and are at the cutting edge of computer science.

Tools

A plethora of tools are available to the analyst including debuggers, disassemblers, de-compilers, memory dumpers, unpackers as well as many other tools common to the discipline of software engineering. All of these tools require niche expertise and a thorough understanding of the principles of their operation and the computers they execute on.

However, whether or not the tools are forensically sound and their use acceptable in a court of law is a matter that needs to be seriously considered. Some useful tools are available from hacking and software cracking sites that would not be considered forensically sound without considerable validation or black box testing.

Such tools could contain trojans and could easily hide a malicious purpose. They may not be forensically acceptable without significant due diligence on the part of the person or organisations using these types of tools. Other software cracking or reverse engineering sites have scripts for debuggers that can be easily and readily examined. These scripts are useful to extract the known algorithm for dealing with particular packers or to mitigate particular anti-forensic techniques used by creators of such software.

Legal/Ethical Considerations

Consideration must be given to legal and ethical issues in the analysis of malware. From a legal perspective, analysis of malware may require correct handling, preservation and presentation of evidence appropriate for a

court of law. It may also include consideration of disclosure of private data that has been uncovered during the course of analysis. This private data could include, but not limited to, usernames, passwords, and personal particulars such as date of birth, address, relationships and financial data such as credit card numbers or bank account details.

Furthermore, there is an ethical consideration to ensure that the analysis process is secure and that the spread of malware is not a possibility. Also in the case of extraction or identification of a particular new malware, the responsible sharing of this knowledge is also a matter for ethical consideration.

Environment

Analysis of malware will typically require configuring a complete virtual environment suitable for it to run in, not only from an operating systems perspective, but also the inclusion of network infrastructure and services. Modern malwares are increasingly network borne and network enabled. So it may be necessary to provide an environment in which the malware can utilize commonly used services such as Domain Name System (DNS) server, Simple Mail Transfer Protocol (SMTP) server or an Internet Relay Chat (IRC) server. Establishment of this style of environment allows for the malware initiating communications with these services to allow the dynamic capture of target data to assist in the dynamic analysis of malware. This type of environment may be supported by a virtualised environment using commercial virtualisation environments such as *VMWare* or *Virtual PC* (Microsoft, 2007).

It should be noted that because malware can contain the ability to detect these virtualised environments as a result of their hardware and software fingerprints, the ability to configure real systems and devices may need serious consideration. This will require the configuration of a particular computing host environment, or network device or other system administrative tasks in order to achieve this. This type of environment would need strict control and isolation to prevent the spread of malware.

Collection

Malware can be collected in a variety of ways. It can be forensically acquired during incident response, through a honeypot or through the course of research. The handling of the collected malware may see a need to be handled in a forensically acceptable manner to enable the malware to be admissible in court.

CONCLUSION

Malware analysis is becoming an important field of specialisation for forensic analysts. Authors of malware are becoming increasingly profit driven and are incorporating techniques to make their code as stealthy and undetectable as possible. Malware is being written by professional programmers who are very knowledgeable in their craft. They have a very good understanding of digital forensic methods and endeavour to make forensic analysis as difficult as possible.

The knowledge domain required to competently analyse malware is very broad. This paper has presented a brief introduction to a Malware Analysis Body of Knowledge that would be suitable for establishing a framework for competency development and assessment for the field of malware analysis and for incorporation into academic curricula. A learning taxonomy is central to the malware analysis process and eight domain areas were identified. These areas include malware, programming, anti-forensics, malware analysis, tools, legal and ethical considerations, environment and collection. The application of Blooms Taxonomy to this body of knowledge is the next phase of development.

REFERENCES

Amini, P. (2008). PaiMei.

Aycock, J. (2006). *Computer Viruses and Malware*. New York: Springer

Erdélyi, G. (2008). IDA Python.

Falliere, N. (2007). Windows Anti-Debug Reference. Retrieved October 1, 2007 from http://www.securityfocus.com/infocus/1893

Ferrie, P. (2008). *Anti-Unpacker Tricks*. Retrieved October 8, 2008 from http://www.datasecurity-event.com/uploads/unpackers.pdf

Hex-Rays. (2008). IDA Pro.

Kessler, G. (2007). *Anti-Forensics and the Digital Investigator*. Retrieved May 04, 2008, from http://scissec.scis.ecu.edu.au/conference_proceedings/2007/forensics/01_Kessler_Anti-Forensics.pdf

Larsson, L. (2007). *Meeting the Swedish Bank Hacker*. Retrieved April 14, 2007 from http://computersweden.idg.se/2.2683/1.93344

Masood, S. G. (2004). *Malware Analysis for Administrators*. Retrieved 17 March, 2007 from http://www.securityfocus.com/infocus/1780

Metasploit LLC. (2008). Metasploit.

Microsoft. (2007). Virtual PC.

Python Software Foundation. (2008). Python.

Rogers, M. (2006). *Panel session at CERIAS 2006 Information Security Symposium*. Retrieved October 8, 2008, from http://www.cerias.purdue.edu/symposium/2006/materials/pdfs/antiforensics.pdf

Smith, S., & Quist, D. (2006). Hacking Malware: Offense is the new Defense. Retrieved July 24, 2007 from http://www.offensivecomputing.net/dc14/valsmith__dquist_hacking_malware_us06.pdf

Symantec. (2008). *Symantec Global Internet Security Threat Report. Trends for July-December 07*. Retrieved October 8, 2008 from http://www.symantec.com/en/uk/business/theme.jsp?themeid=threatreport

Yason, M. (2007). The Art of Unpacking. Retrieved Feb 12, 2008 from https://www.blackhat.com/presentations/bh-usa-07/Yason/Whitepaper/bh-usa-07-yason-WP.pdf

VMware. (2008). VMware.

Zeltser, L. (2007). *Reverse Engineering Malware: Tools and Techniques Hands-On*. Bethesda: SANS Institute.

COPYRIGHT

Craig Valli and Murray Brand ©2008. The author/s assign Edith Cowan University a non-exclusive license to use this document for personal use provided that the article is used in full and this copyright statement is reproduced. Such documents may be published on the World Wide Web, CD-ROM, in printed form, and on mirror sites on the World Wide Web. The authors also grant a non-exclusive license to ECU to publish this document in full in the Conference Proceedings. Any other usage is prohibited without the express permission of the authors.