Feature Extraction and Static Analysis for Large-Scale Detection of Malware Types and Families

Lars Strande Grini

Master's Thesis
Master of Science in Information Security
30 ECTS
Department of Computer Science and Media Technology
Gjøvik University College, 2015

Avdeling for
informatikk og medieteknikk
Høgskolen i Gjøvik
Postboks 191
2802 Gjøvik

Department of Computer Science
and Media Technology
Gjøvik University College
Box 191
N-2802 Gjøvik
Norway

Feature Extraction and Static Analysis for Large-Scale Detection of Malware Types and Families

Lars Strande Grini

15/12/2015

Abstract

There exist different methods of identifying malware, and widespread method is the one found in almost every antivirus solution on the market today; the signature based approach. This approach uses a one-way cryptographic function to generate a unique hash of each file. Afterwards, each hash is checked against a database of hashes of known malware. This method provides close to none false positives, but this does also mean that this approach can only detect previously known malware, and will in many cases also provide a number of false negatives. Malware authors exploit this weakness in the way that they change a small part of the malicious code, and thereby changes the entire hash of the file, which then leaves the malicious code undetectable until the sample is discovered, analyzed and updated in the vendors database(s). In the light of this relatively easy mitigation for malware authors, it is clear that we need other ways to identify malware. The other two main approaches for this are static analysis and behavior based/dynamic analysis. The primary goal of such analysis and previous research has been focused around detecting whether a file is malicious or benign (binary classification). There has been comprehensive work in these fields the last few years. In the work we are proposing, we will leverage results from static analysis using machine learning methods, to distinguish malicious Windows executables. Not just benign/malicious as in many researches, but by malware family affiliation. To do this we will use a database consisting of about of 330.000 malicious executables. A challenge in this work will be the naming of the samples and families as different antivirus vendors labels samples with different names and follows no standard naming scheme. This is exemplified by e.g. the VirusTotal online scanner which scans a hash in 57 malware databases. For the static analysis we will use the VirusTotal scanner as well as an open source tool for analyzing portable executables, PEframe. The work performed in the thesis presents a novel approach to extract and construct features that can be used to make an estimation of which type and family a malicious file is an instance of, which can be useful for analysis and antivirus scanners. This contribution is novel because multinominal classification is applied to distinguish between different types and families.

Acknowledgements

I would like to express my greatest appreciation to my supervisors, Katrin Franke and Andrii Shalaginov for the extraordinary guidance and feedback through this project. Further, I will thank my classmates Espen, Lars, David and Martin for discussions, feedback and company during the entire process. Lastly, I would like to thank my current classmate Jan William, and my former classmate Simen for valuable discussions, feedback and proof reading of my work.

Contents

List of Figures

List of Tables

Abbreviations

ANN	Artificial Neural Network
API	Application Programming Interface
AUC	Area Under Curve
AV	AntiVirus
CPU	Central Processing Unit
DLL	Dynamic Link Library
FN	False Negative
FP	False Positive
JSON	JavaScript Object Notation
LOO	Leave-one-out
MBR	Master Boot Record
ML	Machine Learning
MLP	Multi-Layered Perceptron
OAA	One-against-all
OAO	One-against-one
OS	Operating System
PE	Portable Executable
PE	In the context of the feature names, we use *PE* at the beginning of the feature name to indicate that the feature was collected with the tool *PEframe*, e.g. *pe_api*.
ROC	Receiver Operating Characteristics
TN	True Negative
TP	True Positive
VT	VirusTotal

1 Introduction

This chapter provides a description of the topics covered by- and the problems targeted in the thesis, the motivation and justification for choosing this subject. Furthermore the proposed research questions are presented, as well as the planned contributions. Lastly the methodology and thesis outline is given.

1.1 Topics covered by project

Malicious software or malware are software that perform unwanted actions in the system on which it are installed. The McAfee malware zoo now includes 400,000,000 samples [19]. Malware authors pose a significant threat to every device connected to the Internet in terms of both privacy and economics. Before an antivirus vendor can protect against a unique type of malware, it has to be discovered and analyzed thoroughly before it is possible to update the malware definitions. And still, the end user will not be protected against this malware until he or she updates the malware definitions. To analyze malware samples there are two main approaches, static and dynamic analysis. Static analysis scans the file to collect data about the files behavior extracted from the content of the file, while dynamic analysis reveals information about files from running them in an isolated environment. In addition, as a result of obfuscation, different methods utilized by malware to avoid detection, some dynamic analysis methods requires user interaction to trigger the malicious behavior. As a result, dynamic analysis is more time consuming than static analysis methods. From this we considered that static analysis is more appropriate for our project considering the large number of samples and the time restraint.

1.2 Keywords

Machine Learning, Pattern Recognition, Digital Forensics, Malware Detection, Static Analysis

1.3 Problem description

Antivirus vendors implement signature-based matching to detect malware. This is easily avoidable from a malware authour's position, when a small change in the code will lead to a completely different hash of the same malware. Other work on using machine learning/pattern recognition done to classify malware has focused on the binary classification problem, i.e. classify samples as malicious or benign. A problem with the major part of application of machine learning/data mining techniques to classification are mainly that the data set consists of a rather low number samples. Even though other research produce good results such as low false positive rate and high accuracy, one can assume, but not know that these results will scale when analyzing a larger pool of malware. Such analysis needs to be made as there are discovered hundreds of thousands of new malware samples each day. The McAfee Labs Malware Zoo grew by about 45,000,000 malware samples in Q2 2015 [19]. A limitation of previous work is also that the majority focuses on detection, determining if an inspected file is either malicious or benign, making our work a novel contribution to the field of malware analysis. We have limited

our research to focus on Windows executables exclusively, due to the popularity of the Windows platforms.

1.4 Justification, motivation and benefits

With the ever increasing volume of different families and types of malware, new approaches to the fields of malware analysis and malware detection can be automated and in theory be able to detect new malware before the antivirus vendors. If possible, malware family classification from numeric features extracted from static analysis would be a great improvement over the most popular antivirus solutions which almost exclusively utilize signature based recognition. This will also have the potential to speed up the work of malware analysts and other security vendors, as most other similar projects focuses on binary classification [20], and not type or malware family. For security professionals and other people responsible for security in a corporation, there is crucial to know more about a malware than just the fact that a file probably is malicious, to be better suited to mitigate the threat that it poses. There exists no known solution to perform this as of today. This also gives preliminary threats indicator or attack indicators for malware analysts and antivirus vendors.

1.5 Research questions

1. What features that can extracted from the static analysis tools *PEframe* and *VirusTotal*, will be most relevant for distinguishing malware into type and family?
2. What accuracy can be achieved with features derived from the static analysis tools *PEframe* and *VirusTotal*?
3. Which methods for feature selection and classification performs the best on the features constructed?

1.6 Planned contributions

This thesis aims to perform malware classification in the form of classifying samples into both type of family and type, rather than conventional binary classification (malicious/benign). If we can achieve good performance, we are also able to say something about which features that are most important for such classification. To the author's knowledge, this is per end of 2015 a novel approach, and there were no major work related to large scale static malware analysis according to published peer-reviewed journal articles.

1.7 Thesis outline

The thesis is divided into several chapters, and this sections provides a brief description of each chapter.

- Chapter 2 presents state of the art and definitions from published literature regarding malware analysis before discussing a taxonomy of the most general/high level types of malware. Further, we discuss the most common detection techniques utilized by antivirus scanners. Thereafter, we present the problem of malware naming before an overview of the portable executable format is given. We conclude by discussing obfuscation techniques used by malware.

- Chapter 3 provides an overview of the field of machine learning and pattern recognition, which includes the machine learning process, and common challenges to take into account when performing analysis using machine learning.
- Chapter 4 gives an overview of existing research related to our research in detection of types and family of malware.
- Chapter 5 provides an overview of the methodology used in the practical part of the thesis. It includes a description of the data acquisition, feature construction, and a discussion of the different methods that will be used for feature selection and classification.
- Chapter 6 presents the experiments performed, technical specification from the system on which the experiments were conducted, as well as the software requirements necessary to perform the experiments. Further the results from feature selection and classification as well as discussion of the results is given.
- Chapter 7 sums up the thesis and the most important findings, before concluding the thesis with a discussion of theoretical implications and practical considerations and proposals of future work.

2 Malware: Taxonomy, Analysis & Detection

This chapter wil provide a state of the art and discussion of malware analysis.

2.1 Methods for malware analysis

Most antivirus scanners use signature-based and heuristic-based detection methods, where they search for known patterns in executable code as well as the hash of the file against known malicious files against a database. A limitation of signature-based methods are that the malware must be obtained and analyzed before the antivirus vendors can update their databases [21, 16]. In this section we will provide an overview of the general methods for malware analysis and detection.

2.1.1 Detection and Analysis

Static analysis is a term that covers a range of techniques on how to dissect a malware sample and gather as much information as possible without executing the file. A commonly used tool is the *VirusTotal* online scanner[1] [22] which provides output from different scanning tools, and also checks the submitted sample against 65 antivirus databases and then gives the detection ratio as well as the names that each vendor has labeled the sample with. Another useful tool is the *strings* command, which returns strings of unicode and ASCII within the file with length of 3 characters or more. Depending on if/how the functionality of the file is obfuscated, one can see method calls, commands, filenames, resources accessed and IP addresses, which can yield suspicious behavior.

In addition, the *PEframe* Python script[2] [22] is able to detect packers, anti-debug and anti-VM techniques as well as url's and filenames used by the sample. While *PEframe* and *strings* are somewhat redundant, the different tools can yield different information as well; while strings dumps everything, *PEframe* structures all the data.

Reverse engineering is a growing discipline to perform this dissection thoroughly in terms of using software to generate the assembly instructions to then be able to determine the actions that the inspected sample performs on a system [23]. We will however not use this approach in our work.

Due to the increasingly used and more complex obfuscation techniques, static malware analysis are becoming increasingly difficult to perform [24, 25]. The biggest advantage of static analysis methods however, are that they considerably quicker, making it appropriate in large-scale analysis.

2.1.2 Dynamic Analysis

Dynamic analysis is the method of running and monitoring a malware in a secure environment, usually sandboxed and/or virtualized, to monitor its behavior [23, 25]. This approach differs from the static approach, most importantly, by that the malware specimen is executed. With the use of system snapshots and system monitoring tools, it is possible to determine the actual behavior of the file.

[1] https://www.virustotal.com
[2] https://github.com/guelfoweb/peframe

There are, however, a trade-off between using a virtualized and a physical host. On a virtualized host, the setup and re-imaging is a quicker process than on a physical host. There exists techniques for malware to identify that it is being executed on a virtual host, and are thus capable of changing the behavior to avoid detection from dynamic analysis. Some malware types are also capable to infect the host on which the virtual environment is run [23]. With respect to this, it might be beneficial to perform the analysis on a physical host.

2.2 Malware Taxonomy

This section provides an overview of the most common types of malware. As we will discuss later in the thesis, there exists several more types than the ones mentioned in this section. These however, can arguably be characterized as subclasses of the types mentioned.

2.2.1 Virus

A virus is a file that inserts itself into one or more files to perform one or more actions. A virus infection usually consists of two phases; insertion and execution [26].

Boot Sector Virus

This type of virus infects the master boot record to be able to run alongside the host operating system. This is achieved by that the virus makes a copy of the MBR, the first section of the hard drive that includes the identifier of where the operating system is located, so that when the system is booted, the virus is run first. Then the MBR is executed, so that the virus can control execution [26].

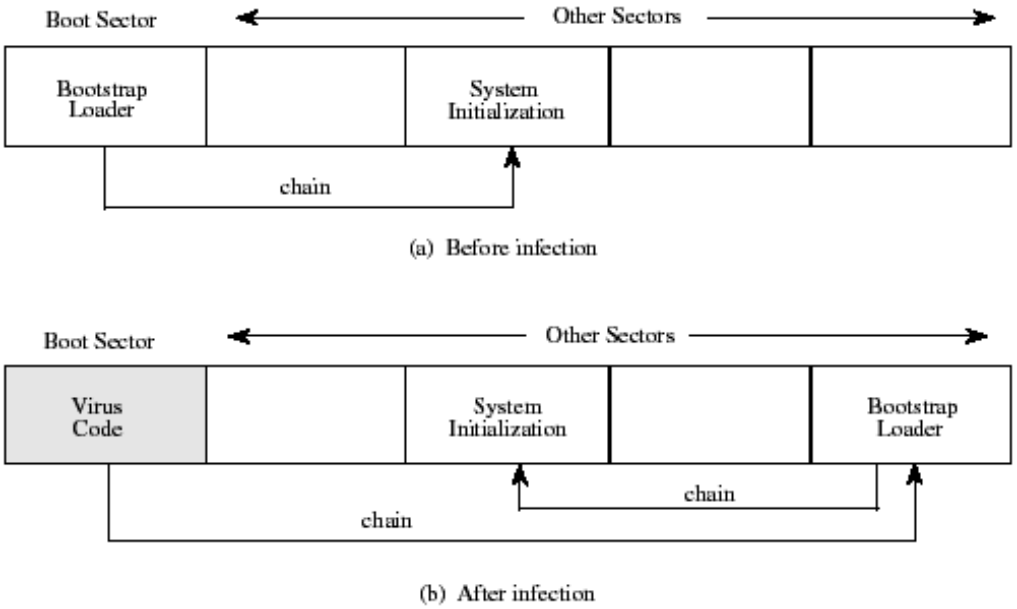


Figure 1: Illustration of boot sector virus [1]

Executable

An executable virus infects executable programs. This is commonly achieved by inserting itself into the file right after the file header, so that the payload is run first when the file is executed [26].

Multipartite

A multipartite virus is a combination of the two former types, i.e. a virus that can infect boot sectors or programs [26].

TSR Virus

Abbreviation for Terminate and Stay Resident. Such a virus is able to stay resident in memory after the infected application is terminated, or when bootstrapping is done (in case of boot sector infection) [26].

Encrypted

An encrypted virus encrypts the malicious payload in an attempt to avoid antivirus detection. The only clear text code is the decryption routine [26].

Polymorphic

A polymorphic virus changes its form each time it infects another program. In the case of an encrypted virus, the antivirus scanner was ultimately able to detect these when recognizing the decryption routine. The polymorphic virus use another way to conceal itself from antivirus scanners by the means that the decryption routine was changed from each new infection [26].

Macro

A macro virus is a set of interpreted instructions. Usually within a word, excel or pdf file. Especially utilized in Microsoft Office 97, but execution of macros have been disabled by default in later versions [26].

2.2.2 Worm

A worm is a malware that is able to propagate to other hosts [26], and actively seeks to do this by exploiting software or network vulnerabilities. Other propagation methods includes email, file sharing and removable USB-drives [27].

2.2.3 Trojan

A Trojan Horse is a malware that presents itself as a file or a program with legitimate purposes while having concealed malicious behavior as well [26], e.g. password stealing. Often used to capture and send various information from the infected host. It can also be used as a part of an attack such that the trojan, when executed downloads another piece of malware, for example infecting the host to turn in into a bot [25].

2.2.4 Backdoor

A backdoor is an entry point to a system that allows someone aware of this to gain access to a system or program without having to go through the standard security procedures. Backdoors are commonly used by programmers as well, that are utilized to debug software and where special privileges are in need. This means that a backdoor is not necessarily malicious by nature, but it is malicious when people use it for malicious purposes [27].

> *"During the development of Multics, penetration tests were conducted by an Air Force "tiger team" (simulating adversaries). One tactic employed was to send a bogus operating system update to a site running Multics. The update contained a Trojan horse that could be activated by a backdoor and that allowed the tiger team to gain access. The threat was so well implemented that the Multics developers could not find it, even after they were informed of its presence."* [27].

2.2.5 Rootkit

Rootkits are tools that are able to alter the host's behavior to remain hidden in the system while enabling an attacker to have all privileges on the infected host. The administrator account on unix systems are called "root", which has lead to the name of this type of malware [27]. Rootkits can be divided into different groups, depending on their specific behavior:

> ***Persistent**: Activates each time the system boots. The rootkit must store code in a persistent store, such as the registry or file system, and configure a method by which the code executes without user intervention. This means it is easier to detect, as the copy in persistent storage can potentially be scanned [27].*
>
> ***Memory based**: Has no persistent code and therefore cannot survive a reboot. However, because it is only in memory, it can be harder to detect [27].*
>
> ***User mode**: Intercepts calls to APIs (application program interfaces) and modifies returned results. For example, when an application performs a directory listing, the return results don't include entries identifying the files associated with the rootkit [27].*
>
> ***Kernel mode**: Can intercept calls to native APIs in kernel mode. The rootkit can also hide the presence of a malware process by removing it from the kernel's list of active processes [27].*
>
> ***Virtual machine based**: This type of rootkit installs a lightweight virtual machine monitor, and then runs the operating system in a virtual machine above it. The rootkit can then transparently intercept and modify states and events occurring in the virtualized system [27].*
>
> ***External mode**: The malware is located outside the normal operation mode of the targeted system, in BIOS or system management mode, where it can directly access hardware [27].*

2.2.6 Bot

Bot, short for robot, is a type of malware that usually makes the infected host a part of a botnet. A botnet is a network consisting of other infected hosts which a botmaster can control to perform a variety of malicious actions, such as distributed denial of service attacks, click fraud or bitcoin mining [28]. The last few years, botnets have become more business-oriented in the way that there are possible for "everyone" to rent a botnet, or a part of a botnet, enabling the "customer" to perform these malicious actions by him-/herself [29]. Other utilizations of infected hosts are the usage of keyloggers to collect user credentials, banking information and credit card details and email addresses from the bots, but these capabilities are typically not part of the bot itself. Instead this is achieved by downloading a specific trojan horse to perform the wanted actions. Originally, *bot*-malware was named "remote access trojan horses" [30].

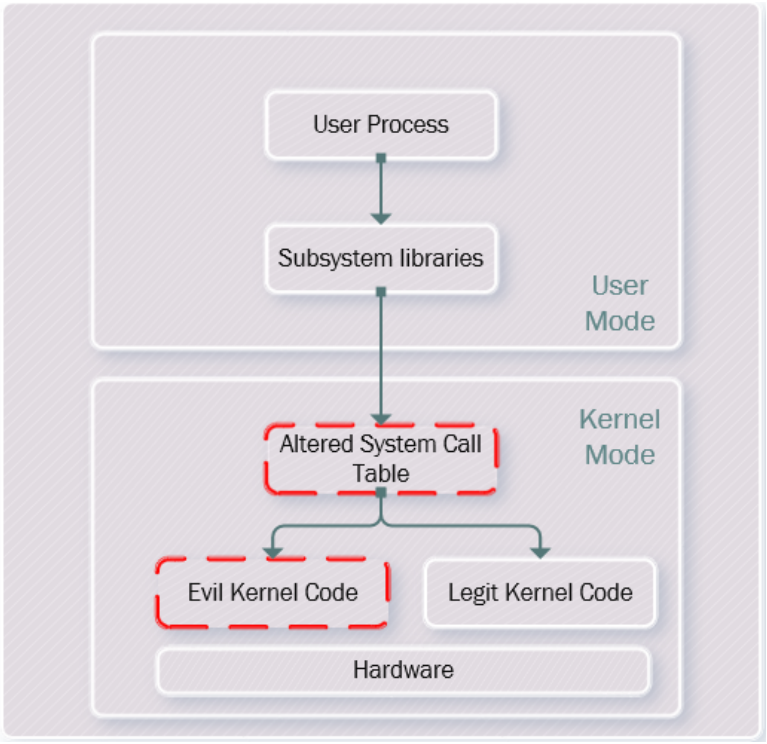


Figure 2: Kernel mode rootkit [2]

2.3 Malware detection in antivirus scanners

To combat the threat of malware to end-users, private persons and companies, the prevention, detection and removal of malware has grown to be a large business, in the form of antivirus solutions. Each of the different main detection techniques have advantages and disadvantages and are thus used in conjunction with each other in most antivirus software today [31]. In this section, we will discuss how most of these solutions perform malware detection.

2.3.1 Signature based

Vinod et al. defines a malware detector as a "function whose domain and range are the set of an executable program, and the set {malicious, benign}" [32]. In other words, a tool to determine if the scanned file is malicious or benign. Malware scanners use signatures to detect known bad files or code segments in file. An example of a signature can be the binary pattern of the machine code. Antivirus scanners includes a database with all known signatures of malicious content, which the signature of the scanned files are compared against to determine if the file is malicious or benign [32].

2.3.2 Anomaly based

Another common detection technique implemented by malware scanners is the anomaly based technique. In contrast to signature-based which works in the principle of blacklisting known bad signatures, anomaly-based detection works by the whitelist principle. In other words, the antivirus vendor has a database of specification for known-good. If the inspected file breaks one or more of these rules, the file is labeled as malicious [32].

2.3.3 Heuristic based

The latest model for malware detection is called *Heuristic Based*, and refers to the methods of applying machine learning methods to the file to learn the behavior [31]. Different feature sets used for heuristic based analysis includes according to Bazrafshan et al. [31]: API calls, Opcodes and control flow graphs.

> "Most heuristics methods are based on feature extraction. The antivirus engine extracts static features, such as file size or number of sections, or dynamic features based on behaviour. Classification of the code as either malware or benign is then made based on which features the sample possesses. In more traditional heuristic methods an antivirus analyst creates either rules (e.g. if target has feature 1 and feature 2 then it is malicious) or thresholds (e.g. if target has more than 10 features it is malicious)" [33].

2.4 Obfuscation Techniques

'Obfuscation' is defined by the Oxford Dictionary[3] as "*The action of making something obscure, unclear, or unintelligible*". Obfuscation, in respect to malware, are thus techniques malware authours utilize to make the malware harder to detect or undetectable by antivirus scanners [34]. In this chapter we provide an overview of the most common techniques.

Obfuscation in malware is a result of the never ending war between antivirus vendors and malware authors. Since the time of the internet and the first antivirus solutions, the malware authors began to adapt their habits to bypass detection by antivirus scanners. This in turn, forces the antivirus providers to adapt as well.

2.4.1 Encryption

Obfuscation through encryption is achieved by that the malicious code is encrypted in the file. The only segment unencrypted are the decryption routine. Every time the file is run, the decryption routine recovers the original code. This technique is also implemented such that the malicious code is encrypted with a different key for each new infection. As mentioned, this requires that the decryption routine is unencrypted and will therefore not change, which thus makes the malicious code detectable [4, 35]. XORing has been used widely as encryption in malware because it is practical for the author, since XORing with the same value twice will produce the initial value, so that the author don't have to implement separate algorithms for encryption and decryption. And even though this encryption is cryptographically weak, antivirus scanners did not have the same range of tools to detect and identify malware in the mid 90's, which meant that files with similar decryption and yet completely different functionality was indistinguishable from the decryption routine alone. In addition, similar decryption routines can be found in benign files as well [3].

Figure 3 illustrates a countermeasure against detection of encryption routines; changing the direction of the encryption/decryption loop, and changing the order of what are being encrypted/decrypted to make the process non-linear.

2.4.2 Polymorphism

The evolution of encryption that is able to construct a large number of distinct decryptors, and thus avoid detection. In 1992, a toolkit named "The Mutation Engine" was developed, which made it possible for malware authors to mutate an unencrypted malware into a

[3] http://www.oxforddictionaries.com/definition/english/obfuscation

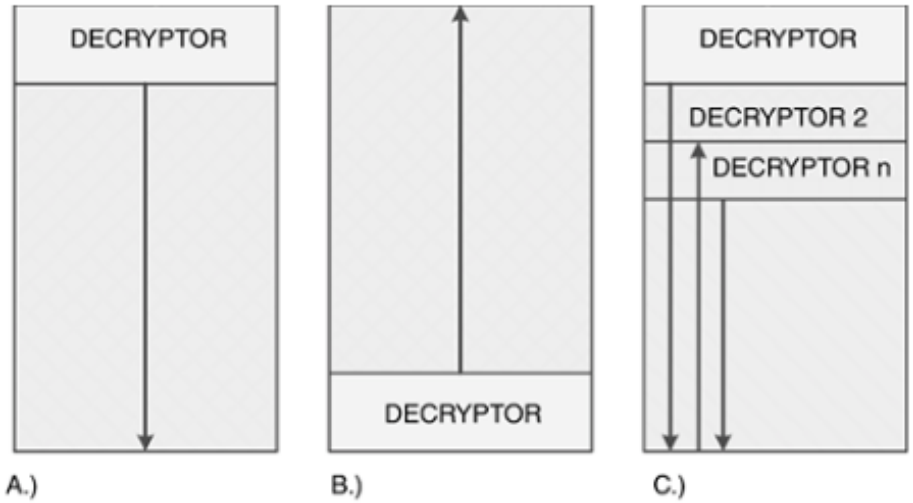


Figure 3: Encrypted malware [3]

highly polymorphic sample which, at the time, was undetectable for antivirus scanners [36, 4]. Polymorphism ensures that the decryption routine is changed from each new infection, which means that signature-based detection was a hard task at the time for antivirus scanners.

To counteract this, antivirus scanners made use of "sandboxing", execution in a secure environment, to enable detection for polymorphed malware. This is possible because, even though the decryption routine changes from each infection, the virus body remains intact. Thus, by running the malware in a sandbox, the program is loaded into memory, and signature based detection can be applied on the constant virus body [4].

2.4.3 Metamorphism

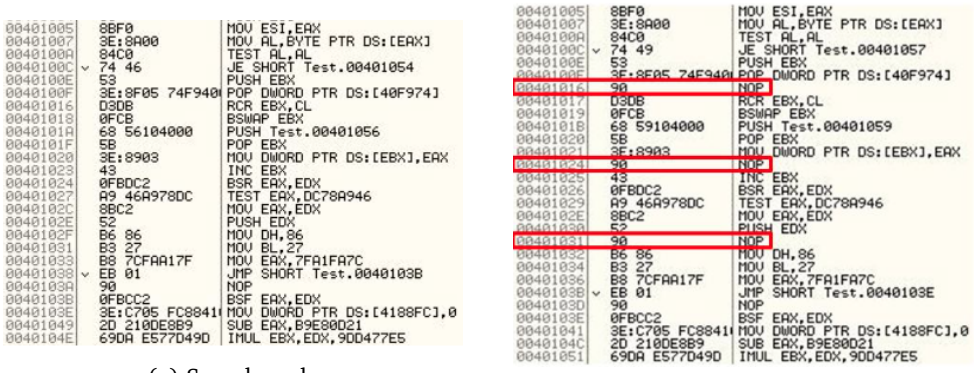
The next step in the evolution of malware obfuscation is metamorphism. As mentioned in the previous sections, antivirus scanners were enabled to detect first the decryptors (encryption), and then the constant virus body (polymorphism) [4]. Metamorphism refers to the techniques implemented to make the virus body itself change from each new infection. That is, changing the code, so that it looks different, but performs the same actions [4]. This techniques would then harden the challenge for antivirus detection once again, as the approaches to antivirus detection with sandboxing and signature detection could not detect the ever changing malicious code.

2.4.4 Specific obfuscation techniques

In the previous sections, we have discussed different obfuscation techniques in a general matter. This section provides a discussion of some of the most common, specific techniques that are implemented by malware authors to make malware polymorphic and metamorphic.

2.4.5 Dead-Code Insertion

To alter the code itself, but to keep the functionality, *dead-code insertion* makes use of simple instructions that does nothing, thus the name *dead-code*. A simple way to achieve this is by inserting NOP instructions in arbitrary location in the code. The NOP operator "performs an operation without behavior" [37].



(a) Sample code

(b) Sample code with dead-code insertion

Figure 4: Example of dead-code insertion [4]

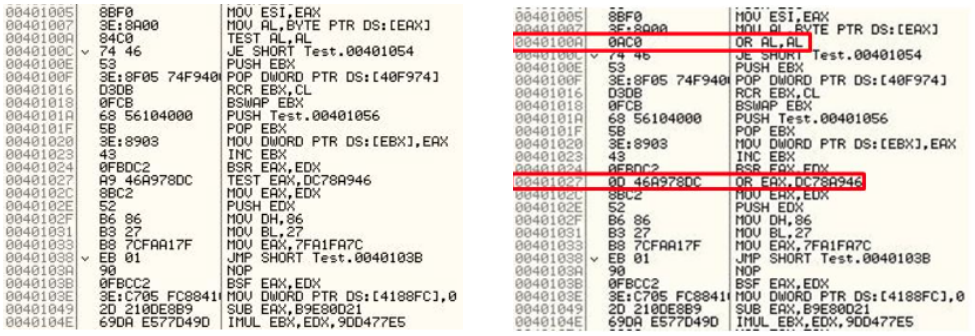
2.4.6 Register Reassignment

> *"Register reassignment refers to the change of registers used by live variables. If a particular register R1 is not used during the live range of a variable, then the register R2 used currently to store the live variable can be replaced by R1" [34].*

This technique will thus alter the code, but keeping the functionality, which means that register reassignment would be resistant to signature based detection, but not anomaly-based detection.

2.4.7 Instruction Substitution

In programming, there will be different commands that will achieve the same result. Instruction substitution is the method of replacing instructions with different but equivalent ones to alter the code and keep the functionality [4].



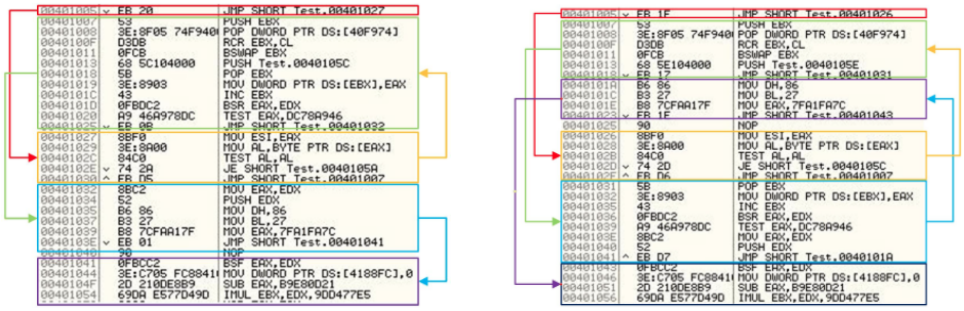
(a) Sample code

(b) Sample code with substituted instructions.

Figure 5: Example of instruction substitution [4]

2.4.8 Code Transposition

Code transposition refers to changing the instruction order, with jump-instructions so that the instructions are not executed in descending order. I.e. a "cosmetic movement of code within a file" [34].



(a) Conditional transposition

(b) Independent code transposition

Figure 6: Example of code transposition [4]

2.5 Windows Portable Executables

According to *netmarketshare*[4], Microsoft Windows Operating Systems have a market share of 90.41% (October 2015). The entire OS distribution for October 2015 can be seen in Table 1. Although we cannot validate the precision of these measurements, this indicates that Windows Operating systems are the dominating operating system. Due to this, we have chosen to analyze Windows executables exclusively in this project, namely *Portable Executables*(PE). PE is an overarching term that includes several file types; acm, .ax, .cpl, .dll, .drv, .efi, .exe, .mui, .ocx, .scr, .sys, .tsp [38, 39].

Operating System	Total Market Share
Windows 7	55.71%
Windows XP	11.68%
Windows 8.1	10.68%
Windows 10	7.94%
Mac OS X 10.10	3.45%
Windows 8	2.54%
Mac OS X 10.11	2.18%
Windows Vista	1.74%
Linux	1.57%
Mac OS X 10.9	1.10%
Mac OS X 10.6	0.45%
Mac OS X 10.7	0.37%
Mac OS X 10.8	0.35%
Windows NT	0.11%
Mac OS X 10.5	0.07%
Mac OS X 10.4	0.02%
Windows 2000	0.01%
Mac OS X (no version reported)	0.01%
FreeBSD	0.00%

Table 1: Operating System distribution

Windows Portable Executables consists of different sections, which are explained in this section. Figure 7 illustrates the PE format.

[4]https://www.netmarketshare.com

MS-DOS MZ header

The MS-DOS MZ header, the first 64 bytes of the file, contains information about the file's compatibility, basically whether the file is a DOS-program or not [40]. The least field of this header also contains a 4-byte offset in the file, which is necessary to locate the PE-header [38].

PE header

The PE header contains information about the remaining sections, such as location and size of the remaining sections and timestamps [40, 5].

PE optional header

The optional header includes 224 bytes in the file, and is, contrary to its name, not optional [38, 40]. This section contains the operating system the file is intended for and initial stack size [38].

Section header

This section includes the section name, raw size and virtual size for each header [5].

Section data

The last header in a PE-file is called section data, and contains the file's original entry point, and the location in the program when execution of code starts [5].
According to the linux *file*-command, the PE files are defined as "PE32 executable (GUI) Intel 80386, for MS Windows". Due to the popularity of PE32 (and PE32+) files, we will concentrate on this. An entire distribution of the filetype definition can be found in Appendix E.3.

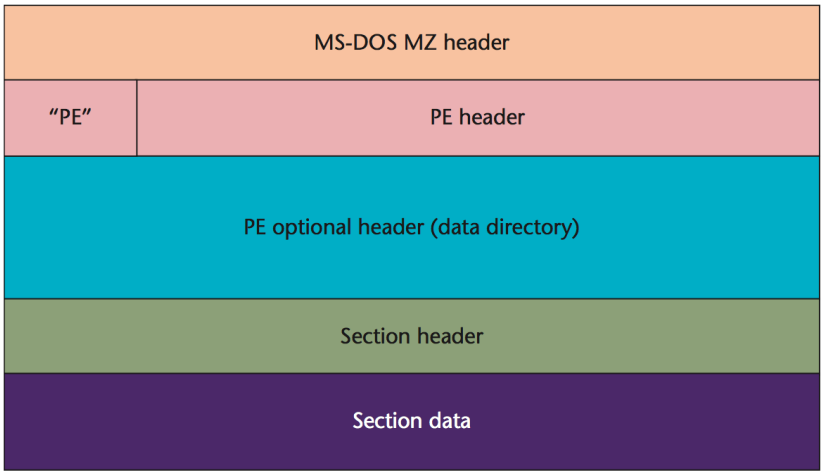


Figure 7: The portable executable format [5]

2.6 Naming of malware

In the world of malware, evil is known by many names, which is easily demonstrated by analyzing an arbitrary hash checksum with *VirusTotal*[5]. Each vendor has its own naming scheme, as seen in Table 2. The research questions stated in the introduction demands that we follow a strict methodology for labeling the collected samples in a consistent way.

[5]https://www.virustotal.com

AntiVirus Vendor	*Malware*
ALYac	Worm.Sql.Slammer.Dump.A
AVG	SQLSlammer
Ad-Aware	Worm.Sql.Slammer.Dump.A
Agnitum	Win32.SQLExp.A
Avast	Win32:SQLSlammer
BitDefender	Worm.Sql.Slammer.Dump.A
Microsoft	Worm:Win32/SQLSlammer.remnants

Table 2: Example of different names for the Slammer worm

In 1991 the Computer Antivirus Research Organization (CARO) proposed a standardized naming scheme for malware [41]. Although CARO states that this naming scheme is "*widely accepted*", we found that from all the vendors on *VirusTotal*, apparently Microsoft is the only one that complies to this. As a result, it is challenging to establish common pattern in results across antivirus databases.

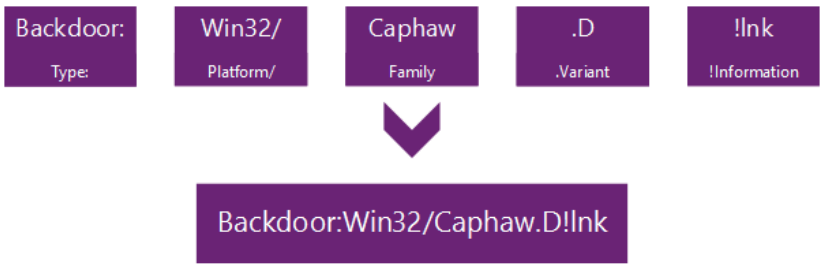


Figure 8: Implementation of the CARO naming scheme [6]

Type

The keyword that describes the general functionality of the malware, e.g. worm, password stealer, backdoors. A complete list of all malware types included in the data used in this project is provided in Appendix E.1.

Platform

This includes, at minimum, the platform the malware requires to run on, but can also include interpreter, code language and file formats [6, 41].

Family

A family defines a group of threats that are similar in one or more ways; for instance different variants of the same malware, different malware that targets the same exploit or type of credentials to steal. Often, a name is given to a new malware by the malware author, which then may be used as the family name. However, as the family name is open to "artistic license", this is also the hardest one for antivirus providers to agree on [41].

Variant

This field is increased one letter at a time, as a malware discovered is found to be a newer version of a previously detected malware.

Additional Information

This field is optional, and includes information regarding files or components used by the threat in relation to other threats. "In the example above, the *!lnk* indicates that the threat is a shortcut file used by the *Trojan:Win32/Reveton.T* variant, as shortcut files usually use the extension *.lnk*." [6].

3 Machine Learning & Pattern Recognition

Machine learning is a statistical subfield of artificial intelligence which has become increasingly utilized over the past years [42, 13]. The basic property of machine learning is to learn a model from labeled or unlabeled data to be able to distinguish new, unseen data. This section will provide an overview of the basic properties of machine learning. Bishop [42] defines machine learning as "*the automated detection of meaningful patterns in data*". The goal of machine learning can then be described as the process of training a data model to be able to make decisions itself, or to predict the future based on the past. This is achieved either by grouping similar data points together (clustering) or to label a data point as a or b, where a and b are the classes (two-class classification). How this is achieved will be further explained in this chapter.

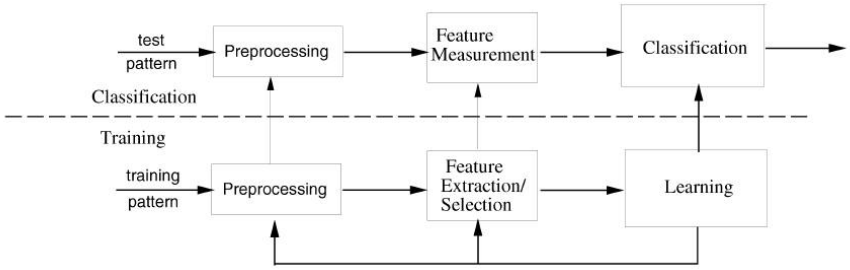


Figure 9: The machine learning process [7]

3.1 Preprocessing

Preprocessing refers to the methods to prepare the data and transform raw measurements for analysis. Different methods for this includes [13]:

- *Transforming continuous attributes into discrete* - necessary when using an algorithm that only is able to process discrete values.
- *Transforming discrete attributes into continuous* - necessary when using an algorithm that only is able to process continuous values.
- *Transforming numerical attributes into nominal* - necessary when using certain classifiers, e.g. Naive Bayes.
- *Accounting for missing value in the data* - in real world problems, we do not always have the luxury of having all data. To account for this, there exist different approaches, e.g. to ignore the missing values completely, or to set them as the most probable value.
- *Visualization* - humans are able to process visual information quicker than numerical information. This means that visualization can be useful to gain an understanding of the data before further analysis is performed.
- *Feature construction* - construction of features from raw data.

3.2 Feature Selection

Feature selection refers to selecting the optimal subset of features. A data set can contain an arbitrary number of features where different features will contribute in different degrees to the classification. Selecting the optimal subset of features is therefore an important step before further analysis should be performed. This is especially necessary when there is a large number of both features (dimensions) and/or samples. Proper feature selection will contribute to higher classification accuracy and reduce the computational complexity, which in turn will increase the overall classification performance and reduce computational time [13]. Reducing the number of features can also contribute to avoid, or reduce the problem of *overfitting*, which will be further discussed in Section 3.4.3.

> *Feature selection is necessary when there are too many attributes or the set of attributes consists of irrelevant, random, redundant or correlated attributes that may degrade the learning performance* [13].

Methods for feature selection can be divided into two categories; *filter* and *wrapper* methods [13]. *Filter* methods are usually the quickest, and work in the way that every feature in the data set is ranked according to their influence to classification, in descending order. The number of features that should be selected can either be determined on forehand, or the algorithm can be set to select features that exceeds a threshold. In *wrapper* methods, machine learning methods are used alongside cross validation, and requires as a result of this, more computational time. Even though *wrapper* methods takes more time to perform, there is no way to know beforehand that these or other methods for feature selection will perform better than another one. This principle is called "*No free lunch*", which we will discuss in Section 3.4.1.

3.3 Learning

Learning refers to adjusting parameters of a model according to predefined algorithm from training data. This process is also known as *training*. Learning algorithms can be divided into two groups, namely *supervised* and *unsupervised*. In *supervised* learning, we have the labels for each sample in the data set on forehand. The most common application of supervised learning is *classification* [13], which means to label all samples from the learned model, and then compare the results of the classification to the actual labels. Algorithms with these capabilities are referred to as *classification* algorithms, which are

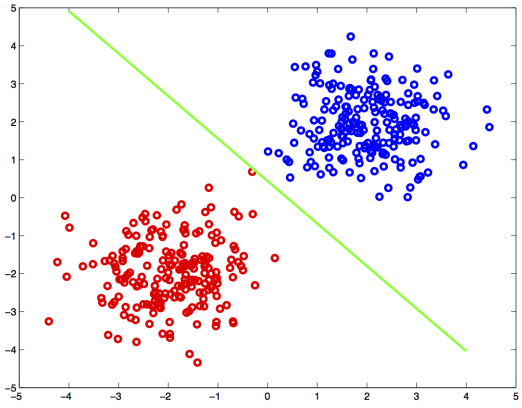


Figure 10: Linearly separable data [8]

a subset of *supervised learning* algorithms.

Figure 10 illustrates linearly separable data in a two-dimensional feature space. In this figure we can clearly see that the data points are linearly separable. The figure also touches upon the main principle about a commonly used classifier, Support Vector Machines (SVM). The objective of this method is to find the optimal line for separation, a hyperplane, for separating the data. Note that this example is simplified, and in many real world applications the data will not be linearly separable. When dealing with data that are not linearly separable, many algorithms can be run with other settings to account for this. In the case of *SVM*, this algorithm can also be utilized to utilize a polynomial hyperplane to facilitate classification. This method will in many cases be necessary, but requires more computational time, and is more prone to *overfitting*, which will be explained later in this chapter. We also usually have a larger feature space than two, three or four, making visualization difficult.

Clustering and unsupervised learning are often used interchangeably in literature, and means to group samples with similar features together. Actually, clustering is a group of algorithms under unsupervised learning methods, and in unsupervised learning, we do not have the actual labels of the samples, and the idea is thus to group similar data together.

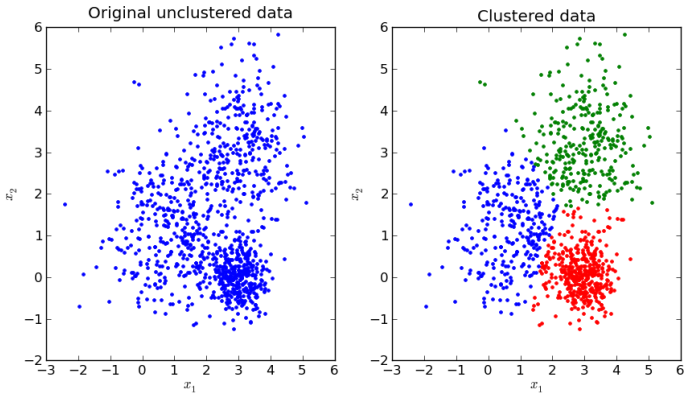


Figure 11: K-Means clustering [9]

Figure 11 illustrates the basic idea of *clustering*, that is grouping similar data points together. The *K-Means* algorithm works in the way that a number of clusters, k, is set initially. Samples are then assigned to the cluster with the nearest center point (centroid). When all the data points are *clustered*, the centroids for each clusters are updated by computing the average of all data points in the cluster. This process is repeated until there are no changes in assignment [13]. In respect to choosing the number of clusters, k, it is impossible to know in advance what would be the best value. This means that the *no free lunch* principle applies to this process as well, and will be further explained in Section 3.4.1.

Discussion

Up until this point, we have discussed the key properties of machine learning aspects. We have used two simple examples to explain the principle of learning. *Classification* and *clustering* are the most popular utilizations of *supervised* and *unsupervised* learning,

respectively [13]. The reason for using these two algorithms as examples, is that both are easy to understand and illustrated in a two-dimensional feature space.

Multinominal classification

A problem related to machine learning not yet discussed in the thesis, is the problem of *multinominal classification* or *multi-class classification*, that is classifying samples into more than two classes (*binary classification*). The core of the problem, is that several algorithms for machine learning was designed to handle two-class problems, and thus initially unsuitable where we are facing problems with more than two classes [13]. There are, however, methods for adapting algorithms that were natively designed for two-class problems to be suitable for application on multi-class problems.

Originally designed for	
Two-class	*Multi-class*
Decision trees	Naive Bayes
Neural Networks	Random Forests
Support Vector Machines	Nearest Neighbors

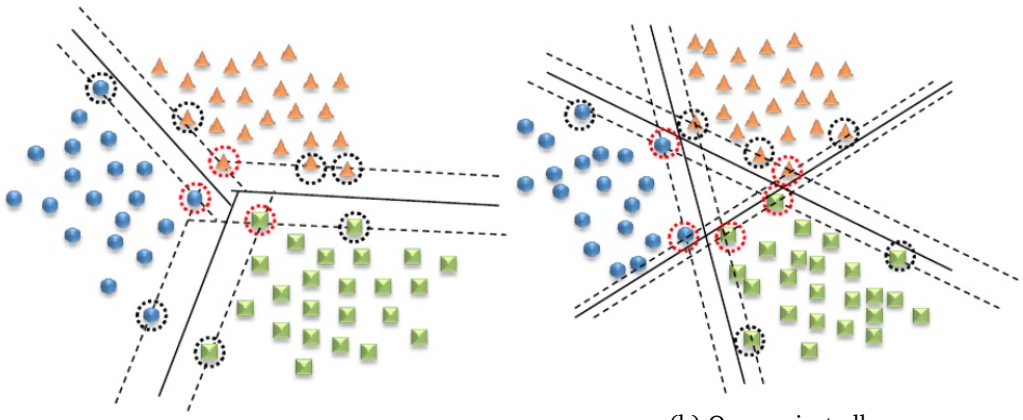
Table 3: Examples of two-class and multi-class algorithms [14, 15]

One-against-one (OAA)

One-against-all refers to training a classifier K times, where K is the number of classes. For each training, one class is considered positive and all the remaining classes as negative [43]. The same approach used in Mohaisen & Alravi [17] to detect Zeus-malware.

One-against-one (OAO)

Contrary to OAA, we have *one-against-all*, which trains a classifier for each *pair* of classes [44]. To achieve this, $K(K-1)/2$ classifiers must be trained, where K is the number of classes [43].



(a) One-against-one

(b) One-against-all

Figure 12: Modes for multi-class classification [10]

3.4 Challenges

There exist several challenges related to the general machine learning process that the reader should be aware of. This section provides an overview of the most general challenges to machine learning.

3.4.1 "No free lunch"

The *No Free Lunch* theorem states that there are, in general, simply no way to expect that a certain classifier performs better than another [45].

> *The apparent superiority of one algorithm or set of algorithms is due to the nature of the problems investigated and the distribution of data [45].*

From this, it is clear that we cannot draw conclusions from the performance of a single classifier. Instead, the data should be evaluated with multiple classifiers.

The *No Free Lunch* theorem can also be applied to feature selection; if no prior assumptions are made, we cannot know which features, or which number of features that will be the optimal feature subset for a classification task.

3.4.2 "Ugly Duckling"

The *Ugly Duckling* theorem applies to feature selection in general. As stated in 3.2, the different features may, and will in most cases, contribute to classification of a different degree. The main takeaway from the *Ugly Duckling* theorem is that to achieve reliable classification performance, we need to select the features that contribute most to classification.

> *Given that we use a infinite set of predicates that enables us to distinguish any two patterns under consideration, the number of predicates shared by any two such patterns is constant and independent of the choice of those patterns. Furthermore, if pattern similarity is based on the total number of predicates shared by two patterns, then any two patterns are equally similar [45].*

Based on this, we will in our work use features selected by different feature selection methods.

3.4.3 Overfitting and underfitting

Overfitting occurs when the model fits the training data "too" good, but fail to generalize new, unseen samples [13]. Opposite, we have *underfitting*, which occurs when the trained

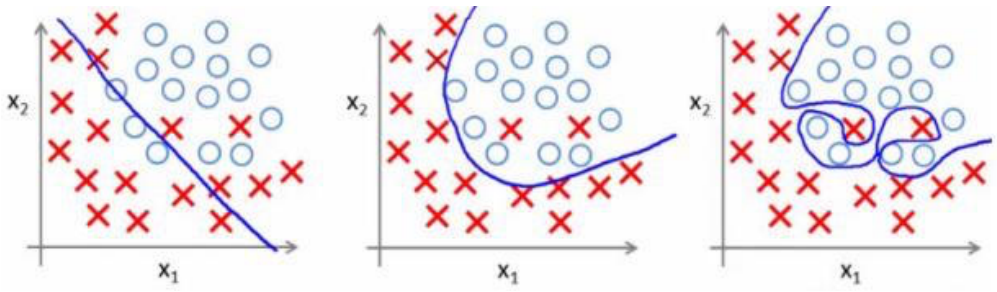


Figure 13: Overfitting and underfitting [11]

model do not fit the training set good enough, also resulting in bad classification accuracy. Figure 13 illustrates the problem of over- and underfitting. The data is, as in many other cases, not linearly separable. We can see that the polynomial in the middle figure will

probably be the one that discriminates the new samples best. The example in the left do not fit the training data accurately, whilst the figure on the right hand side fits the training data perfectly, and yet we can assume that this model would not make good predictions about new unseen data points. The figure is very simplistic as opposed to real world problems where we will almost exclusively have more features, and thus dimensions, making visualization harder. This again results in that overfitting and underfitting will not be visible to us through visualization. Therefore it is of great importance to be aware this problem, and a few methods to reduce it.

Splitting the data

Splitting the data set into mutually exclusive subsets, one for training and one for testing. The model is trained on the first subset of data, and then evaluated with the second subset.

Leave-one-out (LOO)

When the number of samples is low, it might be unreasonable to split the data, as it may deprive the learning algorithm of samples to make the model representable [13]. Another approach is the leave-one-out method, which refers to running the learning algorithm k times, where k is the number of samples in the data set. Each time, leave *one* of the samples out of the training and evaluate the classification on that sample. Repeat until every sample has been excluded from the learning algorithm one time. The classification accuracy is then computed as the average of each classification score. This method is computationally expensive on large datasets, as the learning algorithm must be run k times [13].

k*-fold cross validation*

This refers to a compromise between the two previously mentioned methods, keeping the same principle as the LOO method while decreasing the computational cost. By splitting the data into k folds (subsets) and then train the model on $k - 1$ folds and evaluate the model on the last fold. This is repeated until the every fold has been left out of the learning algorithm exactly once. The performance is computed as the average of each of the k runs [13].

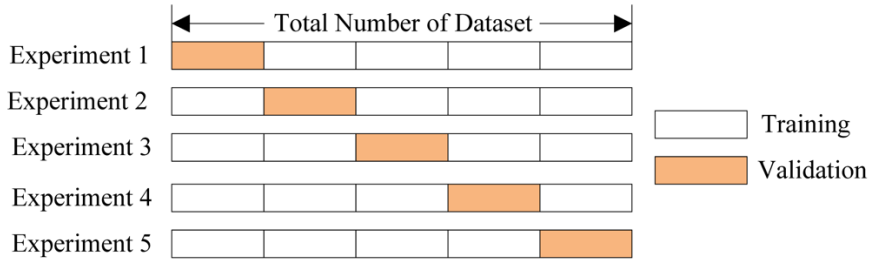


Figure 14: 5-fold cross validation [12]

3.4.4 Validation of results

A pitfall to avoid when evaluating performance is that authors may fine-tune parameters in their own research to achieve the best performance possible, and when comparing against other algorithms, use the pre-set parameters. To avoid this, a data set should be split into a learning set and a validation set. The idea of this is that the parameters can

be tuned to perform as good as possible on the learning set, and the settings must then be kept when running on the validation set [13].

4 Related work

Today there exists, to the authors knowledge, no research which focuses on classification of malware into types and families based on features derived from static analysis. The previous research rather focuses on labeling a file as either benign or malicious. In this binary classification, there exist some work which still are partly relevant to the thesis. We will in this chapter use the term *malware detection* for binary classification of malware, whilst *malware classification* will be used for labeling malware in the terms of family and/or type.

4.1 Binary classification

Kolter et al. in 2007 did a study in where they used the hexdump sequences from files as data, where they also analyzed different sequence length, n-grams of this as features. This yielded very high accuracy, noted as Area Under (ROC) Curve(AUC). table [16] shows the achieved results with different classifiers and $n = 500$ [16]. The results suggests that the results will remain high when the methods are applied to larger datasets.

Method	**AUC**
Boosted J48	0.9836
Boosted SVM	0.9744
IBk,k=5	0.9695
SVM	0.9671
Boosted Naive Bayes	0.9461
J48	0.9235
Naive Bayes	0.8850

Table 4: Results from Kolter et. al [16]

Bragen in 2015 applied machine learning on opcode sequences to achieve ~95% accuracy with RandomForest method [39]. A limitation in the mentioned research are the size of the samples; Kolter et al. [16] used 1971 malicious files and 1651 benign, while Bragen [39] used 992 malicious and 771 benign.

Cohen [46] implied in 1987 that there are no algorithm that will able to detect all computer viruses. This assertion that was strenghtened by Chess and White [47]. From this, we can assume that our work will not result in perfect classification accuracy among the large dataset we are using. An important question can however be derived from these claims, that is, even if we cannot implement an algorithm to classify *all* malware, how good predictions *can* we achieve?

Leder et al. [48] achieved in 2009, a 100% accuracy with zero false positives with an approach based on similarity scoring of defined *points of interest* from static analysis through code disassembly. This work differs from our approach as the approach does not implement machine learning, but also in that the accuracy is calculated from determining if a certain malware specimen can be labeled as a member of a certain malware family or not. In total, there was used seven different malware families for this research. Mohaisen

and Alrawi [17] was able to achieve close to 95% accuracy in classificating malware as Zeus or non-Zeus malware.

Class	Features
File System	Created, modified, deleted, size (quartiles), unique extensions, count of files under common paths
Registry	created keys, modified keys, deleted keys, count of keys with certain type
Network	see below for each sub-class
Ip and port	unique dest IP, certain ports (18 ports)
Connections	TCP, UDP, RAW
Request type	POST, GET, HEAD
Response type	response codes (200s through 500s)
Size	request (quartiles), reply (quartiles)
DNS	MX, NS, A records, PTR, SOA, CNAME

Table 5: Features used in "Unveiling Zeus" [17]

While not sharing the same featues as in our project, this, and Bragen [39] yields an interesting observation. In both research *Support Vector Machine* did achieve different rates of classification accuracy, while in Mohaisen [17], SVMs proved to achieve accuracy for about 94%, while Bragen [39] reached 70%-80% with this type of classifier. In the former, there was used malware exclusively in the work, while in the former there was both malicious and beningn samples. The interesting part about these two approached compared to ours is that SVMs were designed to be applied on binary classification problems [49, 50], and usually performs good on two-class problems as well.

4.2 Multi-class Classification

In our work, we will be exploring *multi-class* classification, that is, more than two classes. The "*no-free-lunch*", discussed in Section 3.4.1 states that we cannot assume that one algorithm performs better than another, but from the discussion presented in this section, we will assume that *Support Vector Machines* decrease in classification accuracy in our data set, as we have a higher number of classes than previous reseach. Another observation made, is than decision trees and/or forests often account for the higher accuracy in classification of malware. Althoug we have a multi-class problem, we cannot directly assume that these methods will provide the highest scores. On the contrary, decision trees and forest was designed to handle multi-class problems, which means that these methods potentially can be better suited for classiciation in our dataset. Rieck et al. performed in

1: Backdoor.VanBot (91)	8: Worm.Korgo (244)
2: Trojan.Bancos (279)	9: Worm.Parite (1,215)
3: Trojan.Banker (834)	10: Worm.PoeBot (140)
4: Worm.Allaple (1,500)	11: Worm.RBot (1,399)
5: Worm.Doomber (426)	12: Worm.Sality (661)
6: Worm.Gobot (777)	13: Worm.SdBot (777)
7: Worm.IRCBot (229)	14: Worm.Virut (1,500)

Table 6: Description of data set used in Rieck et al. [18]

2008 work on malware classification with features derived from dynamic analysis. The

focus in this study did not include labeling of malware family. More than 10,000 samples was used in this research, which yielded a result of 88% accuracy on family classification in average. A limitation however, is that Nephentes honeypot[1] was used for data collection, which means that worms will be overrepresented in the data set, which can be shown in Table 6 [18].

Liao 2012 presented a paper on malware detection based on information from PE-header [51]. While this, as well as the majority of other work, is based on binary classification, some of the features included in this research may yet be important in our work as well. Based on the five most important features in this research, only one feature is included in our work, that is the size of initialized data. From this we can assume that this feature will be of importance in distinguishing between family and/or type as well.

Authour(s)	**Features**	**Classification**	**Number of samples**
Rieck et al. [18]	Dynamic	Family Classification	10,072(12 classes)
Bragen [39]	Static *(opcode sequence n-grams)*	Binary Classification	1,763
Kolter et al. [16]	Static *(hexdump n-grams)*	Binary Classification	3,622
Zhang et al. [52]	Static *(Byte sequence n-grams)*	Type Classification	873(3 classes)
Moskovitch et al. [53]	Static *(Opcode sequence n-grams)*	Binary Classification	26,093
Mohaisen et al. [17]	Static *(File system, registry and network)*	Binary Classifcation	2,001

Table 7: Description of previous work

[1] http://sourceforge.net/projects/nepenthes/

5 Large-scale Malware Analysis

This chapter discusses the methodology to be used in the experimental phase of our work. Initially we will discuss the choice of methods for our research, the phases of acquiring data and how the features was *extracted* from the raw characteristics. Further, an overview of the the processed data set is given, as well as a description of the data subset generation. Lastly, we present an overview the different machine learning methods used in our research. In contrast to the majority of other research, we use data that are less time consuming to extract, to reduce computation due to the large number of malware samples we are considering, but also to explore if these data can be reliable in malware classification. The number of samples in our work is also considerably larger than in other research.

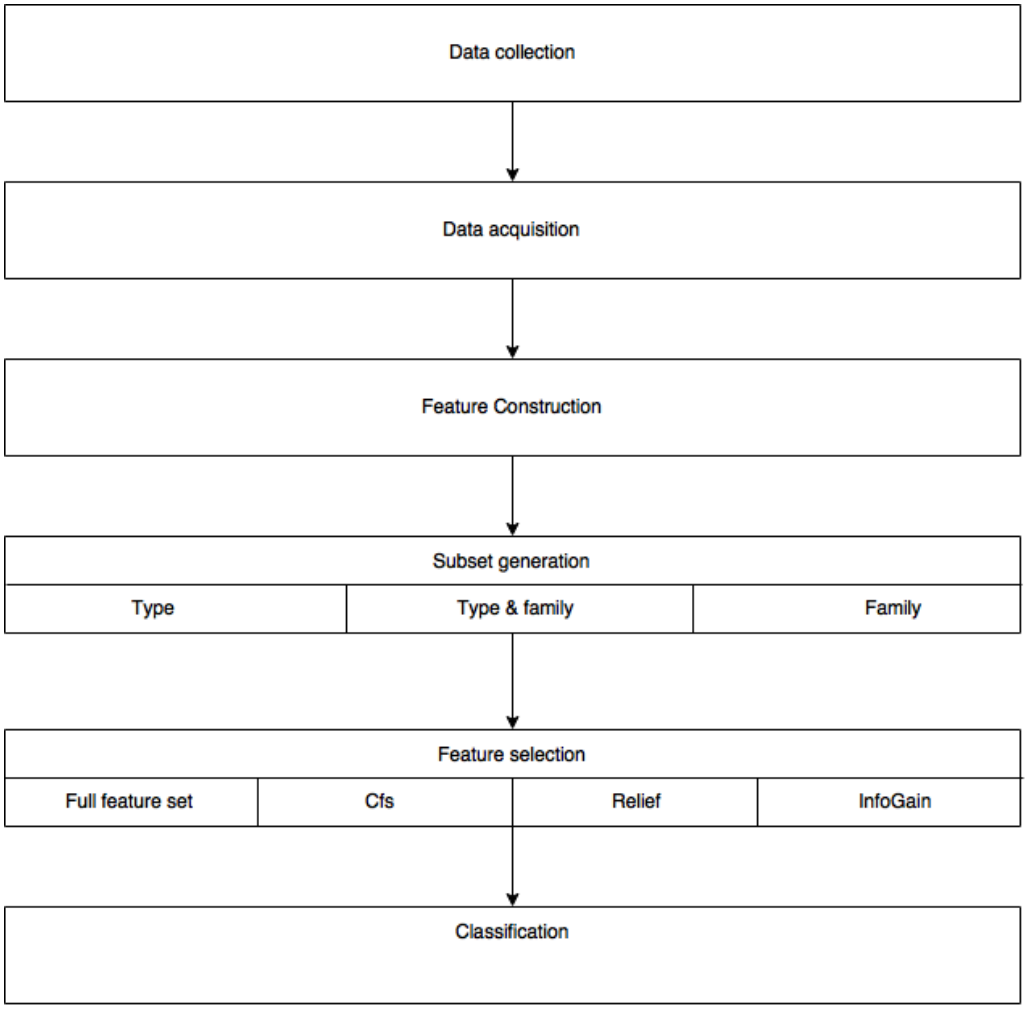


Figure 15: Methodology for large-scale static malware analysis and classification

5.1 Choice of methods

This section discusses the methods that we will utilize to try and answer the different research questions. Quantitative research methods relies around the *quantification* of observations, utilizing numerical and statistical techniques to provide an objective reasoning for the results [54]. Qualitative research aims to gain a deeper understanding of underlying reasons for a problem without quantification of data, and to generate new ideas for further research problems and/or hypotheses [54]. To answer our research questions, we will utilize *quantitative* research methods.

- What features that can extracted from the static analysis tools *PEframe* and *VirusTotal*, will be most relevant for distinguishing malware into type and family?
- What accuracy can be achieved with features derived from the static analysis tools *PEframe* and *VirusTotal*?
- Which methods for feature selection and classification performs the best on the features constructed?

To investigate these research questions, we will use several algorithms for machine learning, both for feature selection and classification. Due to the important theorems of "*no free lunch*" and "*ugly duckling*", different methods are necessary for us to be able to draw conclusion of the results, and only through performing a variety of experiments, we can argue about the validity of the results. The same approach will be be highlighted from three different points of view to answer all three questions. This is due to the fact that it is impossible to draw any conclusions about

- the quality of the features,
- what accuracy we can achieve or,
- which methods for feature selection or classification that performs best

without the results from classification.

5.2 Data acquisition

The motivation was to perform a study of different malware categories and families that are available for Windows OS. If we look on previous studies that involve PE32 file formats for MS Windows, we can see that majority only target differentiation between "malicious" and "benign" samples. However, there are quite many malware categories that have different characteristics and functionality. Therefore, our idea is to study how static analysis of PE32 files with help of Machine Learning can facilitate large-scale malware detection into families and categories.

Malware samples acquisition. There was a malware collection initiative that took place within the Testimon Research group[1] at HiG during April - June 2015. It resulted in a number of samples from students, 10 first archives of *VirusTotal* and files from VxHeaven were collected in addition. After thorough analysis and filtering, we derived all possible PE32 files and removed other types of files. In overall we ended up with 407,741 malware samples, yet some of them will be eliminated. The total size of all the executables is 136GB.

[1] https://testimon.ccis.no/

Characteristics acquisition

Since we targeted a static analysis due to large number of samples, it was decided to extract as much characteristic raw data as possible. Two main sources that we used were *PEframe* and *VirusTotal*. *PEframe* presents comprehensive set of attributes that can be found in the PE header. There were some works before showing that it can be possible to identify malware using such headers information. *VirusTotal* presents scan results from over 50 antivirus databases, information about possible packers and compressors in addition to basis PE headers data. Moreover, we used standard linux tools to get more file characteristics, e.g. size of different sections, strings and also entropy. Finally, we created a MySQL database that contains raw characteristics of the PE32 windows executables (all malware) filtered out of the mess of different malwares that were present in gathered earlier sets. Fields in the raw data SQl dataset are following:

1. *md5* - md5 of the file.
2. *virustotal_file_report* - retrieved Virustotal report using private API, serialized JSON-formatted[2].
3. *virustotal_file_behaviour* - retrieved Virustotal report using private API, serialized JSON-formatted[3].
4. *virustotal_file_network_traffic* - retrieved Virustotal report using private API, serialized JSON-formatted [4].
5. *peframe* - Report from executing the peframe script agains the malware, JSON-formatted[5].
6. *file* - output of the Linux command 'file', particularly interesting different architectures and so on.
7. *strings* - output of the Linux command 'strings', we can think of counting the number of strings, etc.
8. *size* - output of the second string of the Linux command 'size', contains information about size of different sections of the binary file.

   ```
   text data bss dec hex filename
   105182 2044 3424 110650 1b03a /bin/ls
   ```

9. *file_entropy* - entropy of the file, may indicate strong encryption in case if close to 8.0.
10. *size_of_file* - size of the file in bytes.
11. *do_not_process* - just a binary flag, used to indicate which malwares could not be processed by peframe script due to time-out and have to be excluded from further experiments.

The target of the project is to process raw characteristics into numerical features and extract more or less consistent/conventional names of malware families and malware categories. Both can be used later in classification tasks. Malware families defined in work by Microsoft[6] [55]

[2]https://www.virustotal.com/en/documentation/private-api/#get-report
[3]https://www.virustotal.com/en/documentation/private-api/#get-behaviour
[4]https://www.virustotal.com/en/documentation/private-api/#get-network-traffic
[5]https://github.com/guelfoweb/peframe
[6]http://www.microsoft.com/security/pc-security/malware-families.aspx

5.3 Feature construction

Before meaningful analysis can be performed, we must extract numerical features from the acquired data. While we, in a strictly technical manner, are acquiring raw static characteristics, 'feature extraction' is an important part of the machine learning process, and we therefore choose to use the term 'feature construction' to describe the process of how to get numerical data from the raw characteristics set.

To accomplish this, we had to analyze the raw characteristics, and in particular the content from *VirusTotal* and *PEframe*. All content from both parties was in the format of a string formatted as a serialized JSON object[7]. An example of such a string is given in the Appendix B.2. Appendix B.2 shows the same string unserialized and printed with the "human readable print" (`print_r`) in PHP[8]. In this step, we focused on extracting as many numerical features as possible. In this step, we did not emphasize what features we anticipated to be most important, as we will perform feature selection before the analysis in the experiments execution to determine which features that contribute most to classification. After deciding on what data we wanted to gather, a python script was written to first select the data, and then put it into a new database table. The size of the 'rawData' table was 19.5GB and the processed table named 'data' are of the size 98.7MiB.

Features

This section provides an overview of the names of the features constructed, as well as from which tool they was extracted in the preprocessing step in 'data' SQL table. Example outputs from *PEframe* and *Virustotal* can be found in Appendix B.1 and B.2 respectively.

1. *md5* - md5sum of the sample.

*From **VirusTotal***

1. *vt_codesize* - The size of the code in the file, retrieved from virustotal.
2. *vt_res_langs* - The number of resource languages detected in the file. Retrieved from virustotal.
3. *vt_res_types* - The number of PE resource types detected in the file. Retrieved from virustotal.
4. *vt_sections* - The number of PE sections in the file. Retrieved from virustotal.
5. *vt_entry_point* - Decimal value of entry point, i.e. the location in code where control is transfered from the host OS to the file. Retrieved from virustotal.
6. *vt_initDataSize* - The size of the initialized data. Retriced from virustotal
7. *vt_productName* - The length of the field ProductName in the response from virustotal. Can e.g. be "Microsoft(R) Windows(R) Operating System".
8. *vt_originalFileName* - The length of the original file name, Retrieved from virustotal.
9. *vt_unitializedDataSize* - The size of the part of the file that contain all global, uninitialized variables or variables initialized to zero. Retrieved from virustotal.
10. *vt_legalCopyright* - The length of the field LegalCopyRight in the response from virustotal .E.g."(C) Microsoft Corporation. All rights resad."

[7] http://www.json.org
[8] http://www.php.net

*From **PEframe***

1. *pe_api* - The number of suspicious API class, from PEframe.
2. *pe_debug* - The number of opcodes recognized as common anti debug techniques, discovered by PEframe.
3. *pe_packer* - The number of packers discovered by PEframe.
4. *pe_library* - The number of dll calls in the file. Retrieved from PEframe.
5. *pe_autogen* - The number of autogens discoveren by PEframe.
6. *pe_object* - The number of object calls discovered by PEframe.
7. *pe_executable* - The number of calls executable .exe files within the file.
8. *pe_text* - The length of the 'text'-field in the PEframe response.
9. *pe_binary* - The number of binary(.bin) files called by the file.
10. *pe_temporary* - The number of .tmp files accessed by the file.
11. *pe_database* - The number of .db files accessed by the file.
12. *pe_log* - The number of log entries accessed by the file.
13. *pe_webpage* - The number of web pages access by the file.
14. *pe_backup* - The number of backups the file performs or accesses.
15. *pe_cabinet* - The number of references to .cab[9] files.
16. *pe_data* - The number of .dat files accessed by file.
17. *pe_registry* - The number of registry keys accessed or modified by the file.
18. *pe_directories* - The number of directories accessed by the file.
19. *pe_dll* - The number of dll's accessed by the file.
20. *pe_detected* - The number of suspicious sections in the file.

Other

1. *size_TEXT* - The first output from the file-command on unix systems. This is the size of the instructions.
2. *size_DATA* - The second output from the file-command on unix systems. This is the size of all declared/initialized variables, i.e the 'data' field of the file.
3. *size_OBJ* - The third output from the file-command on unix systems. This is the size of all unitialized data, i.e the BSS-field of the file.
4. *size_TOT* - The fourth output from the file-command, i.e the sum of the text, data and bss fields of the file.
5. *filesize* - The total size of the file, including metadata.

5.4 Subset generation

To be able to perform feature selection and/or classification on the data, we must create subsets of the full data set as Weka are not able to perform these tasks on the high

[9] https://msdn.microsoft.com/en-us/library/ms974336.aspx

number of classes we achieve for families (10,362). The number of classes for malware types in our data set is 35. We therefore created five subsets in total, three excluding malware type, including the 10, 100 and 500 most frequent malware families, for family classification. For type classification we created two subsets, both excluding malware family, including the 10 most frequent types, and one with all malware types. This is important because otherwise the family distribution would be highly imbalanced, as the 500 most frequent families makes up 292,596 samples, meaning that the other 9,862 malware families are distributed over the remaining 35,747 samples, which are relatively rare.

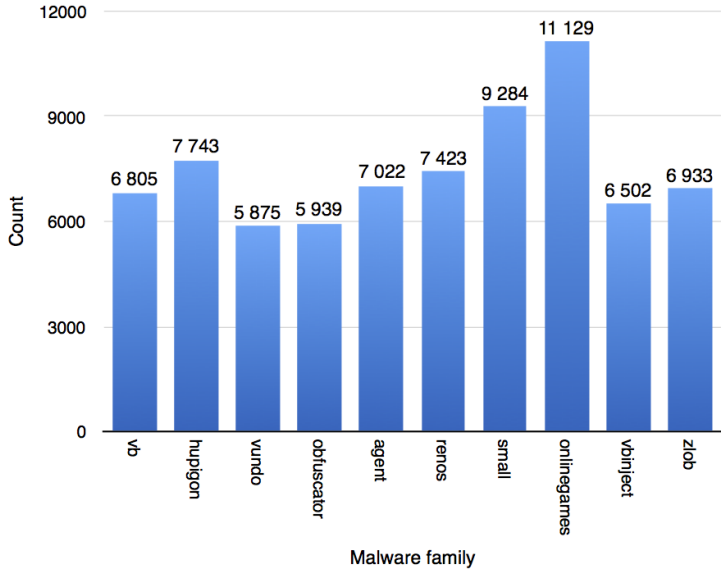


Figure 16: 10 most frequent families

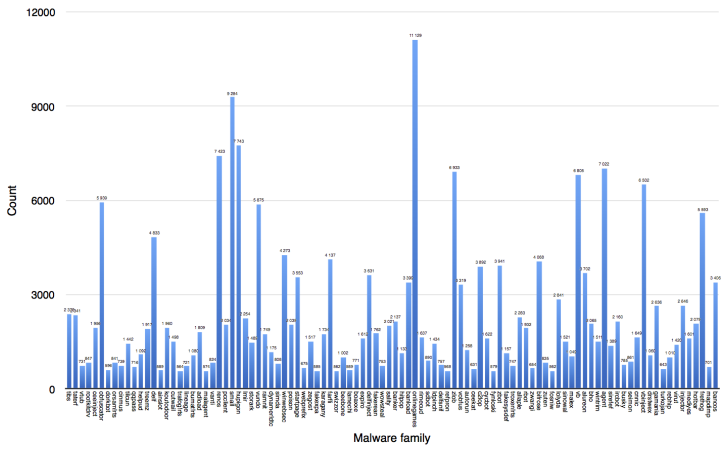


Figure 17: 100 most frequent families

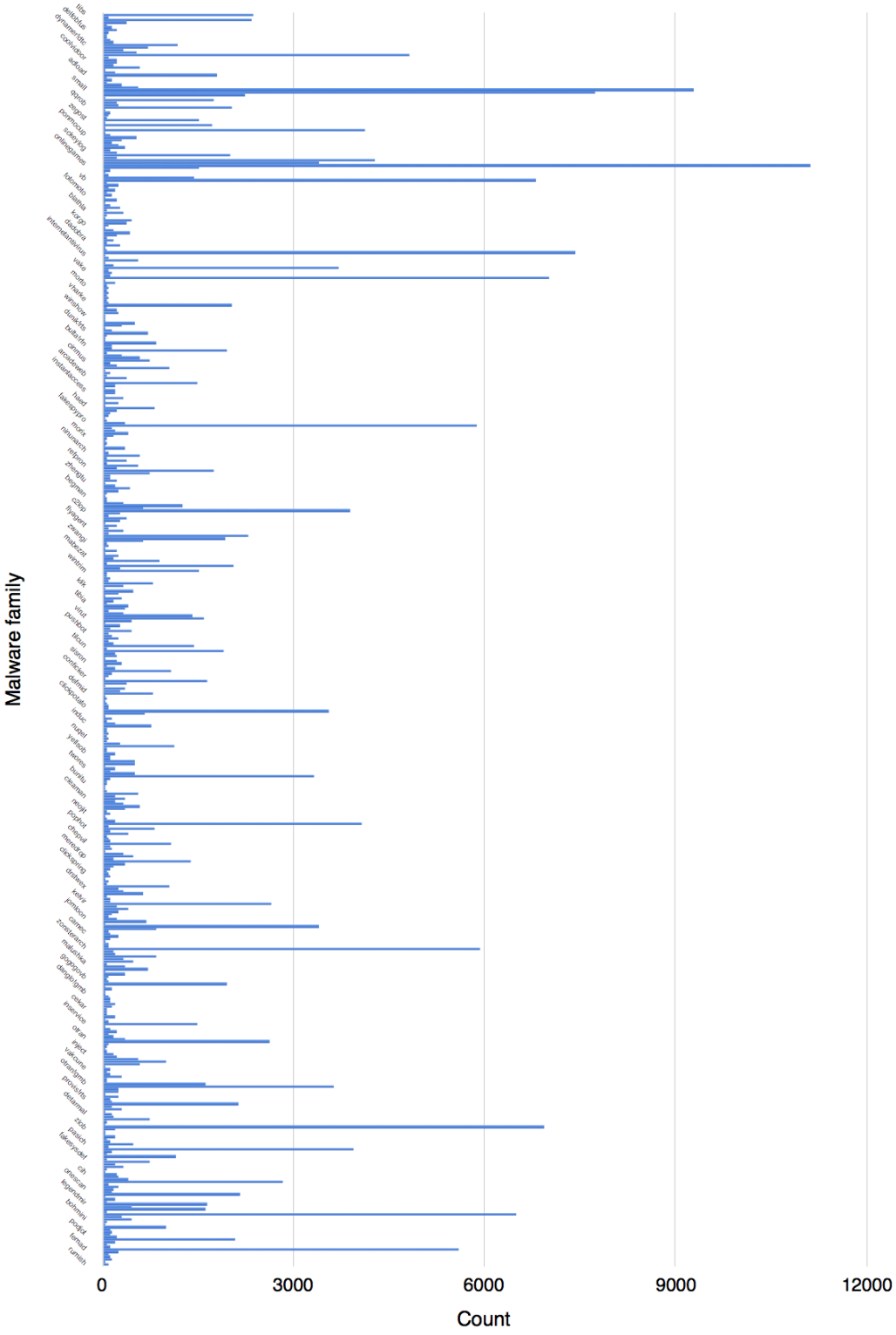


Figure 18: 500 most frequent families

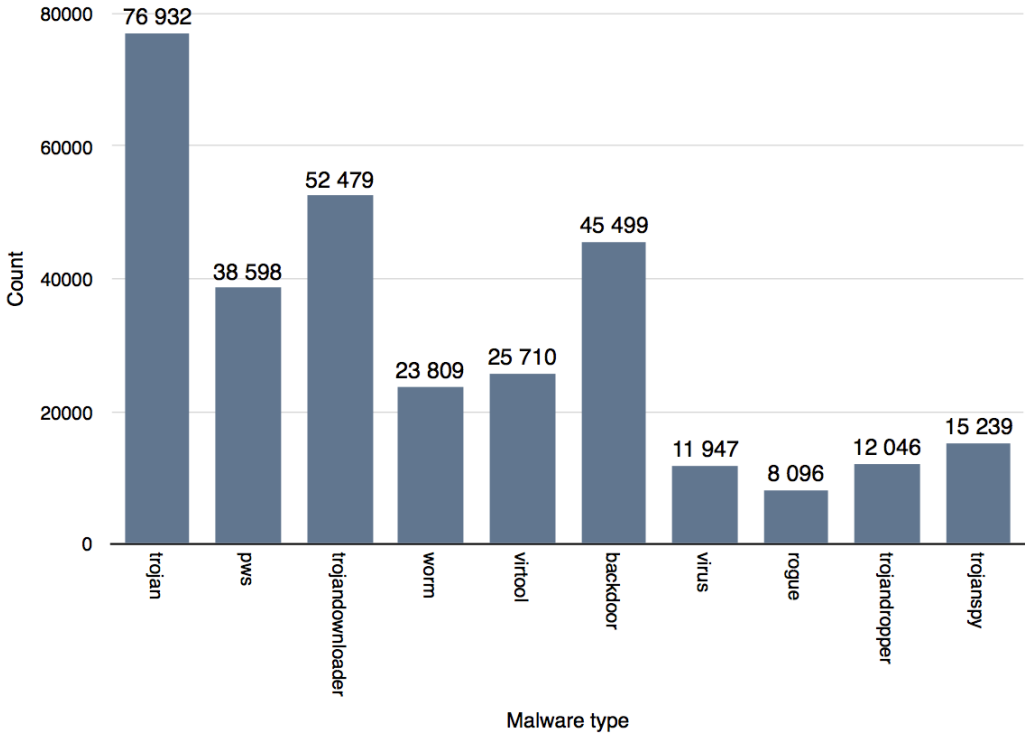


Figure 19: 10 most frequent types

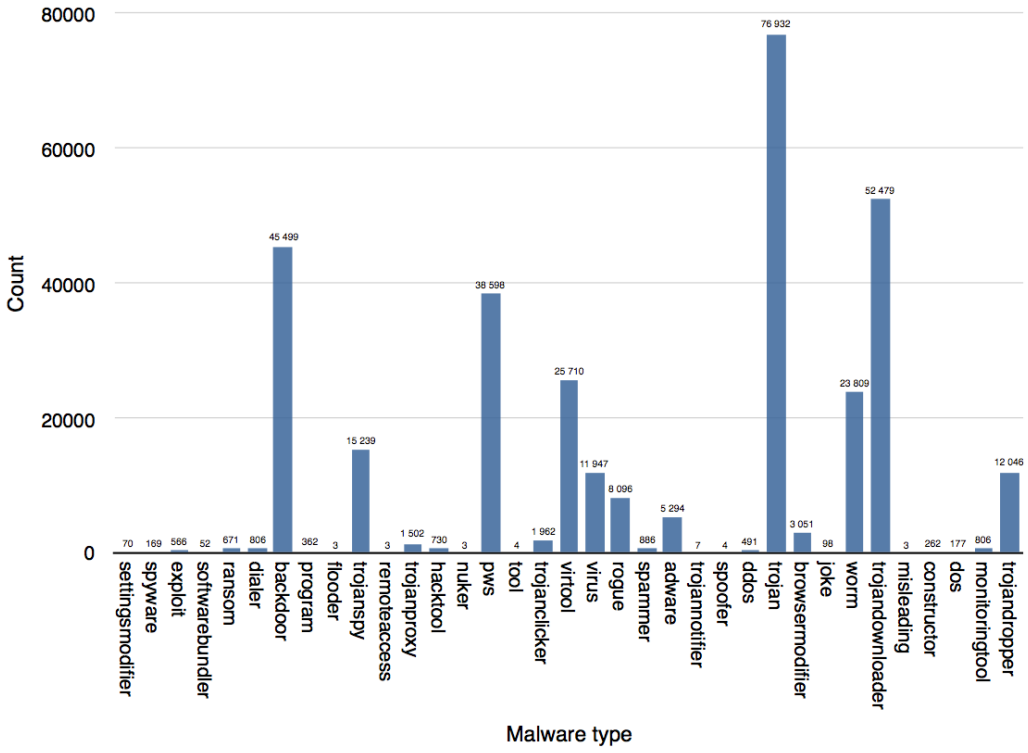


Figure 20: All malware types

5.5 Machine Learning Methods Used

In this section we present the different methods for feature selection and classification used in the experiments. For feature selection and classification, the Weka Data Mining software, and the names used for the different methods in this sections corresponds to the name of the methods in Weka [56]. There exist other software that implements machine learning methods, such as RapidMiner[10] and Orange Data Mining[11], but he have chosen to focus exclusively on Weka since it is most commonly used and implements a broad range of algorithms. Own implementations of algorithms were considered, but due to the limitation of time, we used Weka for all experiments, which also enabled us to have time to utilize multiple algorithms for feature selection and classification.

5.5.1 Feature selection

- *CfsSubsetEval* - The Cfs feature selection methods selects features with high correlation to classes and disregards features with high correlation to each other. Experiments by Hall shows that CFS "*quickly identifies and screens irrelevant, redundant, and noisy features, and identifies relevant features as long as their relevance does not strongly depend on other features.*" [57].
- *InfoGainAttributeEval* - The InformationGain feature selector ranks features by entropy. "*Concretely, it measures the expected reduction in entropy*" [58].
- *ReliefF* - Attributes are ranked from how well features separates classes on values that are close to each other [13]. This is an extension from the *RELIEF* algorithm, originally designed for two-class problems and prone to noise. The *ReliefF* algorithm is extended to better suit multi-class problems and to robust on data with missing values [13].

5.5.2 Classification

- *RandomTree* - The randomtree implementation creates an unpruned decision tree using K number of random features for each node [59].
- *J48(C4.5)* - A variant of decision tree which utilizes information gain in order to prune the tree to avoid overfitting [60, 13].
- *RandomForest* - This method combines several decision trees, usually 100 [13]. Each tree is generated from the whole learning set. In the test phase, a vote is collected from each tree to label the samples [13].
- *IBk* - A variant of K-Nearest Neighbor classification algorithm that are able to perform cross validation to select the optimal number of K. In our experiments, we did use this option. *In K-Nearest Neighbor*, K, is set initially to an odd number. Distance is calculated from the new sample to all other samples. The new sample is then labeled as an instance of the class that is highest represented by the K number of nearest neighbors [61].
- *NaiveBayes* - The Naive Bayes classifier assumes "that features are independent given class" [62]. This assumption is usually bad (thus 'naive'), the Naive Bayesian classifier often performs good compared to other classifiers [62].

[10] https://rapidminer.com
[11] http://orange.biolab.si

- *BayesNet* - A *Bayesian Network* consists of joint probabilities of variables in a directed acyclic graph [63]. The probabilities for the observable variables are computed by the rule for posterior probability. An example a *BayesNet* is depicted in Figure 21.
- *MultiLayerPerceptron* - MLP is a type of Artificial Neural Network that simulates how the human brain neurons learn from observations [13]. This is achieved by that a set of weights and corresponding input nodes is set to the same number as the size of the feature vector. Each weights are set by random at initialization. For each observed value, the weight is adjusted to some degree, determined by a learning step, and a backpropagation function, usually the delta learning rule [13].
- *Support Vector Machine* - Learns by determining the optimal *hyperplane* to best separate the data. This is achieved by that SVM can project the data to a higher dimension feature space to create such a hyperplane. With a *kernel function*, this hyperplane is processed to fit the original feature space [64].

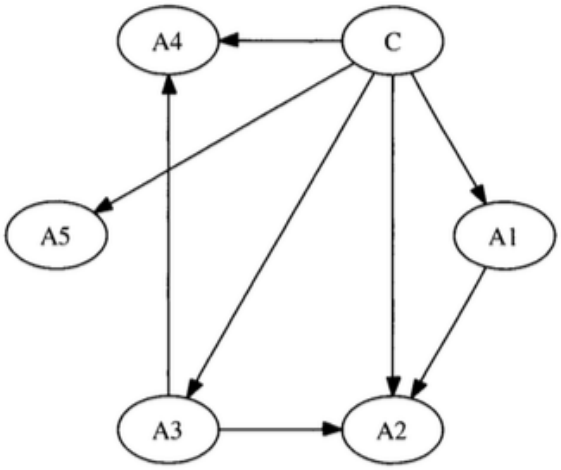


Figure 21: A simple example of a Bayesian Network [13]

6 Experiments, results and discussion

In this chapter we provide an overview of the experiments performed, information gained from preprocessing, results and discussion of results of both feature selection and classification.

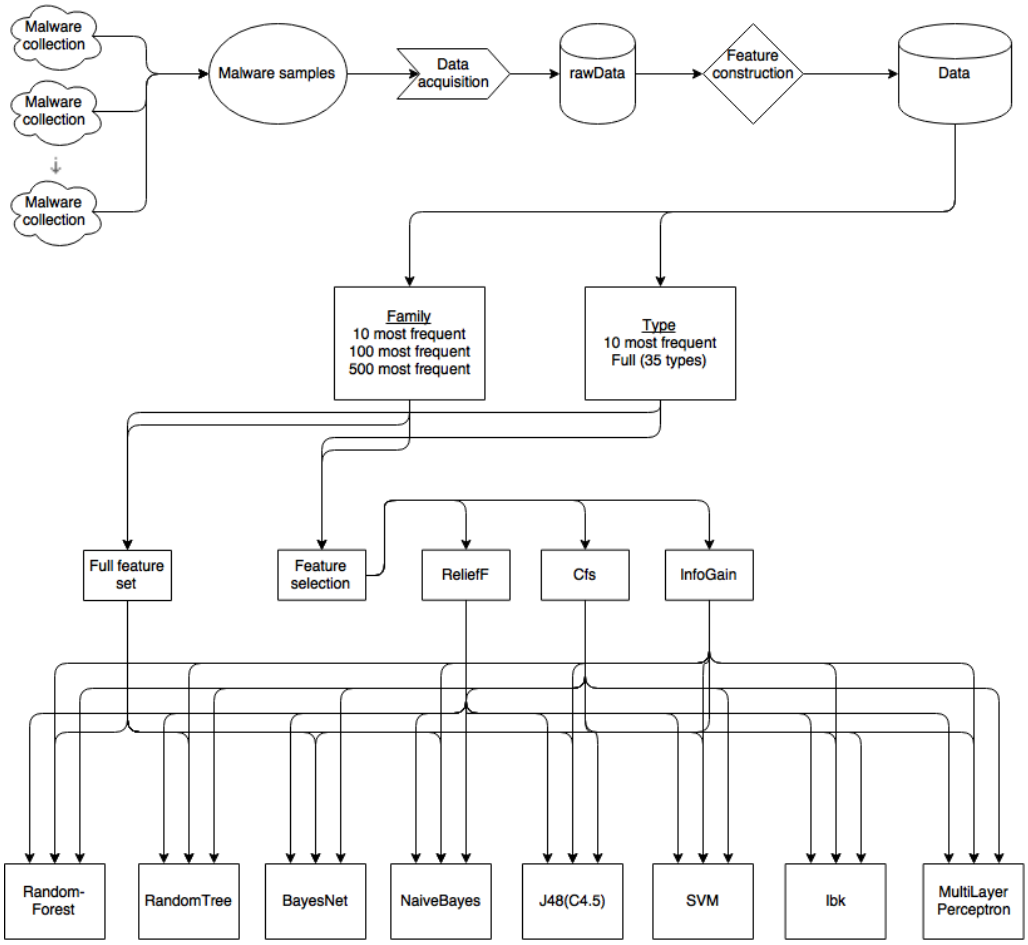


Figure 22: Workflow for large-scale analysis

6.1 Experimental Environments

As previously discussed, the collection of malware samples and raw data extraction was performed before the work performed in this thesis. Due to this, and the fact that the size of the processed dataset required extensive computational resources, experiments have been performed in different environments.

Software

- Python 2.7 [65]
- Pip 7.1.2 [66] (used to download PyMySQL)
- PyMySQL 0.6.7 [67]
- MySQL 5.5.46 [68]
- PHP 5.5.9 [69]
- BigDump 0.36b [70]
- Weka 3.7.3 [56]
- phpMyAdmin 4.5.1 [71]

Different environments used in our work

- Name: Apple MacBook Pro
- Processor: 2.5GHz Intel Core i5
- Memory: 8GB DDR3
- Hard Drive: 500GB HDD
- Operating System: OS X 10.10.5

- Name: iMac
- Processor: 3.4GHz Intel Core i7
- Memory: 16GB DDR3
- Hard Drive: 1TB HDD
- Operating System: OS X 10.11

- Name: Tigger Server
- Processor: 3.6GHz Intel Core i7
- Memory: 8GB DDR3
- Hard Drive: SSD RAID
- Operating System: Ubuntu 14.04

In the process of feature construction, MacBook was used, as we estimated that there were no need for more computational power in that phase. There were, however some problems arising due to the size of the rawData SQL dump of approximately 20GB. Initially we had some difficulties with the way MySQL[1] handles importing, requiring available RAM space equal to the size of MySQL table to be imported. For this purpose, we utilized BigDump [70] PHP software, designed to split the SQL dump into smaller chunks before importing one by one. In the process of feature selection and classification, all

[1] https://www.mysql.com

three described environments were utilized as a result of computational time, especially in the two largest subsets of data.

6.2 Data acquisition

The original data used in this experiments, consists of 407,730 different samples downloaded from VxHeaven. With the large number of samples we can assume that samples are distributed over a large number of both types and families of malware. Without a certain distribution over multiple types and families, we would not have been able to argue for the validity of the results.

6.3 Feature Construction

This section describes the process of feature construction and data subset generation.

```
connect to database;
save first md5 in md5_temp;
for n in range (0, 40773) where 'md5' > md5_temp LIMIT 10 do
    for each row do
        if peframe contains a valid result then
            if sample detected by Microsoft from virustotal then
                Extract data and store in new database table;
            end
        end
        set md5_temp equal to md5 in row;
    end
end
```

Algorithm 1: Feature construction algorithm

The python script that implements the algorithm is included in Appendix C.1. Note that we do not select the entire database. This is because the size of the table. Our practical implementation for this was subsequently to use the md5 as the variable to perform the queries with offsets. MySQL do also have a built-in query option called 'limit'. The problem with this option is that MySQL iterates through every value in every row to reach the offset, making the queries slower when the offset reaches the point where MySQL requiring more memory than available to select the desired rows. Our rationale for selecting *md5* to achieve queries with offset is that the raw characteristics database table is sorted on the *md5* values ascending. With this approach, we were able to decrease the query time of a query with offset $= 200,000$ returning exactly 1 row, from 140 seconds to 0.7 seconds.

Since we focus on determining the families of the malware, we decided to exclude all results that was not recognized by Microsoft (according to *VirusTotal*), to be as consistent as possible. The reason for this, is that Microsoft was the only provider we found that implemented the CARO naming scheme[41] systematically.

```
if 'Microsoft' in vt.scans._asdict() and vt.scans.Microsoft.result is not None:
  tekst = vt.scans.Microsoft.result.split(':')
  vt_type = tekst [0]
  vt_family = tekst[1].split('/')[1].split('.')[0]
```

Listing 6.1: Code snippet for family and type name extraction

Listing 6.1 shows the code in the algorithm that extracts the type and family names from each sample. These two names, and all features provided in Section 5.3 is stored in a new

database table. The processed dataset after performing feature construction algorithm contain 328,337 samples distributed over 35 distinct malware types and 10,362 malware families.

Note that for Weka to be able to read data, it must be stored in **Attribute-Relation File Format** .arff[2] [72]. The code for creating the .arff is included in Appendix D.1.

Subset Generation

Due to size of the dataset and the amount of malware families (10,362), computation was infeasible. To cope with this we split the data in five different subsets, three containing the features describes in Section 5.3 and only the malware families as class. The remaining two contains the same features, and includes the malware type as class. By infeasible, we mean that computation either failed, or took too long (more than 72 hours) to complete. Algorithm 2 describes how the subsets was generated. In the process of generating the subsets for malware type classification, we used $t = 10$ and $t = 0$. The same algorithm was used for generating the subsets for family classification. The only difference from Algorithm 2, is that instead of the variable *type* we use family. For the family subset generation we used $t = 10$, $t = 100$ and $t = 500$.

```
connect to database containing processed data;
for each row do
    count all occurrences of each malware type and store in list;
end
sort list on malware type count, descending;
set threshold, t;
remove all malware types from list with frequency < t;
for each row do
    if Malware type is in the list then
        store the row in a new database table;
    end
end
```

Algorithm 2: Algorithm for generating data subsets

Naming of data subsets

The three first letters indicates whether the classes in the data file are instances of *types* or *families*, while *t_xxx* denotes the minimum threshold in regards to frequency in our full data set.

1. **fam_t_10** - *Includes the 10 most frequent families,* 74,655 *instances,* 11.8 *MB*
2. **fam_t_100** - *Includes the 100 most frequent families,* 227,191 *instances,* 36.3 *MB*
3. **fam_t_500** - *Includes the 500 most frequent families,* 292,596 *instances,* 46.7 *MB*
4. **typ_t_10** - *Includes the 10 most frequent types,* 310,355 *instances,* 50.5 *MB*
5. **typ_t_35** (full set) - *Includes the 35 most types families,* 328,337 *instances,* 53.5 *MB*

[2] http://www.cs.waikato.ac.nz/ml/weka/arff.html

6.4 Feature selection

In this section we present results and analysis of feature selection of the different subsets of data used in the thesis. Note that for the feature selection method *CfsSubsetEval*, the results from feature selection contain only the selection features, and not the features *and* the feature's merits. This is because the *CfsSubsetEval* utilizes another search algorithm than *InfoGain* and *ReliefF*. For the following subsections (one for each data subset) future selection is presented and some of the most relevant features are discussed. Note that what we refer to by 'most frequent family/type' is the family/type that are most frequent in *our* data. Also note that by # in the tables, we refer to the order of the attributes in the order that they are stated in the .arff file.

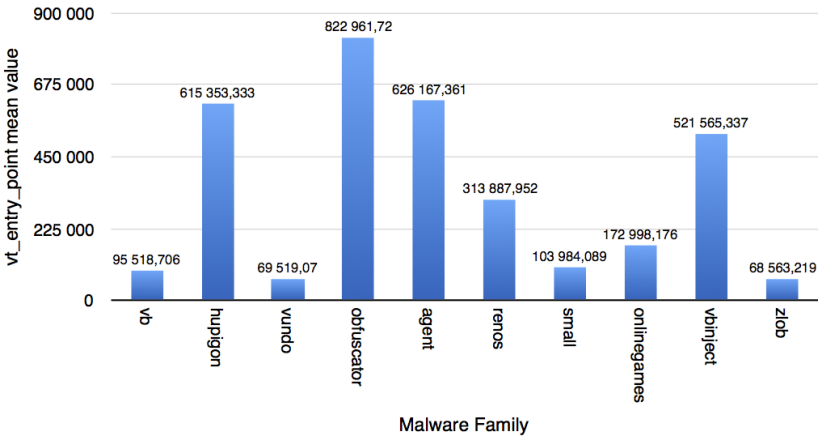
6.4.1 10 most frequent families

In this section we provide an overview of the results from feature selection on the data subset consisting of the 10 most frequent malware families in our data set, and a discussion of the results. Note that for the methods *InfoGain* and *ReliefF*, all features are included, but what we refer to as *selected* is a merit of above zero.

Cfs	**InfoGain**		**ReliefF**	
Attribute	*Average merit*	*#, Attribute*	*Average merit*	*#, Attribute*
vt_codesize	1.203 +- 0.012	5 vt_entry_point	0.047 +- 0	36 entropy
vt_res_langs	0.94 +- 0.002	6 vt_initDataSize	0.034 +- 0	11 pe_api
vt_res_types	0.937 +- 0.006	34 size_TOT	0.03 +- 0	13 pe_packer
vt_sections	0.917 +- 0.01	35 filesize	0.022 +- 0	12 pe_debug
vt_entry_point	0.909 +- 0.006	31 size_TEXT	0.017 +- 0.002	14 pe_library
vt_initDataSize	0.85 +- 0.004	1 vt_codesize	0.017 +- 0.002	29 pe_dll
vt_productName	0.849 +- 0.004	32 size_DATA	0.016 +- 0.001	2 vt_res_langs
vt_originalFileName	0.645 +- 0.001	11 pe_api	0.012 +- 0	8 vt_originalFileName
vt_unitializedDataSize	0.531 +- 0.004	36 entropy	0.012 +- 0.001	4 vt_sections
pe_api	0.43 +- 0.003	9 vt_unitializedDataSize	0.011 +- 0	30 pe_detected
pe_debug	0.393 +- 0.001	7 vt_productName	0.007 +- 0	3 vt_res_types
pe_dll	0.39 +- 0.002	4 vt_sections	0.007 +- 0	25 pe_cabinet
size_DATA	0.362 +- 0.001	2 vt_res_langs	0.006 +- 0	35 filesize
filesize	0.34 +- 0.001	14 pe_library	0.003 +- 0	31 size_TEXT
	0.317 +- 0.002	8 vt_originalFileName	0.003 +- 0	22 pe_log
	0.281 +- 0.002	3 vt_res_types	0.002 +- 0.001	34 size_TOT
	0.274 +- 0.001	12 pe_debug	0.002 +- 0	27 pe_registry
	0.253 +- 0.002	29 pe_dll	0.002 +- 0	24 pe_backup
	0.251 +- 0.002	13 pe_packer	0.002 +- 0	23 pe_webpage
	0.224 +- 0.001	10 vt_legalCopyright	0.002 +- 0	26 pe_data
	0.091 +- 0.001	30 pe_detected	0.001 +- 0	32 size_DATA
	0.045 +- 0.001	33 size_OBJ	0.001 +- 0	10 vt_legalCopyright
	0.008 +- 0	17 pe_executable	0.001 +- 0	16 pe_object
	0.003 +- 0	21 pe_database	0.001 +- 0	7 vt_productName
	0 +- 0	27 pe_registry	0 +- 0	21 pe_database
	0 +- 0	26 pe_data	0 +- 0	17 pe_executable
	0 +- 0	24 pe_backup	0 +- 0	20 pe_temporary
	0 +- 0	25 pe_cabinet	0 +- 0	6 vt_initDataSize
	0 +- 0	22 pe_log	0 +- 0	15 pe_autogen
	0 +- 0	23 pe_webpage	0 +- 0	1 vt_codesize
	0 +- 0	20 pe_temporary	0 +- 0	9 vt_unitializedDataSize
	0 +- 0	15 pe_autogen	0 +- 0	5 vt_entry_point
	0 +- 0	16 pe_object	0 +- 0	18 pe_text
	0 +- 0	28 pe_directories	0 +- 0	33 size_OBJ
	0 +- 0	19 pe_binary	0 +- 0	19 pe_binary
	0 +- 0	18 pe_text	0 +- 0	28 pe_directories

Table 8: Feature contribution for differentiation: 10 most frequent families

As we can see, several of the same features are selected by the same methods. Table 9 shows a complete list of these features. From *VirusTotal*, we have the feature *vt_entrypoint*, which is selected by *Cfs* and *InfoGain*. Another feature selected by two of the methods, namely *Cfs* and *InfoGain*, is entropy, a measure for the randomness in data. *vt_entry_point* and *entropy* are visualized in Figure 23 and 24 respectively.

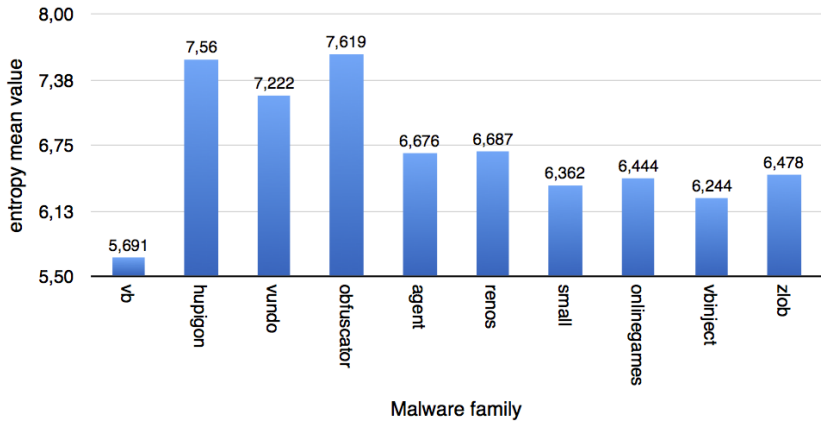


(a) Mean value of vt_entry point

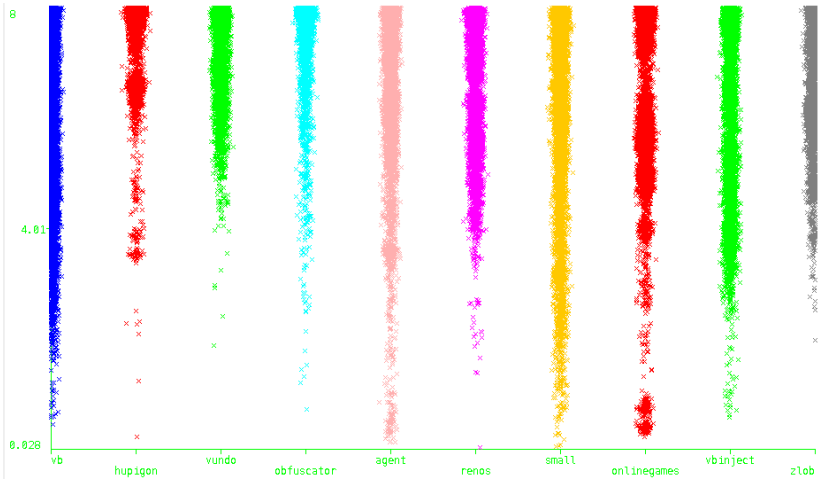
(b) Distribution of entry point by classes

Figure 23: 10 families: distribution of, and mean values for *vt_entry_point*

The observation of the apparently importance of the entry point is interesting, as a known methods for obfuscation is by modification of the file entry point, *entry point obfuscation*. From this it seems like different malware families utilize different methods for entry point obfuscation. Entropy however, is a measure for how uniform or non-uniform the data is. A strong encryption will result in higher entropy in the file. As we discussed earlier in the thesis one of the earliest forms for malware to avoid detection, was to encrypt the malicious part of the code. Subsequent obfuscation techniques builds on the same principle, which is why we should expect that widespread malware has an entropy considerably higher than regular text. According to Shannon, the entropy in the English language is between 0.6 and 1.3 [73].



(a) Mean value of entropy



(b) Distribution of entropy by classes

Figure 24: 10 families: distribution of, and mean values for *entropy*

Cfs and InfoGain	**Cfs and ReliefF**	**InfoGain and ReliefF**
vt_res_langs	vt_res_langs	size_TOT
vt_res_types	vt_res_types	filesize
vt_sections	vt_sections	size_TEXT
vt_entry_point	vt_productName	size_DATA
vt_initDataSize	vt_originalFileName	pe_api
vt_productName	pe_api	entropy
vt_originalFileName	pe_debug	vt_productName
vt_unitializedDataSize	pe_dll	vt_sections
pe_api	size_DATA	vt_res_langs
pe_debug	filesize	pe_library
pe_dll		vt_originalFileName
size_DATA		vt_res_types
filesize		pe_debug
		pe_dll
		pe_packer
		vt_legalCopyright
		pe_detected

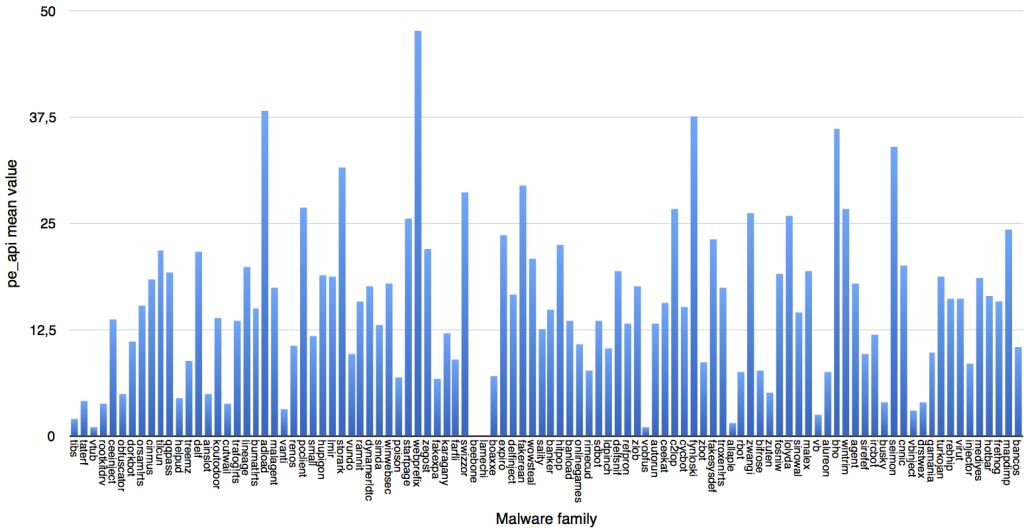
Table 9: 10 families: Features selected by two of the same methods for feature selection

6.4.2 100 most frequent families

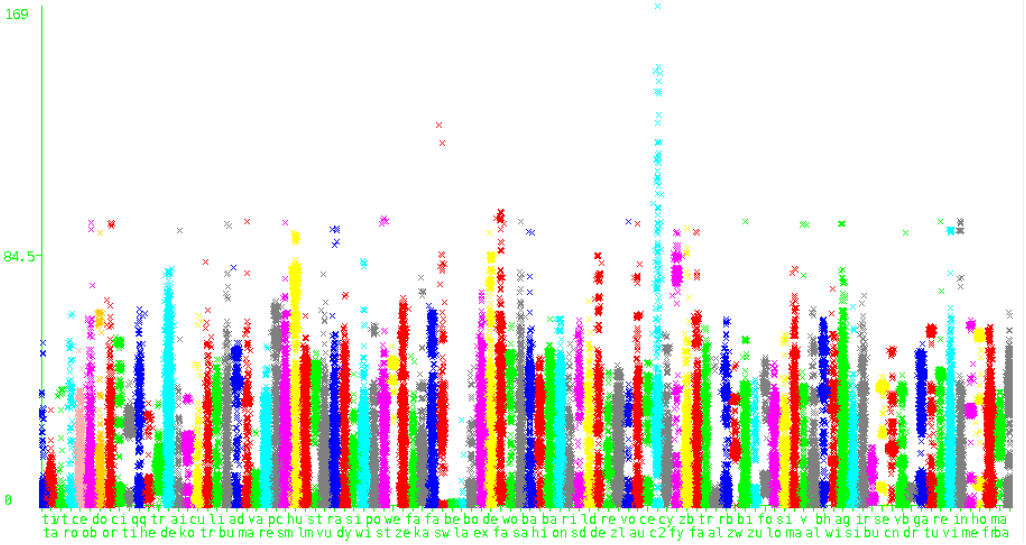
In this section we provide an overview of the results from feature selection on the data subset consisting of the 100 most frequent malware families in our data set, and a discussion of the results. Note that for the methods *InfoGain* and *ReliefF*, all features are included, but what we refer to as *selected* is a merit of above zero.

Cfs	**InfoGain**		**ReliefF**	
Attribute	*Average merit*	*#, Attribute*	*Average merit*	*#, Attribute*
vt_codesize	2.077 +- 0.009	5 vt_entry_point	0.057 +- 0	36 entropy
vt_res_langs	1.991 +- 0.004	6 vt_initDataSize	0.056 +- 0	13 pe_packer
vt_sections	1.957 +- 0.008	1 vt_codesize	0.046 +- 0.001	29 pe_dll
vt_entry_point	1.887 +- 0.004	32 size_DATA	0.031 +- 0.001	11 pe_api
vt_initDataSize	1.879 +- 0.011	31 size_TEXT	0.029 +- 0	30 pe_detected
vt_productName	1.825 +- 0.013	35 filesize	0.028 +- 0	12 pe_debug
pe_api	1.82 +- 0.014	34 size_TOT	0.014 +- 0	4 vt_sections
pe_packer	1.231 +- 0.003	11 pe_api	0.013 +- 0	8 vt_originalFileName
pe_library	1.22 +- 0.004	36 entropy	0.011 +- 0	2 vt_res_langs
pe_log	0.793 +- 0.001	4 vt_sections	0.009 +- 0	25 pe_cabinet
pe_dll	0.792 +- 0.002	14 pe_library	0.006 +- 0.001	3 vt_res_types
size_TEXT	0.768 +- 0.003	9 vt_unitializedDataSize	0.005 +- 0	35 filesize
size_DATA	0.63 +- 0.001	13 pe_packer	0.004 +- 0	22 pe_log
filesize	0.544 +- 0.001	12 pe_debug	0.003 +- 0	24 pe_backup
entropy	0.543 +- 0.001	2 vt_res_langs	0.003 +- 0	14 pe_library
	0.532 +- 0.001	7 vt_productName	0.003 +- 0	23 pe_webpage
	0.474 +- 0.001	10 vt_legalCopyright	0.003 +- 0	27 pe_registry
	0.436 +- 0.001	3 vt_res_types	0.002 +- 0	32 size_DATA
	0.373 +- 0.001	8 vt_originalFileName	0.002 +- 0	26 pe_data
	0.356 +- 0.001	30 pe_detected	0.002 +- 0	34 size_TOT
	0.326 +- 0.001	29 pe_dll	0.002 +- 0	31 size_TEXT
	0.204 +- 0.001	33 size_OBJ	0.001 +- 0	16 pe_object
	0.037 +- 0	17 pe_executable	0.001 +- 0	10 vt_legalCopyright
	0.01 +- 0	24 pe_backup	0.001 +- 0	7 vt_productName
	0.008 +- 0	27 pe_registry	0.001 +- 0	17 pe_executable
	0.007 +- 0	23 pe_webpage	0.001 +- 0	21 pe_database
	0.006 +- 0	25 pe_cabinet	0 +- 0	20 pe_temporary
	0.005 +- 0	22 pe_log	0 +- 0	15 pe_autogen
	0.001 +- 0	15 pe_autogen	0 +- 0	6 vt_initDataSize
	0 +- 0	21 pe_database	0 +- 0	1 vt_codesize
	0 +- 0	26 pe_data	0 +- 0	18 pe_text
	0 +- 0	20 pe_temporary	0 +- 0	5 vt_entry_point
	0 +- 0	19 pe_binary	0 +- 0	9 vt_unitializedDataSize
	0 +- 0	28 pe_directories	0 +- 0	33 size_OBJ
	0 +- 0	16 pe_object	0 +- 0	19 pe_binary
	0 +- 0	18 pe_text	0 +- 0	28 pe_directories

Table 10: Feature contribution for differentiation: 100 most frequent families

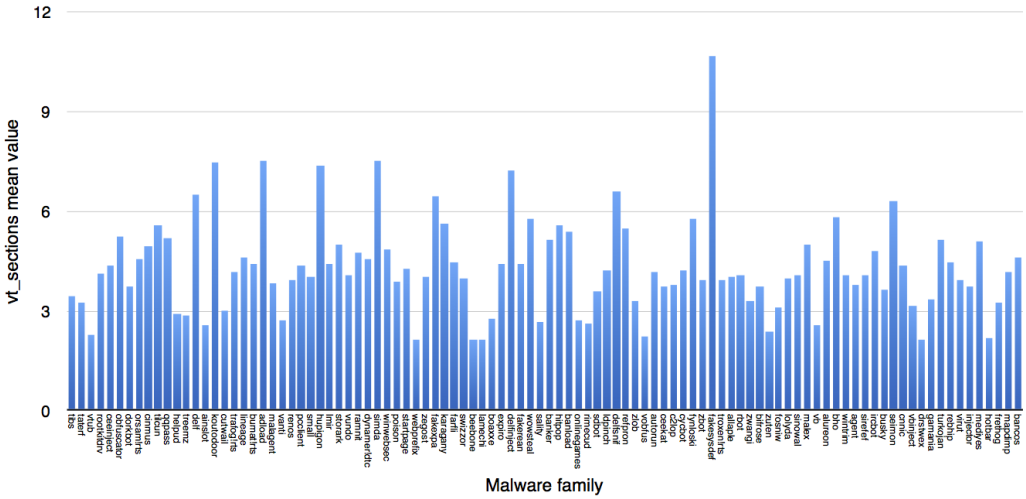


(a) Mean value of *pe_api*

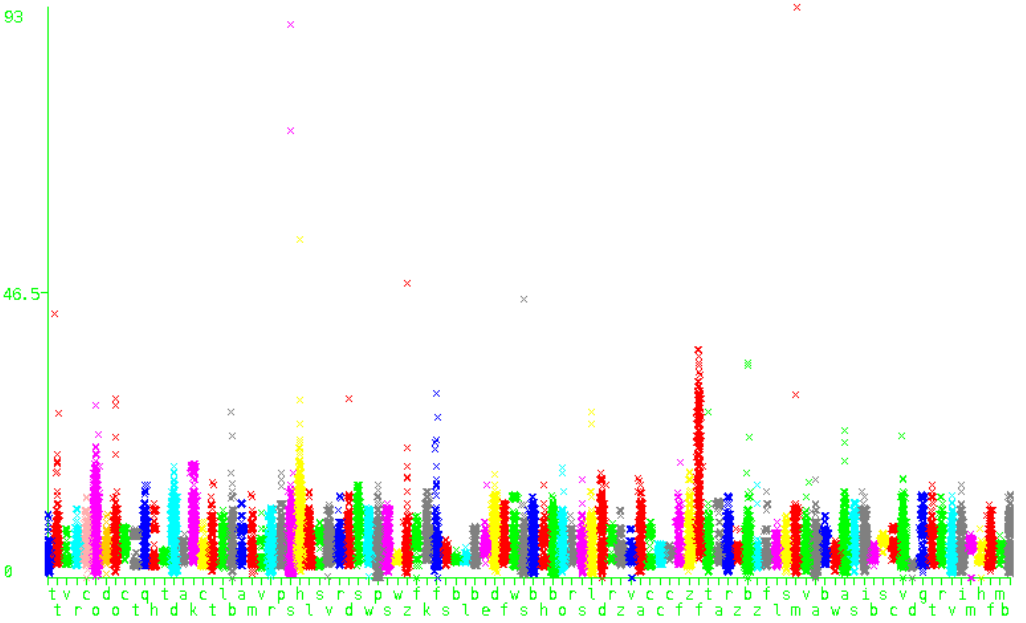


(b) Distribution of *pe_api* by classes

Figure 25: 100 families: distribution of, and mean values for *pe_api*



(a) Mean value of *vt_sections*



(b) Distribution of *vt_sections* by classes

Figure 26: 100 families: distribution of, and mean values for *vt_sections*

Cfs and InfoGain	**Cfs and ReliefF**	**InfoGain and ReliefF**
vt_codesize	vt_res_langs	size_DATA
vt_res_langs	vt_sections	size_TEXT
vt_sections	vt_productName	filesize
vt_entry_point	pe_api	size_TOT
vt_initDataSize	pe_packer	pe_api
vt_productName	pe_library	entropy
pe_api	pe_log	vt_sections
pe_packer	pe_dll	pe_library
pe_library	size_TEXT	pe_packer
pe_dll	size_DATA	pe_debug
size_TEXT	filesize	vt_res_langs
size_DATA	entropy	vt_productName
filesize		vt_legalCopyright
entropy		vt_res_types
		vt_originalFileName
		pe_detected
		pe_dll
		pe_executable
		pe_backup

Table 11: 100 families: Features selected by two of the same methods for feature selection

Two other important features are the *pe_api* and *vt_sections*. Both features was selected by by all three methods for feature selection. What we can derive from this is that the different malware families differs in the the numbers of API-calls made in the file and the number of PE-sections in the files. We have previously discussed the structure of a PE file. While there are a number of sections mandatory for a PE file to run, while there are possible to include more (maximum $2^{16}-1 = 65,535$) [38]. The number of API calls within the different families have a certain variety as well, from which we can tell that similar malware families will have a similar number of API calls. Since the different families have different capabilities and functionality, our assumption that API calls will be of interest in labeling malware is strengthened after this observation.

6.4.3 500 most frequent families

In this section we provide an overview of the results from feature selection on the data subset consisting of the 500 most frequent malware families in our data set, and a discussion of the results. Note that for the methods *InfoGain* and *ReliefF*, all features are included, but what we refer to as *selected* is a merit of above zero.

Cfs	**InfoGain**		**ReliefF**	
Attribute	*Average merit*	*#, Attribute*	*Average merit*	*#, Attribute*
vt_codesize	2.188 +- 0.013	6 vt_initDataSize	0.063 +- 0	13 pe_packer
vt_res_langs	2.169 +- 0.029	5 vt_entry_point	0.061 +- 0	29 pe_dll
vt_sections	2.15 +- 0.01	1 vt_codesize	0.06 +- 0	36 entropy
vt_entry_point	2.118 +- 0.009	32 size_DATA	0.036 +- 0	30 pe_detected
vt_initDataSize	1.899 +- 0.022	31 size_TEXT	0.034 +- 0.002	11 pe_api
vt_productName	1.792 +- 0.015	34 size_TOT	0.032 +- 0	12 pe_debug
pe_api	1.736 +- 0.012	35 filesize	0.013 +- 0	4 vt_sections
pe_packer	1.417 +- 0.003	11 pe_api	0.011 +- 0	2 vt_res_langs
pe_library	1.195 +- 0.005	36 entropy	0.009 +- 0	8 vt_originalFileName
pe_executable	0.9 +- 0.003	14 pe_library	0.009 +- 0	25 pe_cabinet
pe_dll	0.871 +- 0.006	9 vt_unitializedDataSize	0.006 +- 0.001	3 vt_res_types
size_TEXT	0.844 +- 0.001	4 vt_sections	0.005 +- 0	35 filesize
size_DATA	0.676 +- 0.001	13 pe_packer	0.004 +- 0	22 pe_log
filesize	0.606 +- 0.001	7 vt_productName	0.003 +- 0	24 pe_backup
entropy	0.602 +- 0.001	12 pe_debug	0.003 +- 0	14 pe_library
	0.596 +- 0	2 vt_res_langs	0.002 +- 0	23 pe_webpage
	0.558 +- 0.003	10 vt_legalCopyright	0.002 +- 0	27 pe_registry
	0.488 +- 0.001	3 vt_res_types	0.002 +- 0	32 size_DATA
	0.434 +- 0.004	8 vt_originalFileName	0.002 +- 0	34 size_TOT
	0.385 +- 0	30 pe_detected	0.002 +- 0	26 pe_data
	0.332 +- 0	29 pe_dll	0.002 +- 0	31 size_TEXT
	0.235 +- 0.002	33 size_OBJ	0.001 +- 0	10 vt_legalCopyright
	0.03 +- 0	17 pe_executable	0.001 +- 0	16 pe_object
	0.009 +- 0	24 pe_backup	0.001 +- 0	7 vt_productName
	0.007 +- 0	27 pe_registry	0.001 +- 0	17 pe_executable
	0 +- 0	26 pe_data	0.001 +- 0	21 pe_database
	0.001 +- 0.002	23 pe_webpage	0 +- 0	20 pe_temporary
	0 +- 0	25 pe_cabinet	0 +- 0	6 vt_initDataSize
	0 +- 0	21 pe_database	0 +- 0	15 pe_autogen
	0 +- 0	22 pe_log	0 +- 0	5 vt_entry_point
	0 +- 0	20 pe_temporary	0 +- 0	1 vt_codesize
	0 +- 0	15 pe_autogen	0 +- 0	33 size_OBJ
	0 +- 0	16 pe_object	0 +- 0	18 pe_text
	0 +- 0	28 pe_directories	0 +- 0	9 vt_unitializedDataSize
	0 +- 0	19 pe_binary	0 +- 0	19 pe_binary
	0 +- 0	18 pe_text	0 +- 0	28 pe_directories

Table 12: Feature contribution for differentiation: 500 most frequent families

Considering the results from *InfoGain* we can see that *vt_initDataSize* is considered to be one of the most importance when we increased the number of malware families to 500. As previously mentioned, this feature was also considered to one of the most important features in binary classification based on PE-header information [51]. This feature is selected by *Cfs* and *ReliefF* as well, strengthening our assumption that this feature is in fact important. The reason why this is of importance might be that since the different capabilities requires different program code, the entire programming style may also be different as well. Resulting in that for some malware families to be most effective, different numbers of data fields must be initialized.

vt_initDataSize	
Minimum	0
Maximum	4294966784
Mean value	2215973.021
Standard Deviation	83972022.041

Table 13: 500 families: statistics on *vt_initDataSize*

Figure 27: Distribution of *vt_initDataSize* by family

Cfs and InfoGain	Cfs and ReliefF	InfoGain and ReliefF
vt_codesize	vt_res_langs	size_DATA
vt_res_langs	vt_sections	size_TEXT
vt_sections	vt_productName	size_TOT
vt_entry_point	pe_api	filesize
vt_initDataSize	pe_packer	pe_api
vt_productName	pe_library	entropy
pe_api	pe_executable	pe_library
pe_packer	pe_dll	vt_sections
pe_library	size_TEXT	pe_packer
pe_executable	size_DATA	vt_productName
pe_dll	filesize	pe_debug
size_TEXT	entropy	vt_res_langs
size_DATA		vt_legalCopyright
filesize		vt_res_types
entropy		vt_originalFileName
		pe_detected
		pe_dll
		pe_executable
		pe_backup
		pe_registry
		pe_data
		pe_webpage

Table 14: 500 families: Features selected by two of the same methods for feature selection

6.4.4 10 most frequent types

In this section we provide an overview of the results from feature selection on the data subset consisting of the 10 most frequent malware types in our data set, and a discussion of the results. Note that for the method *InfoGain*, all features are included, but what we refer to as *selected* is a merit of above zero.

Cfs	**InfoGain**	
Attribute	*Average Merit*	*#, Attribute*
vt_codesize	0.555 +- 0.004	5 vt_entry_point
vt_entry_point	0.487 +- 0.004	31 size_TEXT
vt_initDataSize	0.465 +- 0.003	34 size_TOT
vt_productName	0.43 +- 0.002	32 size_DATA
vt_unitializedDataSize	0.417 +- 0.002	35 filesize
vt_legalCopyright	0.401 +- 0.001	6 vt_initDataSize
pe_dll	0.383 +- 0.002	1 vt_codesize
size_TEXT	0.185 +- 0.001	36 entropy
size_DATA	0.172 +- 0.001	11 pe_api
size_OBJ	0.169 +- 0.001	9 vt_unitializedDataSize
filesize	0.098 +- 0	7 vt_productName
	0.093 +- 0	10 vt_legalCopyright
	0.093 +- 0	14 pe_library
	0.088 +- 0	29 pe_dll
	0.08 +- 0	4 vt_sections
	0.077 +- 0	13 pe_packer
	0.065 +- 0	12 pe_debug
	0.061 +- 0.001	33 size_OBJ
	0.058 +- 0	8 vt_originalFileName
	0.056 +- 0	2 vt_res_langs
	0.051 +- 0	3 vt_res_types
	0.019 +- 0	30 pe_detected
	0.004 +- 0	17 pe_executable
	0.001 +- 0	24 pe_backup
	0.001 +- 0	27 pe_registry
	0.001 +- 0	23 pe_webpage
	0 +- 0	21 pe_database
	0 +- 0	25 pe_cabinet
	0 +- 0	15 pe_autogen
	0 +- 0	22 pe_log
	0 +- 0	28 pe_directories
	0 +- 0	16 pe_object
	0 +- 0	26 pe_data
	0 +- 0	20 pe_temporary
	0 +- 0	19 pe_binary
	0 +- 0	18 pe_text

Table 15: Feature contribution for differentiation: 10 most frequent types

Cfs and InfoGain
vt_codesize
vt_entry_point
vt_initDataSize
vt_productName
vt_unitializedDataSize
vt_legalCopyright
pe_dll
size_TEXT
size_DATA
size_OBJ
filesize

Table 16: 10 families: Features selected by *Cfs* and *InfoGain*

6.4.5 35 most frequent types (Full feature set)

In this section we provide an overview of the results from feature selection on the data subset consisting of all the malware families in our data set, and a discussion of the results. Note that for the method *InfoGain*, all features are included, but what we refer to as *selected* is a merit of above zero.

Cfs	**InfoGain**	
Attribute	*Average Merit*	*#, Attribute*
vt_codesize	0.639 +- 0.005	5 vt_entry_point
vt_entry_point	0.568 +- 0.003	31 size_TEXT
vt_initDataSize	0.545 +- 0.005	34 size_TOT
vt_productName	0.496 +- 0.005	35 filesize
vt_unitializedDataSize	0.495 +- 0.003	32 size_DATA
vt_legalCopyright	0.463 +- 0.002	6 vt_initDataSize
pe_api	0.461 +- 0.002	1 vt_codesize
pe_dll	0.247 +- 0.001	11 pe_api
size_TEXT	0.241 +- 0.001	36 entropy
size_DATA	0.221 +- 0.001	9 vt_unitializedDataSize
size_OBJ	0.138 +- 0.001	10 vt_legalCopyright
size_TOT	0.132 +- 0	14 pe_library
filesize	0.13 +- 0	7 vt_productName
	0.102 +- 0	4 vt_sections
	0.097 +- 0	13 pe_packer
	0.092 +- 0	12 pe_debug
	0.092 +- 0	29 pe_dll
	0.077 +- 0	8 vt_originalFileName
	0.072 +- 0.001	33 size_OBJ
	0.072 +- 0	3 vt_res_types
	0.072 +- 0	2 vt_res_langs
	0.044 +- 0	30 pe_detected
	0.004 +- 0	17 pe_executable
	0.001 +- 0	24 pe_backup
	0.001 +- 0	27 pe_registry
	0.001 +- 0	23 pe_webpage
	0 +- 0	26 pe_data
	0 +- 0	25 pe_cabinet
	0 +- 0	21 pe_database
	0 +- 0	22 pe_log
	0 +- 0	15 pe_autogen
	0 +- 0	20 pe_temporary
	0 +- 0	16 pe_object
	0 +- 0	28 pe_directories
	0 +- 0	19 pe_binary
	0 +- 0	18 pe_text

Table 17: Feature contribution for differentiation: all 35 types

Cfs and InfoGain
vt_codesize
vt_entry_point
vt_initDataSize
vt_productName
vt_unitializedDataSize
vt_legalCopyright
pe_api
pe_dll
size_TEXT
size_DATA
size_OBJ
size_TOT
filesize

Table 18: All (35) types: Features selected from *Cfs* and *InfoGain*

Discussion

In the previous sections we have observed that the majority of the features selected is the same whether we are classifying into malware type or malware family. This points to that the majority of the malware families are to a large degree limited to a specific malware type. From this, we can also assume that the classification accuracy on family and type will be quite similar.

fam10	**fam100**	**fam500**	**t10**	**t35**
vt_res_langs	vt_codesize	vt_codesize	**vt_codesize**	**vt_codesize**
vt_res_types	**vt_res_langs**	**vt_res_langs**	**vt_entry_point**	**vt_entry_point**
vt_sections	**vt_sections**	**vt_sections**	**vt_initDataSize**	**vt_initDataSize**
vt_entry_point	**vt_entry_point**	**vt_entry_point**	**vt_productName**	**vt_productName**
vt_initDataSize	**vt_initDataSize**	**vt_initDataSize**	**vt_unitializedDataSize**	**vt_unitializedDataSize**
vt_productName	**vt_productName**	**vt_productName**	**vt_legalCopyright**	**vt_legalCopyright**
vt_originalFileName	**pe_api**	**pe_api**	**pe_dll**	pe_api
vt_unitializedDataSize	pe_packer	pe_packer	**size_TEXT**	**pe_dll**
pe_api	pe_library	pe_library	**size_DATA**	**size_TEXT**
pe_debug	**pe_dll**	pe_executable	**size_OBJ**	**size_DATA**
pe_dll	size_TEXT	**pe_dll**	**filesize**	**size_OBJ**
size_DATA	**size_DATA**	size_TEXT		size_TOT
filesize	**filesize**	**size_DATA**		**filesize**
	entropy	**filesize**		
		entropy		

Table 19: Features selected by *Cfs* and *InfoGain*

Table 19 shows the features selected by both *Cfs* and *InfoGain* on each data subset. The features in bold are the ones that was selected by both methods on all three data subsets, for family classification, and for both data subsets for type classification. An interesting observation considering the features selected for family classification is that the features selected converges with the increase in number of classes.

Algorithm	**Settings**
Cfs	−P1 − E1
BestFirst	−D1 − N5
InfoGain	n/a
Ranker	−T − 1.7976931348623157E308 − N − 1
ReliefF	−M − 1 − D1 − K10
Ranker	−T − 1.7976931348623157E308 − N − 1

Table 20: Weka feature selection algorithms settings in experiments

Table 20 presents the algorithm settings and the used search algorithm in the feature selection process, which also are the default settings for Weka. We have not explored the influence of the search algorithms itself, which may be field of interest in future research.

6.5 Classification

This sections provides the results of classification and discussion of the results. All experiments are run with 5-fold cross validation. Experiments with the *resampling* filtering method was only performed on the smallest data subset. The reason for this is that we try to answer the relevance of the features with respect to algorithms for feature selection and classification.

Data set	Feature Selection	Resample filtering	RandomTree	J48(C4.5)	RandomForest	NaiveBayes	BayesNet	IBk, k=1	MLP	SVM
Family_t_10	Full feature set	F	81,5632 %	83,2871 %	88,7965 %	23,9408 %	72,6462 %	65,3084 %	42,3133 %	32,9824 %
		T	90,9651 %	89,0536 %	94,4625 %	25,7143 %	74,7760 %	83,7640 %	*n/a*	*n/a*
	Cfs	F	83,1250 %	83,9180 %	88,1990 %	20,8760 %	70,8285 %	78,9914 %	40,7206 %	32,7962 %
		T	91,4688 %	88,7348 %	93,8812 %	22,8906 %	72,8926 %	89,5318 %	*n/a*	*n/a*
	InfoGain(14)	F	82,9429 %	84,2100 %	88,9572 %	23,9354 %	72,6462 %	80,0107 %	51,5947 %	janw
		T	91,7259 %	89,3256 %	94,6059 %	25,6580 %	74,7760 %	90,3730 %	*n/a*	*n/a*
	ReliefF(t=0)	F	81,3636 %	82,7821 %	87,2279 %	39,3182 %	65,9996 %	77,0585 %	41,0703 %	41,2283 %
		T	90,9544 %	87,9512 %	93,6173 %	39,9277 %	67,8334 %	88,7898 %	*n/a*	*n/a*
Family_t_100	Full feature set	F	74,5012 %	76,6250 %	81,3078 %	20,9062 %	61,9056 %	*n/a*	11,8116 %	*n/a*
	Cfs	F	77,0682 %	*n/a*	77,8019 %	21,0008 %	61,6851 %	*n/a*	18,1451 %	*n/a*
	InfoGain(t=0.5)	F	77,2693 %	*n/a*	77,8204 %	20,1905 %	60,9452 %	*n/a*	15,9034 %	*n/a*
	ReliefF	F	72,6204 %	75,3771 %	79,4565 %	23,9406 %	57,3495 %	*n/a*	17,4743 %	*n/a*
Family_t_500	Full feature set	F	69,9446 %	72,4210 %	*n/a*	16,1718 %	57,2137 %	*n/a*	*n/a*	*n/a*
	Cfs	F	72,6411 %	73,2847 %	*n/a*	16,7610 %	56,8853 %	*n/a*	11,5260 %	*n/a*
	InfoGain(t=0.7)	F	72,0567 %	72,4487 %	*n/a*	13,7654 %	54,1846 %	*n/a*	6,8251 %	*n/a*
	ReliefF	F	*n/a*	*n/a*	*n/a*	*n/a*	*n/a*	*n/a*	*n/a*	*n/a*
Type_t_10	Full feature set	F	71,8719 %	73,6789 %	78,6000 %	10,4316 %	51,3618 %	*n/a*	*n/a*	*n/a*
	Cfs	F	73,5722 %	73,8200 %	77,1639 %	7,3857 %	51,1862 %	*n/a*	*n/a*	*n/a*
	InfoGain(t=0.1)	F	73,1578 %	73,6099 %	77,5177 %	4,9063 %	49,9312 %	*n/a*	*n/a*	*n/a*
	ReliefF	F	*n/a*	*n/a*	*n/a*	*n/a*	*n/a*	*n/a*	*n/a*	*n/a*
Type_t_35(full)	Full feature set	F	70,6347 %	72,6272 %	77,8797 %	2,3576 %	48,4487 %	*n/a*	27,2826 %	*n/a*
	Cfs	F	73,1995 %	73,7480 %	77,3324 %	1,9894 %	48,6290 %	*n/a*	25,1683 %	*n/a*
	InfoGain(t=0.1)	F	73,1706 %	73,8254 %	77,9687 %	1,9785 %	48,4140 %	*n/a*	25,7842 %	*n/a*
	ReliefF	F	*n/a*	71,7842 %	76,4943 %	*n/a*	*n/a*	*n/a*	28,0675 %	*n/a*

Figure 28: Classification results

Figure 28 depicts the classification scores for the different methods on the different data subsets. The highest scores are highlighted in green and the lowest ones in blue.

Discussion

J48(C4.5), RandomTree and RandomForest did perform significantly best opposed to the classifiers; NaiveBayes, BayesNet, MLP(ANN) and SVM. While other research have achieved very high classification scores with the ones that performed bad in our work, we will argue that this is due to the number of classes in our work, as opposed to the more common implementation of *binary classification*.

Algorithm	Settings
J48	$-$C0.25 $-$ M4
RandomTree	$-$K0 $-$ M1.0 $-$ V0.001 $-$ S1
MLP(ANN)	$-$L0.3 $-$ M0.2 $-$ N500 $-$ V0 $-$ S0 $-$ E20 $-$ H3
RandomForest	$-$I10 $-$ K0 $-$ S1
SVM	$-$S0 $-$ K2 $-$ D3 $-$ G0.0 $-$ R0.0 $-$ N0.5 $-$ M40.0 $-$ C1.0 $-$ E0.001 $-$ P0.1 $-$ seed1
NaiveBayes	-
BayesNet	$-$D $-$ QK2 $-$ P1 $-$ SBAYES $-$ E $-$ A0.5
IKk(KNN), K = 1	$-$K1 $-$ W0 $-$ A $<$ linearsearch $>$ $-$AEuclideanDistance

Table 21: Weka classification algorithms settings in experiments

Table 21 lists the different settings used for experiments. Note that for RandomForest method, we discussed in Section 5.5.2 that the usual number of trees is 100[13]. In our experiments, we were only able use 100 trees (`I 10`), on RF-experiments performed on the smallest data subset, containing the samples belonging to the 10 most frequent families. This was due to the computational time required by the RandomForest method, so to be able to perform the experiments with this classifier on the other sets, we lowered the number of trees to 10. A clear pattern previously anticipated can be seen in Figure 28; the difference in classification results in the two decision tree methods and the random forest methods are almost identical whether we are labeling malware families or malware types. The most representable comparison will arguably be the between the the largest data subset for each class.

RandomTree	**Family**(500)	**Type**(Full)
Full feature set	69.9446%	70.6347%
Cfs	72.6411%	73.1995%
Infogain	72.0567%	73.1706%

Table 22: Comparing classification accuracy between type and family

In Figure 28, we have highlighted the best performing classifier for each feature selection method be light green. The best score and thus feature selection method is highlighted in dark green. What we can derive from the observation of the results is that feature selection in fact do increase our classification accuracy in the majority of the experiments. Another takeaway from this is that the implemented methods performs consistently good on the data, which argues for the reliability of features. The highest increase in accuracy from feature selection we have observed is in the experiments on the data subset *fam_t_10* with NaiveBayes classifier, where accuracy with *ReliefF* increased to 39.32% from 23.94% with all features.

7 Conclusion & future work

In this thesis, we have shown that it is possible to extract features that may be useful to estimate what type and family a certain malware belongs to. The tools used in the project are freely available, and the overall analysis process is significantly more efficient compared to dynamic analysis. The number of samples in the data set is very high, and to the author's knowledge, considerably larger than any other similar research conducted. This makes the argument that the results can be considered as non-trivial. To be able to perform analysis, both feature selection and classification, subsets of the complete dataset were created. This is because family classification was impossible to perform in Weka due to the large number of distinct classes (10,000+ families). Yet, this is not a limitation in Weka *per se*, but rather in the Machine Learning methods itself since such large multinomial classification problems are non-trivial and limitations of most machine learning methods are not studied from this perspective. As of today, there exist no other comprehensive tools such as Weka. A challenge in the work is that the occurrences of malware in our data set is very unbalanced, which is a limitation in the validity of the results, as opposed to a balanced, real-world dataset, which to the author's knowledge do not exist.

To be able to perform malware family classification, we decided to generate subsets of the processed data. The largest subset containing malware families contained samples belonging to the 500 most frequently represented malware families in our data, and a total of $292,596$ samples. Meaning that the remaining $35,741$ samples are distributed over the remaining $9,862$ malware families. The average ratio for sample per class is then $\frac{292,596}{500} =\sim 585.2$ samples per class, as opposed to $\frac{35,741}{9,862} =\sim 3.62$ samples per class in average for the samples we have left out of the experiments.

The number of malware types was 35, but to observe what influence the number of classes would have on the classification accuracy, we decided to create a subset of this as well, containing samples belonging to the 10 most frequent malware types in our data.

7.1 Theoretical Implications

The thesis aims to fill a hole in the research area by performing a large scale analysis of features derived from free and open-source tools. While the majority of previous research has been focused on the approach of binary analysis, we could also derive some assumptions from this regarding features of importance and indications on performance of different classifiers for multinominal classification.

From the different experiments considering feature selection, we observed that a number of features was selected by the same feature selection methods on the different data subsets. As a result, these findings can be considered to be indicators of compromise in regards to intrusion detection. A challenge however, is that none of the features alone will provide a reliable indicator, instead, they must be combined with several features in a heuristic approach for types and families.

In light of the "*no free lunch*"-theorem, which tells us that we can not assume a classifier to be superior to another if we have no prior knowledge to the data, we can see that

this applies to the experiments conducted in our work as well. We cannot conclude that the combination of the methods that yielded the best result on our data will be universal, even though *RandomForest* exclusively performed best in our experiments. This may not be universal at all, but nevertheless strengthens the assumption that classifiers originally designed to handle multi-class problems performs best with an increasing number of classes, even though there have been a lot of work in tweaking originally two-class classifiers to handle multi-class problems. The observed superiority of tree-based classifiers on our data can be assumed to be so due the granularity of these algorithms, which are very complex compared to other types of algorithms.

Research question 1

What features that can extracted from the static analysis tools PEframe *and* VirusTotal*, will be most relevant for distinguishing malware into type and family?*

Through the experiments described the thesis, by using several algorithms for feature selection and classification, we are able to to argue for the significance of the findings regarding the features. What we consider the most relevant features for distinguishing malware on type and/or family are the ones which are considered by different feature selection methods to have good merit. In addition we consider the features to be of good relevance if they are also selected by the same methods for feature selection multiple times. Table 19 in the previous chapter presents an overview of the features selected by all methods for feature selection on the different data subsets. *Entropy* of the file was considered by two of our methods to be of high relevance towards family classification, as a high entropy value (close to 8) indicates how much of the file contents that are encrypted. Initialized data size (*vt_initDataSize*) and the file entry point (*vt_entrypoint*) was considered by all implemented methods to be highly relevant for classification for both family and type. The initialized data size may reflect on the malware's capabilities, as different functionality may allow or require that different variables cannot be set on forehand. A file's entry point refers to the byte-offset in the file where the code execution start, and will reflect upon both the capabilities of the malware families and types, and also on how the malicious code is obfuscated.

From the experiments conducted regarding feature selection, we observe that a majority of the same features are considered to be of high relevance, not descending in whether we classify on type or family, which suggest that the majority of the malware families are limited to a certain malware type.

Research question 2

What accuracy can be achieved with features derived from the static analysis tools PEframe *and* VirusTotal*?*

To investigate the achievable classification accuracy, we have, in respect to the "*no free lunch*" theorem, conducted experiments using a variety of classification algorithms. Tree-based methods, *RandomForest*, *J48(C4.5)* and *RandomTrees* achieved the best accuracy, exclusively, and can be explained by the granularity of these methods, which also was designed to be applied on multi-class classification. The highest accuracy was achieved on the smallest data subset, containing malware from the 10 most frequent families, and then, on the larger subsets, the accuracy converges. For the *RandomForest* algorithm, accuracy on the smallest subset was $\sim 87\%-89\%$, and $\sim 76.5\%-78.5\%$ for the remaining data subsets.

Research question 3

Which methods for feature selection and classification performs the best on the features constructed?

Considering the *RandomForest* algorithm which yield the best accuracy in our experiments, *InfoGain* and *ReliefF* contributed to the highest accuracy score on all data subsets. This also applies to the *RandomTree*-algorithm, excluding the **fam_t_10** subset. We consider the larger subsets to be most significant when arguing for the validity of the results, and conclude that *InfoGain* and *ReliefF* performs the best on our data. There were, however variations on which of the two contributing to the best classification accuracy, and from this, we can not conclude which is better suited for large-scale malware analysis.

From our experiments, which yielded the highest classification accuracy for tree-based methods, we observe that especially *RandomForest* performed the best on our data. In regards to the number of samples, *RandomForest* is probably superior to *J48(C4.5)* due to the pruning performed by the latter, which will not generalize well on such complex datasets. *RandomForest* creates a number of trees, where each tree get its vote to decide which class the sample should be labeled as, making forests more complex. This is probably the main reason why *RandomForest* is the algorithm that scales best to our data.

7.2 Practical Considerations

Recreating the experiments are generally feasible for the most part. Both *VirusTotal* and *PEframe* are free to use, and so is the software required to carry out the experiments itself. There are two practical problems related to this: while computation is a trivial challenge, anyone who seeks to recreate the experiments should be aware of the computational cost and memory usage on forehand to tailor the experimental environment to this. Weka is a publicly available tool which is significantly comprehensive in terms of implementations. Optimization and parallel implementation should be a focus area in the future for us to be better able to perform similar large-scale analysis of malware. Weka does not offer support for multi-core processing, a problem that should be investigated when performing large-scale analysis in the future.

Another significant challenge for recreating the experiments, is the availability of the data set itself, which, at the time of writing, is not publicly available.

All the tools are free and open-source, except from *VirusTotal*. *VirusTotal* offers an API, which one can connect to to perform analysis for a range of files. The public API, however is limited, both in terms of scans per minute and scans per day. A private API is also offered, which you have to apply for access for, to make acquisition of raw characteristics significantly more efficient.

Also, importing data might be an issue if the database size exceeds the amount available physical memory, as was the case in this work. There are also limitations for this in PHP, and is most easily avoided by using another tool for the importing procedure, such as BigDump [70] used in this work.

Features of importance

The features considered to be the most relevant for multi-nominal malware classification may be of interest to malware analysts and providers of antivirus solutions and intrusion detection systems as possible indicators of compromise.

7.3 Further Research

For future research, we propose to implement the framework proposed in the thesis on other, existing datasets. Hopefully, our research will motivate future work on malware classification in terms of type and/or family.

Data set balance

An important consideration is the imbalanced data set, as well as the lack of certain classes in our dataset, e.g. ransomware has been observed in the wild frequently as of lately, but is not represented in our data set. An important problem that should be solved for strengthening the future work in this field is most certainly to generate an empirically balanced data set to be used for benchmarking the different approaches. A challenge for making a dataset that represents the real world as good as possible, is the rapid growth and diversity of malware, which is an area that should be further investigated to make large-scale classification of malware reliable in the future.

Include benign files

The data considered in this thesis consist of malware exclusively, and for our methodology to be applicable in e.g. antivirus solution in the future, there should be explored how classification will perform when including malicious samples in the examined data. A problem with this, will probably be the family classification: for benign features to be implemented in our methodology, there must be created a taxonomy of benign samples as well. Another approach might also to be to disregard the families as well, and to give the benign samples the class label "benign".

Computation

We have mentioned through the thesis, computation, as a problem for us. From this, we advice to use more powerful computational platforms to perform similar research, to then be able to, amongst others: implement our methodology in direction towards "best-practices", with e.g. higher number of k in cross validation, higher number of trees in *RandomForest* algorithm, and to run experiments on subsets including an even higher number of classes than we used.

Filtering

In the final stages of our work, we decided to explore other filtering methods, even though this is not a part of our research. A method of interest is the *resample* filtering which have included the classification scores presented in Figure 28 for the smallest data subset (**fam_t_10**). This method works by that a subset of the dataset is randomly generated [74].

Unsupervised learning

Due to the large data set and thus computational time required for the experiments, we decided to limit our work to focus on classification exclusively. In the future, clustering algorithm should be investigated in this context as well.

Adjusting implemented methods

We have used the Weka standard settings for each algorithm throughout the experiments, except for the size of the *RandomForest*. Tweaking the settings of the different algorithms should thus be examined in the future, as this might yield higher classification accuracy. While exploring this, the performance must be validated on several different data sets.

Otherwise, if one were to achieve significantly higher accuracy than other research, one could not argue that these settings will be generalizable to other datasets. In addition we propose that the influence of the different search algorithms for the different algorithms for feature selection algorithms should be explored in the future, which we have decided to not focus on, and use the Weka default algorithms for feature selection.

Documentation of VirusTotal *and* PEframe

For us to argue for the reliability of the features and thus the validity of the result, the origin of the data should be closely documented, which is not the case with *PEframe* nor *VirusTotal*, which is an area of interest as well. Subsequently we observe that the majority of the features from both tools are self-explanatory, but nevertheless, a documentation should exist.

Bibliography

[1] Pfleeger, C. P. & Pfleeger, S. L. 2002. *Security in Computing*. Prentice Hall Professional Technical Reference, 3rd edition.

[2] Mehta, L. Rootkits: User mode. *Last visited: 13/12/15,* `http://resources.infosecinstitute.com/rootkits-user-mode-kernel-mode-part-1/#article`.

[3] Szor, P. 2005. *The art of computer virus research and defense*. Pearson Education.

[4] You, I. & Yim, K. 2010. Malware obfuscation techniques: A brief survey. In *2010 International conference on broadband, wireless computing, communication and applications*, 297–300. IEEE.

[5] Yan, W., Zhang, Z., & Ansari, N. 2008. Revealing packed malware. *Security & Privacy, IEEE*, 6(5), 65–69.

[6] Microsoft Malware Protection Center. Naming malware. *Last visited 13/12/15,* `http://www.microsoft.com/security/portal/mmpc/shared/malwarenaming.aspx`.

[7] Jain, A. K., Duin, R. P., & Mao, J. 2000. Statistical pattern recognition: A review. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 22(1), 4–37.

[8] Kingravi, H. A. Visualizing data in the space it resides. *Pindrop Security Blog Last visited 19/11/2015* `http://www.pindropsecurity.com/visualizing-data-in-the-space-it-resides-in/`.

[9] Petersen, J. P. K-means. *PyPR v0.1rc3 documentation Last visited 19/11/2015* `http://pypr.sourceforge.net/kmeans.html`.

[10] Herrero, S. Multiclass classification using massively threaded multiprocessors. *Last visisted 04/12/2015,* `http://www.slideshare.net/sergherrero/multiclass-classification-using-massively-threaded-multiprocessors`.

[11] The problem of overfitting. *Last visited 17/11/2015* `https://class.coursera.org/ml-005/lecture/39`.

[12] Zhang, Y. & Wu, L. 2012. Classification of fruits using computer vision and a multiclass support vector machine. *Sensors*, 12(9), 12489–12505.

[13] Kononenko, I. & Kukar, M. 2007. *Machine Learning and Data Mining: Introduction to Principles and Algorithms*. Horwood Publishing Limited.

[14] Li, L. & Lin, H.-T. 2006. Ordinal regression by extended binary classification. In *Advances in neural information processing systems*, 865–872.

[15] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, E. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12, 2825–2830.

[16] Kolter, J. Z. & Maloof, M. A. December 2006. Learning to detect and classify malicious executables in the wild. *J. Mach. Learn. Res.*, 7, 2721–2744.

[17] Mohaisen, A. & Alrawi, O. 2013. Unveiling zeus: automated classification of malware samples. In *Proceedings of the 22nd international conference on World Wide Web companion*, 829–832. International World Wide Web Conferences Steering Committee.

[18] Rieck, K., Holz, T., Willems, C., Düssel, P., & Laskov, P. 2008. Learning and classification of malware behavior. In *Detection of Intrusions and Malware, and Vulnerability Assessment*, 108–125. Springer.

[19] McAfee. Part of Intel Security. August 2015. `http://www.mcafee.com/mx/resources/reports/rp-quarterly-threats-aug-2015.pdf`.

[20] Shabtai, A., Moskovitch, R., Elovici, Y., & Glezer, C. 2009. Detection of malicious code by applying machine learning classifiers on static features: A state-of-the-art survey. *Information Security Technical Report*, 14(1), 16–29.

[21] Ye, Y., Wang, D., Li, T., & Ye, D. 2007. Imds: Intelligent malware detection system. In *Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '07, 1043–1047, New York, NY, USA. ACM.

[22] Seltzer, L. 2014. Tools for analyzing static properties of suspicious files on windows. *SANS Digital Forensics and Incident Response Blog Last visited 18/11/2015* `https://digital-forensics.sans.org/blog/2014/03/04/tools-for-analyzing-static-properties-of-suspicious-files-on-windows`.

[23] Ligh, M., Adair, S., Hartstein, B., & Richard, M. 2010. *Malware Analyst's Cookbook and DVD: Tools and Techniques for Fighting Malicious Code*. Wiley Publishing.

[24] Gavrilut, D., Cimpoesu, M., Anton, D., & Ciortuz, L. Oct 2009. Malware detection using machine learning. In *Computer Science and Information Technology, 2009. IMCSIT '09. International Multiconference on*, 735–741.

[25] Idika, N. & Mathur, A. P. 2007. A survey of malware detection techniques. *Purdue University*, 48.

[26] Bishop, M. A. 2002. *The Art and Science of Computer Security*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA.

[27] Stallings, W. & Brown, L. 2008. Computer security. *Principles and Practice*.

[28] Banday, M. T., Qadri, J. A., & Shah, N. A. 2009. Study of botnets and their threats to internet security. *Working Papers on Information Security*.

[29] Elisan, C. 2012. *Malware, Rootkits & Botnets A Beginner's Guide*. McGraw Hill Professional.

[30] McLaughlin, L. 2004. Bot software spreads, causes new worries. *Distributed Systems Online, IEEE*, 5(6), 1.

[31] Bazrafshan, Z., Hashemi, H., Fard, S. M. H., & Hamzeh, A. 2013. A survey on heuristic malware detection techniques. In *Information and Knowledge Technology (IKT), 2013 5th Conference on*, 113–120. IEEE.

[32] Vinod, P., Jaipur, R., Laxmi, V., & Gaur, M. 2009. Survey on malware detection methods. In *Proceedings of the 3rd Hackers' Workshop on Computer and Internet Security (IITKHACK'09)*, 74–79.

[33] Ståhlberg, M. August 11 2009. Malware detection. US Patent App. 12/462,913.

[34] Balakrishnan, A. & Schulze, C. 2005. Code obfuscation literature survey. *CS701 Construction of compilers*, 19.

[35] Sand, L. A. 2012. Information-based dependency matching for behavioral malware analysis.

[36] Schiffman, M. 2010. A brief history of malware obfuscation: Part 1 of 2.

[37] Micrososft Developer Network. Opcodes::nop field. *Last visisted 16/11/2015,* `https://msdn.microsoft.com/en-us/library/system.reflection.emit.opcodes.nop(v=vs.110).aspx?cs-save-lang=1&cs-lang=cpp#code-snippet-1`.

[38] Kath, R. 1993. The portable executable file format from top to bottom. *MSDN Library, Microsoft Corporation*.

[39] Bragen, S. R. 2015. Malware detection through opcode sequence analysis using machine learning. *Gjøvik University College*.

[40] Micrososft Developer Network. Peering inside the pe: A tour of the win32 portable executable file format. *Last visisted 16/11/2015,* `https://msdn.microsoft.com/en-us/library/ms809762.aspx`.

[41] FitzGerald, N. 2002. A virus by any other name: Towards the revised caro naming convention. *Proc. AVAR*, 141–166.

[42] Bishop, C. M. 2006. *Pattern recognition and machine learning*. springer.

[43] Ou, G. & Murphey, Y. L. 2007. Multi-class pattern classification using neural networks. *Pattern Recognition*, 40(1), 4–18.

[44] Milgram, J., Cheriet, M., & Sabourin, R. 2006. "one against one" or "one against all": Which one is better for handwriting recognition with svms? In *Tenth International Workshop on Frontiers in Handwriting Recognition*. Suvisoft.

[45] Duda, R. O., Hart, P. E., & Stork, D. G. 2012. *Pattern classification*. John Wiley & Sons.

[46] Cohen, F. 1987. Computer viruses: theory and experiments. *Computers & security*, 6(1), 22–35.

[47] Chess, D. M. & White, S. R. 2000. An undetectable computer virus. In *Proceedings of Virus Bulletin Conference*, volume 5.

[48] Leder, F., Steinbock, B., & Martini, P. 2009. Classification and detection of metamorphic malware using value set analysis. In *Malicious and Unwanted Software (MALWARE), 2009 4th International Conference on*, 39–46. IEEE.

[49] Aly, M. 2005. Survey on multiclass classification methods. *Neural Netw*, 1–9.

[50] Mehra, N. & Gupta, S. 2013. Survey on multiclass classification methods.

[51] Liao, Y. 2012. Pe-header-based malware study and detection. *Retrieved from the University of Georgia: http://www. cs. uga. edu/˜ liao/PE_Final_Report. pdf*.

[52] Zhang, B., Yin, J., Hao, J., Zhang, D., & Wang, S. 2007. Malicious codes detection based on ensemble learning. In *Autonomic and trusted computing*, 468–477. Springer.

[53] Moskovitch, R., Feher, C., Tzachar, N., Berger, E., Gitelman, M., Dolev, S., & Elovici, Y. 2008. Unknown malcode detection using opcode representation. In *Intelligence and Security Informatics*, 204–215. Springer.

[54] Leedy, P. D. & Ormrod, J. E. 2005. Practical research. *Planning and design*, 8.

[55] Zeltser, L. 2014. Tools for analyzing static properties of suspicious files on windows.

[56] 3 Data Mining Software in Java, W. *Last visited 01/12/2015,* `http://www.cs.waikato.ac.nz/~ml/weka/`.

[57] Hall, M. A. *Correlation-based feature selection for machine learning*. PhD thesis, The University of Waikato, 1999.

[58] Roobaert, D., Karakoulas, G., & Chawla, N. V. 2006. Information gain, correlation and support vector machines. In *Feature Extraction*, 463–470. Springer.

[59] Frank, E. & Kirkby, R. Weka class randomtree. *Last visited: 13/12/15,* `http://weka.sourceforge.net/doc.dev/weka/classifiers/trees/RandomTree.html`.

[60] Quinlan, R. 1993. *C4.5: Programs for Machine Learning*. Morgan Kaufmann Publishers, San Mateo, CA.

[61] Aha, D. & Kibler, D. 1991. Instance-based learning algorithms. *Machine Learning*, 6, 37–66.

[62] Rish, I. 2001. An empirical study of the naive bayes classifier. In *IJCAI 2001 workshop on empirical methods in artificial intelligence*, volume 3, 41–46. IBM New York.

[63] Friedman, N., Geiger, D., & Goldszmidt, M. 1997. Bayesian network classifiers. *Machine learning*, 29(2-3), 131–163.

[64] Hearst, M. A., Dumais, S. T., Osman, E., Platt, J., & Scholkopf, B. 1998. Support vector machines. *Intelligent Systems and their Applications, IEEE*, 13(4), 18–28.

[65] Python Software Foundation. *Last visited 01/12/15,* `https://www.python.org`.

[66] Python Software Foundation. *Last visited 01/12/15,* `https://pypi.python.org/pypi/pip`.

[67] Python Software Foundation. *Last visited 14/12/15,* `https://pypi.python.org/pypi/PyMySQL`.

[68] Oracle Corporation. Mysql homepage. `https://www.mysql.com`.

[69] The PHP Group. Php homepage. `http://www.php.net`.

[70] MySQL Dump Importer, B. *Last visited 01/12/15,* `http://www.ozerov.de/bigdump/`.

[71] phpMyAdmin: Bringing MySQL to the web. *Last visisted 30/11/2015,* `https://www.phpmyadmin.net/`.

[72] Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., & Witten, I. H. 2009. The weka data mining software: an update. *ACM SIGKDD explorations newsletter*, 11(1), 10–18.

[73] Mahoney, M. 1999. Refining the estimated entropy of english by shannon game simulation.

[74] Weka Documentation. Weka class resample. *Last visited: 14/12/15,* `http://weka.sourceforge.net/doc.dev/weka/filters/supervised/instance/Resample.html`.

A RawData database

A.1 Database explanation

Output from `mysql > SHOW FULL COLUMNS FROM rawData;`

```
+---------------------------------+------------------+-----------------+
| Field                           | Type             | Collation       |
+---------------------------------+------------------+-----------------+
| md5                             | binary(16)       | NULL            |
| virustotal_file_report          | mediumtext       | utf8_general_ci |
| virustotal_file_behaviour       | mediumtext       | utf8_general_ci |
| virustotal_file_network_traffic | mediumtext       | utf8_general_ci |
| peframe                         | mediumtext       | utf8_general_ci |
| file                            | text             | utf8_general_ci |
| strings                         | mediumtext       | utf8_general_ci |
| size                            | text             | utf8_general_ci |
| file_entropy                    | double unsigned  | NULL            |
| size_of_file                    | int(10) unsigned | NULL            |
| do_not_process                  | tinyint(1)       | NULL            |
+---------------------------------+------------------+-----------------+
```

Figure 29: rawData table in PhpMyAdmin

A.2 Processed database

```
mysql > SHOW FULL COLUMNS FROM data;
```

```
+------------------------+------------+
| Field                  | Type       |
+------------------------+------------+
| md5                    | binary(16) |
| vt_codesize            | bigint(20) |
| vt_res_langs           | bigint(20) |
| vt_res_types           | bigint(20) |
| vt_sections            | bigint(20) |
| vt_entry_point         | bigint(20) |
| vt_initDataSize        | bigint(20) |
| vt_productName         | bigint(20) |
| vt_originalFileName    | bigint(20) |
| vt_unitializedDataSize | bigint(20) |
| vt_legalCopyright      | bigint(20) |
| pe_api                 | bigint(20) |
| pe_debug               | bigint(20) |
| pe_packer              | bigint(20) |
| pe_library             | bigint(20) |
| pe_autogen             | bigint(20) |
| pe_object              | bigint(20) |
| pe_executable          | bigint(20) |
| pe_text                | bigint(20) |
| pe_binary              | bigint(20) |
| pe_temporary           | bigint(20) |
| pe_database            | bigint(20) |
| pe_log                 | bigint(20) |
| pe_webpage             | bigint(20) |
| pe_backup              | bigint(20) |
| pe_cabinet             | bigint(20) |
| pe_data                | bigint(20) |
| pe_registry            | bigint(20) |
| pe_directories         | bigint(20) |
| pe_dll                 | bigint(20) |
| pe_detected            | bigint(20) |
| size_TEXT              | bigint(20) |
| size_DATA              | bigint(20) |
| size_OBJ               | bigint(20) |
| size_TOT               | bigint(20) |
| filesize               | bigint(20) |
| entropy                | double     |
| type                   | text       |
| family                 | text       |
+------------------------+------------+
```

Please note that this is the database containing the constructed features for every sample, including both family and type of malware. This was not the dataset used in our experiments, as discussed through the thesis. This database from which we generated the data subsets used in the experiments.

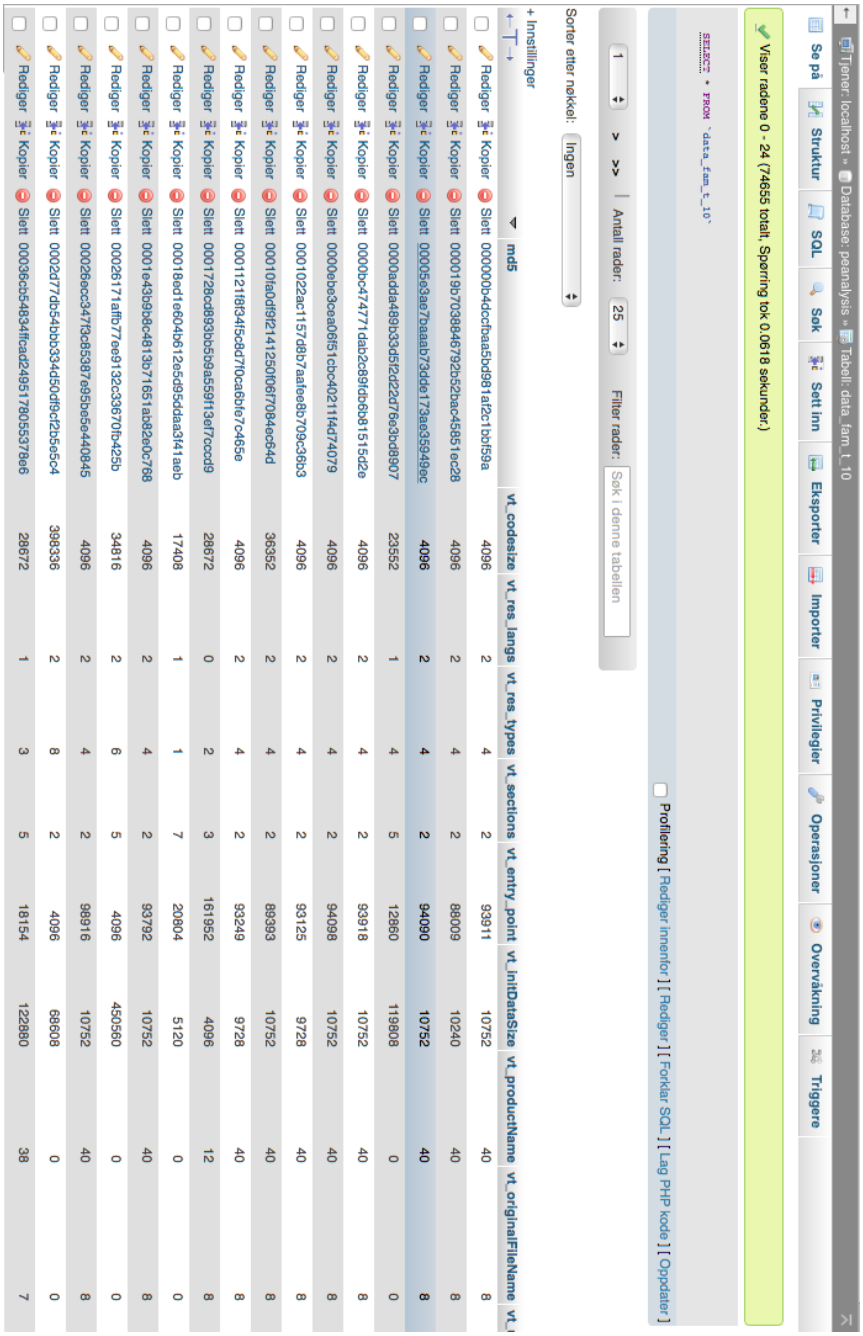


Figure 30: rawData table in PhpMyAdmin

B Sample output froom tools

B.1 PEframe

```
Short information
-------------------------------------------------------------
File Name          malware.exe
File Size          935281 byte
Compile Time       2012-01-29 22:32:28
DLL                False
Sections           4
Hash MD5           cae18bdb8e9ef082816615e033d2d85b
Hash SAH1          546060ad10a766e0ecce1feb613766a340e875c0
Imphash            353cf96592db561b5ab4e408464ac6ae
Detected           Xor, Sign, Packer, Anti Debug, Anti VM
Directory          Import, Resource, Debug, Relocation, Security

XOR discovered
-------------------------------------------------------------
Key length         Offset (hex)       Offset (dec)
1                  0x5df4e            384846
2                  0x5df4e            384846
4                  0x5df4e            384846
8                  0x5df4e            384846

Digital Signature
-------------------------------------------------------------
Virtual Address    12A200
Block Size         4813 byte
Hash MD5           63b8c4daec26c6c074ca5977f067c21e
Hash SHA-1         53731a283d0c251f7c06f6d7d423124689873c62

Packer matched [4]
-------------------------------------------------------------
Packer             Microsoft Visual C++ v6.0
Packer             Microsoft Visual C++ 5.0
Packer             Microsoft Visual C++
Packer             Installer VISE Custom

Anti Debug discovered [9]
-------------------------------------------------------------
Anti Debug         FindWindowExW
Anti Debug         FindWindowW
Anti Debug         GetWindowThreadProcessId
Anti Debug         IsDebuggerPresent
Anti Debug         OutputDebugStringW
Anti Debug         Process32FirstW
Anti Debug         Process32NextW
Anti Debug         TerminateProcess
Anti Debug         UnhandledExceptionFilter

Anti VM Trick discovered [2]
-------------------------------------------------------------
Trick              Virtual Box
Trick              VMware trick

Suspicious API discovered [35]
-------------------------------------------------------------
Function           CreateDirectoryA
Function           CreateFileA
Function           CreateFileMappingA
Function           CreateToolhelp32Snapshot
```

```
Function            DeleteFileA
Function            FindFirstFileA
Function            FindNextFileA
Function            GetCurrentProcess
Function            GetFileAttributesA
Function            GetFileSize
Function            GetModuleHandleA
Function            GetProcAddress
Function            GetTempPathA
Function            GetTickCount
Function            GetUserNameA
Function            GetVersionExA
Function            InternetCrackUrlA
Function            LoadLibraryA
Function            MapViewOfFile
Function            OpenProcess
Function            Process32First
Function            Process32Next
Function            RegCloseKey
Function            RegCreateKeyA
Function            RegEnumKeyExA
Function            RegOpenKeyA
Function            RegOpenKeyExA
Function            Sleep
Function            WSAStartup
Function            WriteFile
Function            closesocket
Function            connect
Function            recv
Function            send
Function            socket

Suspicious Sections discovered [2]
------------------------------------------------------------
Section             .data
Hash MD5            b896a2c4b2be73b89e96823c1ed68f9c
Hash SHA-1          523d58892f0375c77e5e1b6f462005ae06cdd0d8
Section             .rdata
Hash MD5            41795b402636cb13e2dbbbec031dbb1a
Hash SHA-1          b674141b34f843d54865a399edfca44c3757df59

File name discovered [43]
------------------------------------------------------------
Binary              wiseftpsrvs.bin
Data                ESTdb2.dat
Data                Favorites.dat
Data                History.dat
Data                bookmark.dat
Data                fireFTPsites.dat
Data                quick.dat
Data                site.dat
Data                sites.dat
Database            FTPList.db
Database            sites.db
Database            NovaFTP.db
Executable          unleap.exe
Executable          explorer.exe
FTP Config          FTPVoyager.ftp
Library             crypt32.dll
Library             kernel32.dll
Library             mozsqlite3.dll
Library             userenv.dll
Library             wand.dat
Library             wininet.dll
Library             wsock32.dll
Text                Connections.txt
Text                ftplist.txt
```

```
Text                signons.txt
Text                signons2.txt
Text                signons3.txt

Url discovered [2]
------------------------------------------------------------
Url                 RhinoSoft.com
Url                 http://0uk.net/zaaqw/gate.php

Meta data found [4]
------------------------------------------------------------
CompiledScript      AutoIt v3 Script
FileVersion         3, 3, 8, 1
FileDescription
Translation         0x0809 0x04b0
```

B.2 VirusTotal

```
stdClass Object
(
    [vhash] => 12402f0f7bz2?z7
    [submission_names] => Array
        (
            [0] => 000000b4dccfbaa5bd981af2c1bbf59a27333588c6f2b7c076e9a279bb40fa6d6f9afec121670.dll
            [1] => msplay32
            [2] => 000000b4dccfbaa5bd981af2c1bbf59a.dll
            [3] => /var/newbot/vh/Trojan—GameThief.Win32.OnLineGames.ikb
            [4] => Trojan—GameThief.Win32.OnLineGames.ikb
            [5] => 000000B4DCCFBAA5BD981AF2C1BBF59A
            [6] => 000000b4dccfbaa5bd981af2c1bbf59a
        )

    [scan_date] => 2014—04—04 18:38:08
    [first_seen] => 2011—07—04 16:19:24
    [times_submitted] => 15
    [additional_info] => stdClass Object
        (
            [exports] => Array
                (
                    [0] => DllCanUnloadNow
                    [1] => DllGetClassObject
                    [2] => DllRegisterServer
                    [3] => DllUnregisterServer
                    [4] => JumpOff
                    [5] => JumpOn
                    [6] => ThreadPro
                )

            [exiftool] => stdClass Object
                (
                    [LegalTrademarks] => Microsoft
                    [FileDescription] => Windows XP MSPLAY API DLL
                    [InitializedDataSize] => 10752
                    [ImageVersion] => 0.0
                    [ProductName] => Microsoft(R) Windows(R) Operating System
                    [FileVersionNumber] => 5.1.2600.3099
                    [LanguageCode] => Chinese (Simplified)
                    [FileFlagsMask] => 0x003f
                    [CharacterSet] => Windows, Chinese (Simplified)
                    [LinkerVersion] => 0.58
                    [OriginalFilename] => msplay32
                    [MIMEType] => application/octet—stream
                    [Subsystem] => Windows GUI
                    [FileVersion] => 5.1.2600.3099
                    [TimeStamp] => 1970:01:01 02:08:16+01:00
                    [FileType] => Win32 DLL
                    [PEType] => PE32
                    [InternalName] => msplay32
                    [SubsystemVersion] => 4.0
                    [FileAccessDate] => 2014:04:04 19:39:27+01:00
                    [ProductVersion] => 5.1.2600.3099 (xpsp_sp2_gdr.070308—0222)
                    [UninitializedDataSize] => 0
                    [OSVersion] => 4.0
                    [FileCreateDate] => 2014:04:04 19:39:27+01:00
                    [FileOS] => Win32
                    [LegalCopyright] => (C) Microsoft Corporation. All rights resad.
                    [MachineType] => Intel 386 or later, and compatibles
                    [CompanyName] => Microsoft Corporation
                    [CodeSize] => 4096
                    [FileSubtype] => 0
                    [ProductVersionNumber] => 5.1.2600.3099
                    [EntryPoint] => 0x16ed7
                    [ObjectFileType] => Executable application
                )

            [trid] => DOS Executable Generic (100.0%)
            [pe—imphash] => 87bed5a7cba00c7e1f4015f1bdae2183
            [pe—resource—langs] => stdClass Object
                (
                    [NEUTRAL] => 2
                    [CHINESE SIMPLIFIED] => 3
                )

            [clam—av—pua] => ClamAV PUA (Possibly Unwanted Application) detection:
While not necessarily malicious, the scanned file presents certain
characteristics which depending on the user policies and environment may
```

```
or may not configure a threat.
For full details see: http://www.clamav.net/support/faq/pua
            [magic] => PE32 executable for MS Windows (DLL) (GUI) Intel 80386 32-bit
            [sigcheck] => stdClass Object
                (
                    [publisher] => Microsoft Corporation
                    [product] => Microsoft(R) Windows(R) Operating System
                    [description] => Windows XP MSPLAY API DLL
                    [copyright] => (C) Microsoft Corporation. All rights resad.
                    [original name] => msplay32
                    [file version] => 5.1.2600.3099
                    [internal name] => msplay32
                    [link date] => 2:08 AM 1/1/1970
                )

            [compressed_parents] => Array
                (
                    [0] => b484f4f5ed824a75c32f765c3d5f85c19cfd860ea41cce09c8cc977f7fb2bf61
                )

            [imports] => stdClass Object
                (
                    [KERNEL32.DLL] => Array
                        (
                            [0] => LoadLibraryA
                            [1] => GetProcAddress
                        )

                )

            [f-prot-unpacker] => UPack, PE_Patch.MaskPE
            [peid] => WinUpack v0.39 final (relocated image base) -> By Dwing (c)2005 (h2)
            [pe-resource-types] => stdClass Object
                (
                    [RT_ICON] => 1
                    [RT_GROUP_ICON] => 1
                    [RT_VERSION] => 1
                    [RT_RCDATA] => 2
                )

            [pe-timestamp] => 4096
            [pe-resource-list] => stdClass Object
                (
                    [e3b0c44298fc1c149afbf4c8996fb92427ae41e4649b934ca495991b7852b855] => English text
                    [513bd37a0ce952e5142954c6373b2d58e7f078bd17d2f52125daaefcb53bee30] => data
                )

            [pe-entry-point] => 93911
            [sections] => Array
                (
                    [0] => Array
                        (
                            [0] => .Upack
                            [1] => 4096
                            [2] => 69632
                            [3] => 0
                            [4] => 0.00
                            [5] => d41d8cd98f00b204e9800998ecf8427e
                        )

                    [1] => Array
                        (
                            [0] => .rsrc
                            [1] => 73728
                            [2] => 53248
                            [3] => 21106
                            [4] => 7.88
                            [5] => 76baef1d5e7c419db489e7df3bedf462
                        )

                )

            [pe-machine-type] => 332
            [command-unpacker] => UPack, PE_Patch.MaskPE
        )

    [size] => 21670
    [scan_id] => 3ab5d534d493e65f01ee5dd12d650091cbf5a5a83356cbf3f8abf7ad5350ed72-1396636688
    [total] => 51
    [harmless_votes] => 0
    [verbose_msg] => Scan finished, information embedded
    [sha256] => 3ab5d534d493e65f01ee5dd12d650091cbf5a5a83356cbf3f8abf7ad5350ed72
    [type] => Win32 DLL
    [scans] => stdClass Object
        (
            [Bkav] => stdClass Object
                (
                    [detected] => 1
                    [version] => 1.3.0.4959
                    [result] => W32.WowStealBDll.Trojan
                    [update] => 20140404
                )

            [MicroWorld-eScan] => stdClass Object
                (
                    [detected] => 1
                    [version] => 12.0.250.0
                    [result] => Trojan.PWS.OnlineGames.WPK
                    [update] => 20140404
                )

            [nProtect] => stdClass Object
                (
                    [detected] => 1
                    [version] => 2014-04-04.01
                    [result] => Trojan.PWS.OnlineGames.WPK
```

```
        [update] => 20140404
    )

[CMC] => stdClass Object
    (
        [detected] => 1
        [version] => 1.1.0.977
        [result] => Generic.Win32.000000b4dc!MD
        [update] => 20140404
    )

[CAT-QuickHeal] => stdClass Object
    (
        [detected] => 1
        [version] => 12.00
        [result] => TrojanPWS.Ceekat.A2
        [update] => 20140404
    )

[McAfee] => stdClass Object
    (
        [detected] => 1
        [version] => 6.0.4.564
        [result] => PWS-OnlineGames.p
        [update] => 20140404
    )

[Malwarebytes] => stdClass Object
    (
        [detected] => 1
        [version] => 1.75.0001
        [result] => Trojan.FakeMS.ED
        [update] => 20140404
    )

[AegisLab] => stdClass Object
    (
        [detected] =>
        [version] => 1.5
        [result] =>
        [update] => 20140404
    )

[K7AntiVirus] => stdClass Object
    (
        [detected] => 1
        [version] => 9.176.11663
        [result] => Trojan ( 0005cef51 )
        [update] => 20140404
    )

[K7GW] => stdClass Object
    (
        [detected] => 1
        [version] => 9.176.11663
        [result] => Trojan ( 00361abb1 )
        [update] => 20140404
    )

[TheHacker] => stdClass Object
    (
        [detected] => 1
        [version] =>
        [result] => Trojan/PSW.OnLineGames.ikb
        [update] => 20140404
    )

[Agnitum] => stdClass Object
    (
        [detected] => 1
        [version] => 5.5.1.3
        [result] => Packed/Upack
        [update] => 20140404
    )

[F-Prot] => stdClass Object
    (
        [detected] => 1
        [version] => 4.7.1.166
        [result] => W32/Injector.D.gen!Eldorado
        [update] => 20140404
    )

[Symantec] => stdClass Object
    (
        [detected] => 1
        [version] => 20131.1.5.61
        [result] => Infostealer.Gampass
        [update] => 20140404
    )

[Norman] => stdClass Object
    (
        [detected] => 1
        [version] => 7.03.02
        [result] => Packed_Upack.H
        [update] => 20140404
    )

[TotalDefense] => stdClass Object
    (
        [detected] => 1
        [version] => 37.0.10857
        [result] => Win32/Orpdea!generic
        [update] => 20140404
    )
```

```
[TrendMicro-HouseCall] => stdClass Object
    (
        [detected] =>
        [version] => 9.700-1001
        [result] =>
        [update] => 20140404
    )

[Avast] => stdClass Object
    (
        [detected] => 1
        [version] => 8.0.1489.320
        [result] => Win32:OnLineGames-CYO [Trj]
        [update] => 20140404
    )

[ClamAV] => stdClass Object
    (
        [detected] =>
        [version] => 0.97.3
        [result] =>
        [update] => 20140404
    )

[Kaspersky] => stdClass Object
    (
        [detected] => 1
        [version] => 12.0.0.1225
        [result] => Trojan-GameThief.Win32.OnLineGames.ikb
        [update] => 20140404
    )

[BitDefender] => stdClass Object
    (
        [detected] => 1
        [version] => 7.2
        [result] => Trojan.PWS.OnlineGames.WPK
        [update] => 20140404
    )

[NANO-Antivirus] => stdClass Object
    (
        [detected] => 1
        [version] => 0.28.0.58873
        [result] => Trojan.Win32.OnLineGames.bstjn
        [update] => 20140404
    )

[SUPERAntiSpyware] => stdClass Object
    (
        [detected] => 1
        [version] => 5.6.0.1032
        [result] => Trojan.Downloader-Gen/MSPlay-Fake
        [update] => 20140404
    )

[ByteHero] => stdClass Object
    (
        [detected] =>
        [version] => 1.0.0.1
        [result] =>
        [update] => 20140404
    )

[Ad-Aware] => stdClass Object
    (
        [detected] => 1
        [version] => 12.0.163.0
        [result] => Trojan.PWS.OnlineGames.WPK
        [update] => 20140404
    )

[Sophos] => stdClass Object
    (
        [detected] => 1
        [version] => 4.98.0
        [result] => Mal/Behav-327
        [update] => 20140404
    )

[Comodo] => stdClass Object
    (
        [detected] => 1
        [version] => 18047
        [result] => TrojWare.Win32.PSW.OnLineGames.GJV
        [update] => 20140404
    )

[F-Secure] => stdClass Object
    (
        [detected] => 1
        [version] => 11.0.19100.45
        [result] => Trojan.PWS.OnlineGames.WPK
        [update] => 20140404
    )

[DrWeb] => stdClass Object
    (
        [detected] => 1
        [version] => 7.00.8.02260
        [result] => Trojan.PWS.Wow
        [update] => 20140404
    )

[VIPRE] => stdClass Object
```

```
    (
        [detected] => 1
        [version] => 28024
        [result] => Packed.Win32.Upack (v)
        [update] => 20140404
    )

[AntiVir] => stdClass Object
    (
        [detected] => 1
        [version] => 7.11.141.64
        [result] => TR/Crypt.FKM.Gen
        [update] => 20140404
    )

[TrendMicro] => stdClass Object
    (
        [detected] => 1
        [version] => 9.740-1012
        [result] => TROJ_GEN.F0C2C00AG14
        [update] => 20140404
    )

[McAfee-GW-Edition] => stdClass Object
    (
        [detected] => 1
        [version] => 2013
        [result] => Heuristic.BehavesLike.Win32.Suspicious-BAY.G
        [update] => 20140404
    )

[Emsisoft] => stdClass Object
    (
        [detected] => 1
        [version] => 3.0.0.596
        [result] => Trojan.PWS.OnlineGames.WPK (B)
        [update] => 20140404
    )

[Jiangmin] => stdClass Object
    (
        [detected] => 1
        [version] => 16.0.100
        [result] => Trojan/PSW.OnLineGames.lpj
        [update] => 20140404
    )

[Antiy-AVL] => stdClass Object
    (
        [detected] => 1
        [version] => 0.1.0.1
        [result] => Trojan/Win32.WOW.gic[GameThief]
        [update] => 20140404
    )

[Kingsoft] => stdClass Object
    (
        [detected] => 1
        [version] => 2013.04.09.267
        [result] => Win32.PSWTroj.WowT.my.17831
        [update] => 20140404
    )

[Microsoft] => stdClass Object
    (
        [detected] => 1
        [version] => 1.10401
        [result] => PWS:Win32/OnLineGames.CPL
        [update] => 20140404
    )

[ViRobot] => stdClass Object
    (
        [detected] => 1
        [version] => 2011.4.7.4223
        [result] => Packed.Win32.UPack
        [update] => 20140404
    )

[GData] => stdClass Object
    (
        [detected] => 1
        [version] => 24
        [result] => Trojan.PWS.OnlineGames.WPK
        [update] => 20140404
    )

[Commtouch] => stdClass Object
    (
        [detected] => 1
        [version] => 5.4.1.7
        [result] => W32/Injector.D.gen!Eldorado
        [update] => 20140404
    )

[AhnLab-V3] => stdClass Object
    (
        [detected] => 1
        [version] => None
        [result] => Trojan/Win32.OnlineGameHack
        [update] => 20140404
    )

[VBA32] => stdClass Object
    (
        [detected] => 1
```

```
                [version] => 3.12.26.0
                [result] => suspected of MalwareScope.Trojan—PSW.Game.7
                [update] => 20140404
            )

        [Panda] => stdClass Object
            (
                [detected] => 1
                [version] => 10.0.3.5
                [result] => Trj/Legmir.AUZ
                [update] => 20140404
            )

        [ESET—NOD32] => stdClass Object
            (
                [detected] => 1
                [version] => 9637
                [result] => Win32/PSW.OnLineGames.GJV
                [update] => 20140404
            )

        [Rising] => stdClass Object
            (
                [detected] => 1
                [version] => 25.0.0.11
                [result] => PE:Trojan.PSW.Win32.GameOL.muv!1075127113
                [update] => 20140404
            )

        [Ikarus] => stdClass Object
            (
                [detected] => 1
                [version] => T3.1.5.6.0
                [result] => Trojan—GameThief.Win32.Nilage
                [update] => 20140404
            )

        [Fortinet] => stdClass Object
            (
                [detected] =>
                [version] => 4
                [result] =>
                [update] => 20140404
            )

        [AVG] => stdClass Object
            (
                [detected] => 1
                [version] => 13.0.0.3169
                [result] => Win32/PEMask
                [update] => 20140404
            )

        [Baidu—International] => stdClass Object
            (
                [detected] => 1
                [version] => 3.5.1.41473
                [result] => Trojan.Win32.OnLineGames.AVj
                [update] => 20140404
            )

        [Qihoo—360] => stdClass Object
            (
                [detected] => 1
                [version] => 1.0.0.1015
                [result] => Trojan.PSW.Win32.WOW.D
                [update] => 20140404
            )

    )

[tags] => Array
    (
        [0] => upack
        [1] => pedll
    )

[unique_sources] => 9
[positives] => 46
[ssdeep] => 384:bWWTEcWWcL9bXUU4Y1tzcz7O/vTHIreN7zNyM3tsbFxJ26IbMpBdGUEf3GR+MAd/:UVhtoHO/vVwZIbvf3BFwvzA
[md5] => 000000b4dccfbaa5bd981af2c1bbf59a
[permalink] => https://www.virustotal.com/file/3ab5d534d493e65f01ee5dd12d650091cbf5a5a83356cbf3f8abf7ad5350ed72/analysis
  /1396636688/
[sha1] => 27333588c6f2b7c076e9a279bb40fa6d6f9afec1
[resource] => 000000B4DCCFBAA5BD981AF2C1BBF59A
[response_code] => 1
[community_reputation] => 0
[malicious_votes] => 0
[ITW_urls] => Array
    (
    )

[last_seen] => 2014—04—04 18:38:08
)
```

B.3 Serialized VirusTotal

```
O:8:"stdClass":27:{s:5:"vhash";s:15:"12402f0f7bz2?z7";s:16:"submission_names";a:7:{i:0;s:81:"000000
    b4dccfbaa5bd981af2c1bbf59a27333588c6f2b7c076e9a279bb40fa6d6f9afec121670.dll";i:1;s:8:"msplay32";i:2;s:36:"000000
    b4dccfbaa5bd981af2c1bbf59a.dll";i:3;s:53:"/var/newbot/vh/Trojan—GameThief.Win32.OnLineGames.ikb";i:4;s:38:"Trojan—
    GameThief.Win32.OnLineGames.ikb";i:5;s:32:"000000B4DCCFBAA5BD981AF2C1BBF59A";i:6;s:32:"000000b4dccfbaa5bd981af2c1bbf59a
    ";}s:9:"scan_date";s:19:"2014—04—04 18:38:08";s:10:"first_seen";s:19:"2011—07—04 16:19:24";s:15:"times_submitted";i:15;s
    :15:"additional_info";O:8:"stdClass":19:{s:7:"exports";a:7:{i:0;s:15:"DllCanUnloadNow";i:1;s:17:"DllGetClassObject";i:2;s
    :17:"DllRegisterServer";i:3;s:19:"DllUnregisterServer";i:4;s:7:"JumpOff";i:5;s:6:"JumpOn";i:6;s:9:"ThreadPro";}s:8:"
    exiftool";O:8:"stdClass":33:{s:15:"LegalTrademarks";s:9:"Microsoft";s:15:"FileDescription";s:25:"Windows XP MSPLAY API
```

DLL";s:19:"InitializedDataSize";s:5:"10752";s:12:"ImageVersion";s:3:"0.0";s:11:"ProductName";s:40:"Microsoft(R) Windows(R) Operating System";s:17:"FileVersionNumber";s:13:"5.1.2600.3099";s:12:"LanguageCode";s:20:"Chinese (Simplified)";s:13:"FileFlagsMask";s:6:"0x003f";s:12:"CharacterSet";s:29:"Windows, Chinese (Simplified)";s:13:"LinkerVersion";s:4:"0.58";s:16:"OriginalFilename";s:8:"msplay32";s:8:"MIMEType";s:24:"application/octet-stream";s:9:"Subsystem";s:11:"Windows GUI";s:11:"FileVersion";s:13:"5.1.2600.3099";s:9:"TimeStamp";s:25:"1970:01:01 02:08:16+01:00";s:8:"FileType";s:9:"Win32 DLL";s:6:"PEType";s:4:"PE32";s:12:"InternalName";s:8:"msplay32";s:16:"SubsystemVersion";s:3:"4.0";s:14:"FileAccessDate";s:25:"2014:04:04 19:39:27+01:00";s:14:"ProductVersion";s:40:"5.1.2600.3099 (xpsp_sp2_gdr.070308-0222)";s:21:"UninitializedDataSize";s:1:"0";s:9:"OSVersion";s:3:"4.0";s:14:"FileCreateDate";s:25:"2014:04:04 19:39:27+01:00";s:6:"FileOS";s:5:"Win32";s:14:"LegalCopyright";s:44:"(C) Microsoft Corporation. All rights resad.";s:11:"MachineType";s:35:"Intel 386 or later, and compatibles";s:11:"CompanyName";s:21:"Microsoft Corporation";s:8:"CodeSize";s:4:"4096";s:11:"FileSubtype";s:1:"0";s:20:"ProductVersionNumber";s:13:"5.1.2600.3099";s:10:"EntryPoint";s:7:"0x16ed7";s:14:"ObjectFileType";s:22:"Executable application";}s:4:"trid";s:31:"DOS Executable Generic (100.0%)";s:10:"pe-imphash";s:32:"87bed5a7cba00c7e1f4015f1bdae2183";s:17:"pe-resource-langs";O:8:"stdClass":2:{s:7:"NEUTRAL";i:2;s:18:"CHINESE SIMPLIFIED";i:3;}s:11:"clam-av-pua";s:284:"ClamAV PUA (Possibly Unwanted Application) detection:
While not necessarily malicious, the scanned file presents certain
characteristics which depending on the user policies and environment may
or may not configure a threat.
For full details see: http://www.clamav.net/support/faq/pua";s:5:"magic";s:61:"PE32 executable for MS Windows (DLL) (GUI) Intel 80386 32-bit";s:8:"sigcheck";O:8:"stdClass":8:{s:9:"publisher";s:21:"Microsoft Corporation";s:7:"product";s:40:"Microsoft(R) Windows(R) Operating System";s:11:"description";s:25:"Windows XP MSPLAY API DLL";s:9:"copyright";s:44:"(C) Microsoft Corporation. All rights resad.";s:13:"original name";s:8:"msplay32";s:12:"file version";s:13:"5.1.2600.3099";s:13:"internal name";s:8:"msplay32";s:9:"link date";s:16:"2:08 AM 1/1/1970";}s:18:"compressed-parents";a:1:{i:0;s:64:"b484f4f5ed824a75c32f765c3d5f85c19cfd860ea41cce09c8cc977f7fb2bf61";}s:7:"imports";O:8:"stdClass":1:{s:12:"KERNEL32.DLL";a:2:{i:0;s:12:"LoadLibraryA";i:1;s:14:"GetProcAddress";}}s:15:"f-prot-unpacker";s:22:"UPack, PE_Patch.MaskPE";s:4:"peid";s:68:"WinUpack v0.39 final (relocated image base) -> By Dwing (c)2005 (h2)";s:17:"pe-resource-types";O:8:"stdClass":4:{s:7:"RT_ICON";i:1;s:13:"RT_GROUP_ICON";i:1;s:10:"RT_VERSION";i:1;s:9:"RT_RCDATA";i:2;}s:12:"pe-timestamp";i:4096;s:16:"pe-resource-list";O:8:"stdClass":2:{s:64:"e3b0c44298fc1c149afbf4c8996fb92427ae41e4649b934ca495991b7852b855";s:12:"English text";s:64:"513bd37a0ce952e5142954c6373b2d58e7f078bd17d2f52125daaefcb53bee30";s:4:"data";}s:14:"pe-entry-point";i:93911;s:8:"sections";a:2:{i:0;a:6:{i:0;s:6:".Upack";i:1;i:4096;i:2;i:69632;i:3;i:0;i:4;s:4:"0.00";i:5;s:32:"d41d8cd98f00b204e9800998ecf8427e";}i:1;a:6:{i:0;s:5:".rsrc";i:1;i:73728;i:2;i:53248;i:3;i:21106;i:4;s:4:"7.88";i:5;s:32:"76baef1d5e7c419db489e7df3bedf462";}}s:15:"pe-machine-type";i:332;s:16:"command-unpacker";s:22:"UPack, PE_Patch.MaskPE";}s:4:"size";i:21670;s:7:"scan_id";s:75:"3ab5d534d493e65f01ee5dd12d650091cbf5a5a83356cbf3f8abf7ad5350ed72-1396636688";s:5:"total";i:51;s:14:"harmless_votes";i:0;s:11:"verbose_msg";s:35:"Scan finished, information embedded";s:6:"sha256";s:64:"3ab5d534d493e65f01ee5dd12d650091cbf5a5a83356cbf3f8abf7ad5350ed72";s:4:"type";s:9:"Win32 DLL";s:5:"scans";O:8:"stdClass":51:{s:4:"Bkav";O:8:"stdClass":4:{s:8:"detected";b:1;s:7:"version";s:10:"1.3.0.4959";s:6:"result";s:23:"W32.WowStealBDll.Trojan";s:6:"update";s:8:"20140404";}s:16:"MicroWorld-eScan";O:8:"stdClass":4:{s:8:"detected";b:1;s:7:"version";s:10:"12.0.250.0";s:6:"result";s:26:"Trojan.PWS.OnlineGames.WPK";s:6:"update";s:8:"20140404";}s:8:"nProtect";O:8:"stdClass":4:{s:8:"detected";b:1;s:7:"version";s:13:"2014-04-04.01";s:6:"result";s:26:"Trojan.PWS.OnlineGames.WPK";s:6:"update";s:8:"20140404";}s:3:"CMC";O:8:"stdClass":4:{s:8:"detected";b:1;s:7:"version";s:9:"1.1.0.977";s:6:"result";s:27:"Generic.Win32.000000b4dc!MD";s:6:"update";s:8:"20140404";}s:13:"CAT-QuickHeal";O:8:"stdClass":4:{s:8:"detected";b:1;s:7:"version";s:5:"12.00";s:6:"result";s:19:"TrojanPWS.Ceekat.A2";s:6:"update";s:8:"20140404";}s:6:"McAfee";O:8:"stdClass":4:{s:8:"detected";b:1;s:7:"version";s:9:"6.0.4.564";s:6:"result";s:17:"PWS-OnlineGames.p";s:6:"update";s:8:"20140404";}s:12:"Malwarebytes";O:8:"stdClass":4:{s:8:"detected";b:1;s:7:"version";s:9:"1.75.0001";s:6:"result";s:16:"Trojan.FakeMS.ED";s:6:"update";s:8:"20140404";}s:8:"AegisLab";O:8:"stdClass":4:{s:8:"detected";b:0;s:7:"version";s:3:"1.5";s:6:"result";N;s:6:"update";s:8:"20140404";}s:11:"K7AntiVirus";O:8:"stdClass":4:{s:8:"detected";b:1;s:7:"version";s:11:"9.176.11663";s:6:"result";s:20:"Trojan (0005cef51)";s:6:"update";s:8:"20140404";}s:4:"K7GW";O:8:"stdClass":4:{s:8:"detected";b:1;s:7:"version";s:11:"9.176.11663";s:6:"result";s:20:"Trojan (00361abb1)";s:6:"update";s:8:"20140404";}s:9:"TheHacker";O:8:"stdClass":4:{s:8:"detected";b:1;s:7:"version";N;s:6:"result";s:26:"Trojan/PSW.OnLineGames.ikb";s:6:"update";s:8:"20140404";}s:7:"Agnitum";O:8:"stdClass":4:{s:8:"detected";b:1;s:7:"version";s:7:"5.5.1.3";s:6:"result";s:12:"Packed/Upack";s:6:"update";s:8:"20140404";}s:6:"F-Prot";O:8:"stdClass":4:{s:8:"detected";b:1;s:7:"version";s:9:"4.7.1.166";s:6:"result";s:27:"W32/Injector.D.gen!Eldorado";s:6:"update";s:8:"20140404";}s:8:"Symantec";O:8:"stdClass":4:{s:8:"detected";b:1;s:7:"version";s:12:"20131.1.5.61";s:6:"result";s:19:"Infostealer.Gampass";s:6:"update";s:8:"20140404";}s:6:"Norman";O:8:"stdClass":4:{s:8:"detected";b:1;s:7:"version";s:7:"7.03.02";s:6:"result";s:14:"Packed_Upack.H";s:6:"update";s:8:"20140404";}s:12:"TotalDefense";O:8:"stdClass":4:{s:8:"detected";b:1;s:7:"version";s:10:"37.0.10857";s:6:"result";s:20:"Win32/Orpdea!generic";s:6:"update";s:8:"20140404";}s:20:"TrendMicro-HouseCall";O:8:"stdClass":4:{s:8:"detected";b:0;s:7:"version";s:10:"9.700-1001";s:6:"result";N;s:6:"update";s:8:"20140404";}s:5:"Avast";O:8:"stdClass":4:{s:8:"detected";b:1;s:7:"version";s:12:"8.0.1489.320";s:6:"result";s:27:"Win32:OnLineGames-CYO [Trj]";s:6:"update";s:8:"20140404";}s:6:"ClamAV";O:8:"stdClass":4:{s:8:"detected";b:0;s:7:"version";s:6:"0.97.3";s:6:"result";N;s:6:"update";s:8:"20140404";}s:9:"Kaspersky";O:8:"stdClass":4:{s:8:"detected";b:1;s:7:"version";s:11:"12.0.0.1225";s:6:"result";s:38:"Trojan-GameThief.Win32.OnLineGames.ikb";s:6:"update";s:8:"20140404";}s:11:"BitDefender";O:8:"stdClass":4:{s:8:"detected";b:1;s:7:"version";s:3:"7.2";s:6:"result";s:26:"Trojan.PWS.OnlineGames.WPK";s:6:"update";s:8:"20140404";}s:14:"NANO-Antivirus";O:8:"stdClass":4:{s:8:"detected";b:1;s:7:"version";s:12:"0.28.0.58873";s:6:"result";s:30:"Trojan.Win32.OnLineGames.bstjn";s:6:"update";s:8:"20140404";}s:16:"SUPERAntiSpyware";O:8:"stdClass":4:{s:8:"detected";b:1;s:7:"version";s:10:"5.6.0.1032";s:6:"result";s:33:"Trojan.Downloader-Gen/MSPlay-Fake";s:6:"update";s:8:"20140404";}s:8:"ByteHero";O:8:"stdClass":4:{s:8:"detected";b:0;s:7:"version";s:7:"1.0.0.1";s:6:"result";N;s:6:"update";s:8:"20140404";}s:8:"Ad-Aware";O:8:"stdClass":4:{s:8:"detected";b:1;s:7:"version";s:10:"12.0.163.0";s:6:"result";s:26:"Trojan.PWS.OnlineGames.WPK";s:6:"update";s:8:"20140404";}s:6:"Sophos";O:8:"stdClass":4:{s:8:"detected";b:1;s:7:"version";s:6:"4.98.0";s:6:"result";s:13:"Mal/Behav-327";s:6:"update";s:8:"20140404";}s:6:"Comodo";O:8:"stdClass":4:{s:8:"detected";b:1;s:7:"version";s:5:"18047";s:6:"result";s:34:"TrojWare.Win32.PSW.OnLineGames.GJV";s:6:"update";s:8:"20140404";}s:8:"F-Secure";O:8:"stdClass":4:{s:8:"detected";b:1;s:7:"version";s:13:"11.0.19100.45";s:6:"result";s:26:"Trojan.PWS.OnlineGames.WPK";s:6:"update";s:8:"20140404";}s:5:"DrWeb";O:8:"stdClass":4:{s:8:"detected";b:1;s:7:"version";s:12:"7.00.8.02260";s:6:"result";s:14:"Trojan.PWS.Wow";s:6:"update";s:8:"20140404";}s:5:"VIPRE";O:8:"stdClass":4:{s:8:"detected";b:1;s:7:"version";s:5:"28024";s:6:"result";s:22:"Packed.Win32.Upack (v)";s:6:"update";s:8:"20140404";}s:7:"AntiVir";O:8:"stdClass":4:{s:8:"detected";b:1;s:7:"version";s:11:"7.11.141.64";s:6:"result";s:16:"TR/Crypt.FKM.Gen";s:6:"update";s:8:"20140404";}s:10:"TrendMicro";O:8:"stdClass":4:{s:8:"detected";b:1;s:7:"version";s:10:"9.740-1012";s:6:"result";s:20:"TROJ_GEN.F0C2C00AG14";s:6:"update";s:8:"20140404";}s:17:"McAfee-GW-Edition";O:8:"stdClass":4:{s:8:"detected";b:1;s:7:"version";s:4:"2013";s:6:"result";s:44:"Heuristic.BehavesLike.Win32.Suspicious-BAY.G";s:6:"update";s:8:"20140404";}s:8:"Emsisoft";O:8:"stdClass":4:{s:8:"detected";b:1;s:7:"version";s:9:"3.0.0.596";s:6:"result";s:30:"Trojan.PWS.OnlineGames.WPK (B)";s:6:"update";s:8:"20140404";}s:8:"Jiangmin";O:8:"stdClass":4:{s:8:"detected";b:1;s:7:"version";s:8:"16.0.100";s:6:"result";s:26:"Trojan/PSW.OnLineGames.lpj";s:6:"update";s:8:"20140404";}s:9:"Antiy-AVL";O:8:"stdClass":4:{s:8:"detected";b:1;s:7:"version";s:7:"0.1.0.1";s:6:"result";s:31:"Trojan/Win32.WOW.gic[GameThief]";s:6:"update";s:8:"20140404";}s:8:"Kingsoft";O:8:"stdClass":4:{s:8:"detected";b:1;s:7:"version";s:14:"2013.04.09.267";s:6:"result";s:27:"Win32.PSWTroj.WowT.my.17831";s:6:"update";s:8:"20140404";}s:9:"Microsoft";O:8:"stdClass":4:{s:8:"detected";b:1;s:7:"version";s:7:"1.10401";s:6:"result";s:25:"PWS:Win32/OnLineGames.CPL";s:6:"update";s:8:"20140404";}s:7:"ViRobot";O:8:"stdClass":4:{s:8:"detected";b:1;s:7:"version";s:13:"2011.4.7.4223";s:6:"result";s:18:"Packed.Win32.UPack";s:6:"update";s:8:"20140404";}s:5:"GData";O:8:"stdClass":4:{s:8:"detected";b:1;s:7:"version";s:2:"24";s:6:"result";s:26:"Trojan.PWS.OnlineGames.WPK";s:6:"update";s:8:"20140404";}s:9:"Commtouch";O:8:"stdClass":4:{s:8:"detected";b:1;s:7:"version";s:7:"5.4.1.7";s:6:"result";s:27:"W32/Injector.D.gen!Eldorado";s:6:"update";s:8:"20140404";}s:9:"AhnLab-V3";O:8:"stdClass":4:{s:8:"detected";b:1;s:7:"version";s:4:"None";s:6:"result";s:27:"Trojan/Win32.OnlineGameHack";s:6:"update";s:8:"20140404";}s:5:"VBA32";O:8:"stdClass":4:{s:8:"detected";b:1;s:7:"version";s:9:"3.12.26.0";s:6:"result";s:43:"suspected of MalwareScope.Trojan-PSW.Game.7";s:6:"update";s:8:"20140404";}s:5:"Panda";O:8:"stdClass":4:{s:8:"detected";b:1;s:7:"version";s:8:"10.0.3.5";s:6:"result";s:14:"Trj/Legmir.AUZ";s:6:"update";s:8:"20140404";}s:10:"ESET-NOD32";O:8:"stdClass":4:{s:8:"detected";b:1;s:7:"version";s:4:"9637";s:6:"result";s:25:"Win32/PSW.OnLineGames.GJV";s:6:"update";s:8:"20140404";}s:6:"Rising";O:8:"stdClass":4:{s:8:"detected";b:1;s:7:"version";s:9:"25.0.0.11";s:6:"result";s:41:"PE:Trojan.PSW.Win32.GameOL.muv!1075127113";s:6:"update";s:8:"20140404";}s:6:"Ikarus";O:8:"stdClass":4:{s:8:"detected";b:1;s:7:"version";s:10:"T3.1.5.6.0";s:6:"result";s:29:"Trojan-GameThief.Win32.Nilage";s:6:"update";s:8:"20140404";}s:8:"Fortinet";O:8:"stdClass":4:{s:8:"detected";b:0;s:7:"version";s:1:"4";s:6:"result";N;s:6:"update";s:8:"20140404";}s:3:"AVG";O:8:"stdClass":4:{s:8:"detected";b:1;s:7:"version";s:11:"13.0.0.3169";s:6:"result";s:12:"Win32/PEMask";s:6:"update";s:8:"20140404";}s:19:"Baidu-International";O:8:"stdClass":4:{s:8:"detected";b:1;s:7:"version";s:11:"3.5.1.41473";s:6:"result";s:28:"Trojan.Win32.OnLineGames.AVj";s:6:"update";s:8:"20140404";}s:9:"Qihoo-360";O:8:"stdClass":4:{s:8:"detected";b:1;s:7:"version";s:10:"1.0.0.1015";s:6:"result";s:22:"Trojan.PSW.Win32.WOW.D";s:6:"update";s:8:"20140404";}}s:4:"tags";a:2:{i:0;s:5:"upack";i:1;s:5:"pedll";}s:14:"unique_sources";i:9;s:9:"positives";i:46;s:6:"ssdeep";s:92:"384:bWWTEcWWcL9bXUU4Y1tzcz7O/vTHIreN7zNyM3tsbFxJ26IbMpBdGUEf3GR+MAd/:UVhtoHO/vVwZlbvf3BFwvzA";s:3:"md5";s:32:"000000b4dccfbaa5bd981af2c1bbf59a";s:9:"permalink";s:117:"https://www.virustotal.com/file/3ab5d534d493e65f01ee5dd12d650091cbf5a5a83356cbf3f8abf7ad5350ed72/

analysis/1396636688/";s:4:"sha1";s:40:"27333588c6f2b7c076e9a279bb40fa6d6f9afec1";s:8:"resource";s:32:"000000
B4DCCFBAA5BD981AF2C1BBF59A";s:13:"response_code";i:1;s:20:"community_reputation";i:0;s:15:"malicious_votes";i:0;s:8:"
ITW_urls";a:0:{}s:9:"last_seen";s:19:"2014—04—04 18:38:08";}

C Python Code Example for Feature Construction

C.1 rawdata_to_data.py

```
import os
import pymysql
import phpserialize as php
import json as json

vt_ms_count = 0

vt_family = ""
vt_type = ""

vt_md5 = ""
vt_tot = 0
vt_pos = 0
vt_exp = 0
vt_codesize = 0

vt_res_langs = 0
vt_res_types = 0

vt_entry_point = 0
vt_sections = 0

vt_initDataSize = 0
vt_productName = 0
vt_originalFileName = 0
vt_unitializedDataSize = 0
vt_legalCopyright = 0

pe_directories = 0
pe_dll = 0#
pe_detected = 0

pe_md5 = ""
pe_api = 0
pe_debug = 0
pe_packer = 0

#Data from peframe "File Name"—section.
pe_library = 0
pe_autogen = 0
pe_object = 0
pe_executable = 0
pe_text = 0
pe_binary = 0
pe_temporary = 0
pe_database = 0
pe_log = 0
pe_webpage = 0
pe_backup = 0
pe_cabinet = 0
pe_data = 0
pe_registry = 0
####################
size1 = 0
size2 = 0
size3 = 0
size4 = 0
size_file = 0
entropy = 0
#407730/30=13591
file = open("/Applications/XAMPP/htdocs/temp.txt", "r+")
file.truncate()
#file.write(md5_temp + "\t" + vt_tot + "\t" + vt_pos + "\t" + peframe_md5 + "\t" + pe_api + "\t" + pe_debug + "\t" + pe_packer +
    "\n")
#file.write("md5:\t\t\t\t\tTot:\tPos:\tmd5:\t\t\t\t\tAPI:\tDebug:\tPackers:\n")

md5_temp = "30adeeb55c174e8a1bcd766089c6948d" #Sets this to empty to start queries with offset = 0.

errors_in_peframe = 0
errors_in_vt = 0
con = pymysql.connect(host='localhost', port=3306, user='root', passwd='', db='peanalysis')
#407740/10730=38
for n in range(0,40773):
  cur = con.cursor()
  cur.execute("SELECT md5, virustotal_file_report, peframe, size, file_entropy, size_of_file from rawData where md5 > unhex(\""
      + md5_temp +"\") limit 10")
  row = cur.fetchall()
  for x in row:
    md5_temp = x[0].encode('hex')
    if x[2] != None and len(x[3].split()) == 4:
      peframe = php.unserialize(x[2])
      if peframe == "[Error: invalid file\n]":
        errors_in_peframe+=1
      else:
        peframe = json.loads(peframe)
        if len(peframe) == 10:
          if str(x[1]) != "i:—1;":
            vt = php.unserialize(x[1], object_hook=php.phpobject)
            vt_md5 = vt.md5
            vt_tot = vt.total
```

```
            vt_pos = vt.positives

            if 'Microsoft' in vt.scans._asdict() and vt.scans.Microsoft.result is not None:
              tekst = vt.scans.Microsoft.result.split(':')
              vt_type = tekst[0]
              vt_family = tekst[1].split('/')[1].split('.')[0]
              vt_ms_count+=1

              if 'exports' in vt.additional_info._asdict():
                vt_exp = len(vt.additional_info.exports)
              else:
                vt_exp = 0

              if 'exiftool' in vt.additional_info._asdict():
                if 'CodeSize' in vt.additional_info.exiftool._asdict():
                  vt_codesize = vt.additional_info.exiftool.CodeSize
                else:
                  vt_codesize = 0

                if 'InitializedDataSize' in vt.additional_info.exiftool._asdict():
                  vt_initDataSize = vt.additional_info.exiftool.InitializedDataSize
                else:
                  vt_initDataSize = 0

                if 'ProductName' in vt.additional_info.exiftool._asdict():
                  vt_productName = len(vt.additional_info.exiftool.ProductName)
                else:
                  vt_productName = 0

                if 'OriginalFilename' in vt.additional_info.exiftool._asdict():
                  vt_originalFileName = len(vt.additional_info.exiftool.OriginalFilename)
                else:
                  vt_originalFileName = 0



                if 'UninitializedDataSize' in vt.additional_info.exiftool._asdict():
                  vt_unitializedDataSize = vt.additional_info.exiftool.UninitializedDataSize
                else:
                  vt_unitializedDataSize = 0

                if 'LegalCopyright' in vt.additional_info.exiftool._asdict():
                  vt_legalCopyright = len(vt.additional_info.exiftool.LegalCopyright)
                else:
                  vt_legalCopyright = 0
              else:
                vt_codesize = 0

              if 'pe-resource-langs' in vt.additional_info._asdict():
                vt_res_langs = len(vt.additional_info._asdict()['pe-resource-langs']._asdict())
              else:
                vt_res_langs = 0

              if 'pe-resource-types' in vt.additional_info._asdict():
                vt_res_types = len(vt.additional_info._asdict()['pe-resource-types']._asdict())

              if 'sections' in vt.additional_info._asdict():
                vt_sections = len(vt.additional_info.sections)
              else:
                vt_sections = 0

              if 'pe-entry-point' in vt.additional_info._asdict():
                vt_entry_point = vt.additional_info._asdict()['pe-entry-point']
              else:
                vt_entry_point = 0

              peframe_md5 = peframe[0]['Short Info']['Hash MD5']
              pe_packer = len(peframe[2]['Packer'])

              if len(peframe) > 0:
                pe_api = len(peframe[6]['Suspicious API'])
              else:
                pe_api = 0

              if peframe[3]['Anti Debug'] is None:
                pe_debug = 0
              else:
                pe_debug = len(peframe[3]['Anti Debug'])

              for t in peframe[8]['File Name']:
                if t[0] == "Executable":
                  pe_executable = len(t[1])
                elif t[0] == "Temporary":
                  pe_temporary = len(t[1])
                elif t[0] == "Log":
                  pe_log = len(t[1])
                elif t[0] == "Database":
                  pe_database = len(t[1])
                elif t[0] == "Text":
                  pe_text = len(t[1])
                elif t[0] == "Object":
                  pe_object = len(t[1])
                elif t[0] == "Library":
                  pe_library = len(t[1])
                elif t[0] == "Binary":
                  pe_binary = len(t[1])
                elif t[0] == "Autogen":
                  pe_autogen = len(t[1])
                elif t[0] == "Web Page":
                  pe_webpage = len(t[1])
                elif t[0] == "Backup":
                  pe_backup = len(t[1])
                elif t[0] == "Cabinet":
                  pe_cabinet = len(t[1])
                elif t[0] == "Data":
                  pe_data = len(t[1])
```

```
                elif t[0] == "Registry":
                  pe_registry = len(t[1])

              pe_derictories = len(peframe[0]['Short Info']['Directories'])

              if peframe[0]['Short Info']['DLL'] == True:
                pe_dll = 1
              else:
                pe_dll = 0

              pe_detected = len(peframe[0]['Short Info']['Detected'])

              pe_signature = len(peframe[1]['Digital Signature'])

              p = x[3].split()

              size1 = int(p[0])
              size2 = int(p[1])
              size3 = int(p[2])
              size4 = int(p[3])

              size_file = x[5]
              entropy = x[4]

              cur2 = con.cursor()

              query = """
              INSERT INTO `data`(`md5`, `vt_codesize`, `vt_res_langs`, `vt_res_types`,\
                    `vt_sections`, `vt_entry_point`, `vt_initDataSize`, `vt_productName`, \
                    `vt_originalFileName`,`vt_unitializedDataSize`, \
                    `vt_legalCopyright`,`pe_api`, `pe_debug`, `pe_packer`, `pe_library`, \
                    `pe_autogen`, `pe_object`, `pe_executable`, `pe_text`, `pe_binary`, `pe_temporary`, \
                    `pe_database`, `pe_log`, `pe_webpage`, `pe_backup`, `pe_cabinet`, `pe_data`, \
                    `pe_registry`, `pe_directories`, `pe_dll`, `pe_detected`,\
                     `size_TEXT`, `size_DATA`, `size_OBJ`, `size_TOT`, `filesize`, `entropy`, \
                    `type`, `family`)
              VALUES(%s, %s, %s, %s, %s, %s, %s, %s, %s, %s, %s, %s, %s, %s, %s, %s, %s, %s, %s, %s, %s, %s, \
                        %s, %s, %s, %s, %s, %s, %s, %s, %s, %s, %s, %s, %s, %s, %s, %s, %s)
              """

              cur2.execute(query, (x[0], \
                vt_codesize, vt_res_langs,\
                 vt_res_types, vt_sections,\
                  vt_entry_point, vt_initDataSize, \
                  vt_productName, vt_originalFileName,\
                    vt_unitializedDataSize,\
                    vt_legalCopyright, pe_api,\
                   pe_debug, pe_packer, \
                   pe_library, pe_autogen,\
                    pe_object, pe_executable,\
                     pe_text, pe_binary, \
                     pe_temporary, pe_database,\
                      pe_log, pe_webpage,\
                       pe_backup, pe_cabinet,\
                        pe_data, pe_registry, \
                        pe_directories, pe_dll, pe_detected, \
                size1, size2, size3,\
                 size4, size_file, entropy, vt_type.lower(), \
                 vt_family.lower()))


          else:
            errors_in_vt+=1
        else:
          errors_in_peframe+=1
    else:
      errors_in_peframe+=1
file.write("Errors in peframe: " + str(errors_in_peframe) + "\n")
file.write("Errors in vt: " + str(errors_in_vt) + "\n")
file.close()
con.close()
```

D Python Code Example for generating .arff file from DB table

D.1 table_to_arff.py

```
import pymysql
file = open("/Volumes/Backup/Master/typ_t_35.arff", "r+")
file.truncate()
file.write("@RELATION MALWARE\n")
file.write("@ATTRIBUTE vt_codesize  NUMERIC\n")
file.write("@ATTRIBUTE vt_res_langs NUMERIC\n")
file.write("@ATTRIBUTE vt_res_types NUMERIC\n")
file.write("@ATTRIBUTE vt_sections NUMERIC\n")
file.write("@ATTRIBUTE vt_entry_point NUMERIC\n")
file.write("@ATTRIBUTE vt_initDataSize NUMERIC\n")
file.write("@ATTRIBUTE vt_productName NUMERIC\n")
file.write("@ATTRIBUTE vt_originalFileName NUMERIC\n")
file.write("@ATTRIBUTE vt_unitializedDataSize NUMERIC\n")
file.write("@ATTRIBUTE vt_legalCopyright NUMERIC\n")
file.write("@ATTRIBUTE pe_api NUMERIC\n")
file.write("@ATTRIBUTE pe_debug NUMERIC\n")
file.write("@ATTRIBUTE pe_packer NUMERIC\n")
file.write("@ATTRIBUTE pe_library NUMERIC\n")
file.write("@ATTRIBUTE pe_autogen NUMERIC\n")
file.write("@ATTRIBUTE pe_object NUMERIC\n")
file.write("@ATTRIBUTE pe_executable NUMERIC\n")
file.write("@ATTRIBUTE pe_text NUMERIC\n")
file.write("@ATTRIBUTE pe_binary NUMERIC\n")
file.write("@ATTRIBUTE pe_temporary NUMERIC\n")
file.write("@ATTRIBUTE pe_database NUMERIC\n")
file.write("@ATTRIBUTE pe_log NUMERIC\n")
file.write("@ATTRIBUTE pe_webpage NUMERIC\n")
file.write("@ATTRIBUTE pe_backup NUMERIC\n")
file.write("@ATTRIBUTE pe_cabinet NUMERIC\n")
file.write("@ATTRIBUTE pe_data NUMERIC\n")
file.write("@ATTRIBUTE pe_registry NUMERIC\n")
file.write("@ATTRIBUTE pe_directories NUMERIC\n")
file.write("@ATTRIBUTE pe_dll NUMERIC\n")
file.write("@ATTRIBUTE pe_detected NUMERIC\n")
file.write("@ATTRIBUTE size_TEXT NUMERIC\n")
file.write("@ATTRIBUTE size_DATA NUMERIC\n")
file.write("@ATTRIBUTE size_OBJ NUMERIC\n")
file.write("@ATTRIBUTE size_TOT NUMERIC\n")
file.write("@ATTRIBUTE filesize NUMERIC\n")
file.write("@ATTRIBUTE entropy NUMERIC}\n")

con = pymysql.connect(host='localhost', port=3306, user='root', passwd='', db='peanalysis')
cur = con.cursor()
cur.execute("SELECT type from data_typ_t_35")
row = cur.fetchall()
lst = []
for x in row:
  temp = x[0]
  lst.append(temp)
st = set(lst)
temp = ', '.join(set(lst))
file.write("@ATTRIBUTE type{" + temp + "}\n")
file.write("\n@DATA\n")
cur.execute("SELECT * from data_typ_t_35")
row = cur.fetchall()
for x in row:
  file.write(str(x[1:-1]).replace("(", "").replace(")", "") + "\n")
file.close
con.close()
```

E Data contents

E.1 Types of malware in data set

settingsmodifier, spyware, exploit, softwarebundler, ransom, dialer, backdoor, program, flooder, trojanspy, remoteaccess, trojanproxy, hacktool, nuker, pws, tool, trojanclicker, virtool, virus, rogue, spammer, adware, trojannotifier, spoofer, ddos, trojan, browsermodifier, joke, worm, trojandownloader, misleading, constructor, dos, monitoringtool, trojandropper

E.2 Malware families in data set

prockill, chalmah, yellor, spyking, yellow, tvbmw, suridel, harverster, rmtsvc, mooder, totalvel, scold, sevenc, xtc, lnkget, vorcan,
itsproc!gmb, teginim, lilibu, sixer, lysum, tuaye, rivanon, delf, nedsym, sidesearch, wimad, hideproc, kemed, donise, fakesvc,
bogres_8192, y3krat, adload, ufraie, prizm, skeeyah, lime, refroso, madlet, smfin, simile, smuky, egenav, phuckpow, hookmoot, jhee,
lockdowntrojan, solufina, headunt, cdinfector, dragonirc, higuy, thunder, dalgan, sisrop!rts, errore, tearspear, champ_5447, tenpeq,
splintex, widget, dorv, webct2, daxloyn, dosdekso, arials, ardunk, cult, buxthude, donaldd, dobrokot, eps_150, nacluv, karin, vobfus!dll,
dearis, vobes, sters, darksnow, mowalker, squishmoo, flashkiller, iissnaki, unis, txt, robonanny, ducky, feartrojan, moudoor, dnf,
niblenyo, vbswg, regexploit, yungs, qqeye, fifibe, nirvana_194, music, stegvob, hatfiend, doser_4540, runmalwav, vbswt, fatcat, notin,
slaper, troxen, mexer, yahoo, zemp, tiropot, xine_785, neoval, hole, hola, quimera, ulrbot, yemrok, lnkwinkap, bladi, qqmusic, blade,
famec, blueboy, teman, pidgeon, rc3srv, blackhand, dbl, example, clientman, growom, spygraphica, bzup, titanium, agentsmall, wurmark,
multidropper, riggin, absolute, hezhi, teldoor, hoa, legmir, lepasud, fosen, getqqpass, qqdodo, dnfkill, erjayder, ultisteal, fbsr,
backcgi, shaitan, bogoclak, rumortrojan, lazyadmin, ariver, saros, kazkaz, pbox, finsgra, sherlol, popureb, vance, buzzy, winp, cheri,
vmunix, eteraw, noknok_72, comitsproc!gmb, dambiel, myfip, wanremote, opa_1103, avenuemedia, alfdoor, rasmin, infector_1_3, loring,
weccer, fndialer, disnight, fiu, mianoot, dumadoor, maliciouscleaner, brbkd, remotrol, cinmus, ypdl, passdel, helpud, 7adpower,
doumol, nekotimed, winshare, ghodow, tejanos, pirdir, biscker, rbtg, isphack, silver, momibot, subseven!scanner, fi7, remotehack,
sisproc, xtcp_201, bestpics, arrow, ingrid, klinger, windowinfo, kidala, intell32, dftpserver, deepdo, blinder, winduke, vebject,
barjac, helower, resur, pagebomber, golem, bigbrother, blackblood, smw, zar, magicshadow, sms, masot, rc, rednuker, rm, dse,
spamvkon, navid, nightmar, citifraud, lactine, knig, defsel, systemhijack, elfnotify, rozak, superstealthkeycapturer, ftp99, yanskc,
shipper, zaprom, driplexo, manyasu, rahima, civut, optixpro, doly_1_6, skena, yaludle, debear, selfoner, wass, erase2002, wast,
sillybor, redkod, heibai, hidewindows, vonvod, zperm, sisie, windang, tango, versys, glyn, funmov, rpcnuke, nirvana, pakes,
zombbomber, iisalien, debworm, tiop, blackbat, idly, project, pigsearch, aslike, webdir, heyya, apre, este, napalm, runner, vxgame,
getproc, vidro, vbinder, superutilbar, angelfire, kaaneut, bussdo, comdlr, dasjad, bubica, netlog, zpsm, nukerbdr, redcod, passhack,
geratid, wreck, dameware, treizt, breach_pro, goldenkeylogger, win32rootkit, anxiety_1399, erdam, bomb, wizpop, goldenpalace,
waprox, uaaia, aproposmedia, prix, otello, tracklogger, onver, seekmosearchassistant, qqrbot, lamedon, clicker, buso, branan,
zombot, mena, remotest, ftrap, magicon, eiderf, festeal, scabin, spawn_4096, kaiserlog, levona@mm, remotesh, jeemp, anxiety_1823,

anomail, patch, pandem, rimecud, passmonitor, vbkill, gsmfree, pinom, bizac, cozit, killonce, spyload, blem, fakenuke, ghostspy,
fdar, cydog, sawmabs, comwitproc!rts, silkrope, voicespy, ghostmail,
tearspear!gmb, hetis, legemir, acropolis, leoboard, junksurf,
vncpass, memoone, exploder, futray, neoark, brackash, smelles, degrabi,
fishtrojan, paxok, stealth2, hagkite, createext, genue,
ring0, zellav, wmsender, kangker, winevar, tophead, vbot, matcash, outspace, akosch, nosuh, niya, fishlet, manifestdestiny, fritear,
zombget, doyay, winskr, madang, muce, regul, upxscrambler, telner,
qqphishing, ronoper, rammstn_14520, ellhide, aarica, yalat, spig,
spie, chili, doth, seben, popagerty, popsenong, conlock, intar, lanfiltrator, yosenio, undersor, pecodrop, mailploit, alcaul,
drixed!rfn, aegrus, fotos, sirmiras, nawai, dyfica, deadcode, principale,
honk, supeva, gorganik, copifex, ovogut, oicqlog, nictech,
drstwex, voyager, qqgame, slaman, tops, hulzic, topo, dra, adshow, ylyshell, fareit, altar, cubspewt, orez!enc, kker, beway,
cophlap, sjg, algus, cataclysm, tenuks, oblion, blinerarch, fankr, noesis,
gsecdump, givin, rpcfw, lassa, jinnir, modnar, suceret,
acidreign, sometrouble, parparo, bnc, traeger, kxhack, haw, systry, flecsip, ursus, putotrojan, iagold, svcproc, gernidru, shadow,
danglo!gmb, d, desire, goround, tds_muerte, fimvuc, irapt, smallfeg,
gangplank, anykeylogger, phorpiex, gurnak, gadu, warping,
warbot, ganelp, klepto, sipack, crowl, serialthief, kuge, sillype_180, lecna, crowt, warring, creep, antinny, backorifice, lockdown,
vampore, fox, soth, gokar, passkiller, fog, mindcontrol, hmtoolbar, okror,
riern, sporkeh, morbex, collec, wobwhor, binder, capside,
otran!gmb, delito, gspotbot, hiderun, pdpinch, ixepre, adex, cws_smartfinder, pendex, junet, cluts, dotor, rebut, druagz, phpnawai,
ddoshy, downsys, caspid, tuareg, mbt, axload, sporep, calldal, lifort,
acidsh, qqfile, qqlog, wombot, omssun, nova, incef,
killdientes, higlieder, jane, thirdeye, munfor, orifice2k_plugin, mutopy,
rapet, insurrection, sierrem, piron, crpexp, gorpid,
urlin!rts, piros, begravost, msnfaker, duel, getpassword, saje, progdvb,
shebooren, smasher, paycrack, ircskin, cridex, fakegoog,
dipass, marburg, aolcheeky, digitalnames, scarab, killerwebdownloader,
vecnadoor, tantic, xmightbr, zindos, canesdusk, mincra,
cyberpaky, glacier, synrg, elerad, nosyst, flood, zany, aolprudentor,
prayer_1_2, prayer_1_3, smell, bohmini, sendfile, f_door,
lorex, nakrom, lorez, bettyboo, psyber, qqhelper, mendwar, buzbav,
aolinstantmessengerpws, korad, mizzmo, yabinder, gruel, swfdown,
jolise, blocco, bushtrommel, shellpinger, bo!client, push, smbcrack, uloadis, ntrootkit, winbudget, firecracker, front_0_2, gh0std,
remoterevise, vnckill, osa, 180solutionssearchassistant, exestealth,
addshare, hvlrat!nuker, lara, fubalca, orbina, infospy, netlip,
gwbiner, bakaver, jerk, shadowph, neeris, nemesys, shodi, frapes,
projecttrojan, jusabli, lamerlite, maha, floodor, ders, leave,
phorse, uloader, team, babonock, redzone, helompy, multiweb, fupgrade,
sincom, oecede, solris, xdoor, genme, zombam, stotem, salvme,
acidookie, denisbee, saranwrap, msnsend, deepgal, lnkhit, apler, zrm, axec,
edown, spysheriff, googster, webmoner, wixud, upolyx,
alpha_842, webmoney, wilab, conycspa, malsia, gamedoor, swaduk, plepilm,
dilaga, velost, nabshell, slenfbot, xhx, redemption,
minicomm_1_2, subseven!patcher, projostig, pigax, avgesi, lopinto, dervice,
jenix!rts, fakr, love, jevafus, ypaul, ipthief, fakerm,
saburex, vanbot, sdtsd, rupass, kimiki, penepe, hammerbinder, instan,
duberath, 8192, danglo, slupim, hacktack_2k, benden,
icqsmiley, dopip, cnsmin, calgary, 29128, simospy, nitloc, skun, uandme,
skud, lamot, dwinlo, tiner, acrocra, affc, greap1, neshta,
marcelbomb, whibonline, coced_236b, sgae, remotcon, nether, icqcrasher,
funlove, browserpassview, multex, alureon, passview,
sandrador, vake, iissnakeover, silverftp, delevid, spamthru, provder,
widsoor, hider, chantal, fakesecsen, tenpar, ticton, bluefire,
parkchicers, coldfu, gason, qqtail, isapass, dosnuke, keyemul, layrui, atak, lastas, xazm, jingbay, omexo, worbe, snape, lanzon,
webex_1_3, webex_1_2, atav, winshoot, sv, fiasko, malagent!gmb, belial,
dedler, vika, psychwar, se, wesurf, robobot, tachi, sse,
wintools, smahler, slingshot, floodbots, fakedll, exilfed, conexyo, avid,

rask!vftp, hecscen, kazult, vharke, hallowav, smadow,
logmod, lacrow, szxagent, xupiter, flyst, floodz, matcher, xolondox,
insomnia, tborr, oberststen, rewardnetwork, ahahd, gmsec,
tarca, sklog, jman, possiblehostsfilehijack, naninf, winzlock, revert,
simpat, simpan, maxifiles, reverb, kuluoz, mokead, vundo,
fakeping, miniblacklash, hookja, cigivip, virdic, troll, naked, shareall,
percol, diwink, verver, winical, ybinder, bla, pasur!rts,
rattib, cissi, idonatebho, rtkit, yacspeel, recon, killsysbckup, rundli,
cryect, hamer, jorayser, avupdatescheduler, furby, alcop,
webservlite, trabin!rts, coredoor, zhengtu, acidhead, mysidesearch,
harnbrush, radalatan, tenbot, assira, almat, bankash, vote,
chubby, dialerhub, sluter, brickder, bancorkut, witkinat, inlook, shiver,
sozz, imkata, bits, qakbot, findthewebsiteyouneed, nuwar,
jecam, snatch, fastcounter, socomigo, antic, antim, wychegra, naras,
irccrack, opdoc, mimon, powerloader, biohazard, backstabb,
tophac, savant, opdos, qquse, mserv, gongjitance, 28672, grnu, bolzano_4096,
lydra, jodoor, qqgetpass, sab, getpass, bissldr,
inforyou, nidis, kipodtoolscby, nots, rainsong, simda, note, icqdbkill,
webnexus, youpeer, click_it, shellbot, ackcmd, navup,
buffer, molecule, subroot, kelino, drefir, priwin, biwili, alpich, delicium,
salo, dialxs, pesin, bady, wing, writetokrnl, spabot,
targetad, regrkeyset, dtcold, marijku, maldal, sasser, jpegbuilder, cobra,
ircexp, banger, tvbvk, glaze, atom, bocmen, kagee,
blandie, tubaret, zloyfly, behz, mapsy, rask!zmail, symesta, agemo, robo,
neojoin, agemt, sloc, lolliport, datupwin, usbcillin,
assistant, paloc, sisron, amokjoiner, dtservice, hunclo, esfury, priest,
roger, katalog, perun, morose, turla, dosht, tenrobot,
espy, neospy, xpehbam, dexter, fearlesskeyspy, glutoho, qibongi, emogen,
msblast, massmes, sparv, apparition, domb,
stonedemailbomber, elitor, adept, taragasare, qqthief, reendup, junkdata,
fbound, sshell, flder, passdev, netspykeylogger, spoolas,
madcap, qudos, littlebusters, temporizador, ziquy, belio, sepaibot, pucodex,
pahac, mofeir, hidedrv, derspehe, avp32patch, sentinel,
wratch, apdoor, uropoint, xwxia, venik, stretch, ihook, casus, msshed32,
sillywr_242, shetr, tongkeylogger, tomek, postb, notpa,
wnuke32, fainli, enstan, ceckno, nayrabot, oflica, ms08067, anthena, pacerd,
brantall, keygen, kcom, kwsearchguide, pophot, popclik,
phvg, tongbot, xcan, internetspeedmonitor, cinmeng, icqspoof, tompai, influ,
revert3, nautical, meredrop, nucscan, cashdial,
newmoscowmailtrojan, brok, brof, dkskeyspy, yessim, showgame, radonskra,
licia, spung, inpolah, ulysse, ilookup, pornograph,
scorhost, lookme, cazinat, trirat, oxypumper, zumamumy, idons, archmime,
killler, malintent!gmb, eltusk, keystate, genie, pakks,
sysmon, fivfrom, danschl, udp_bomber, slime, pwkiller, hader, sasfis, perfwo
, ratpacker, parlay, afainternetenhancement, wintcp,
invictusdll, generator, hooker_32, amivida, invisibleftp, zedmac, anpir,
doschald, segax, delautoexec, silin, stekct, cabac, zion,
elitper, niwer, snart, vimm, dark, regin, ranfruct, buizit, dajenmoc, darf,
servlice, koobface, dlena, banwarum_gen!dll, lmir!dll,
fakedel, tsunami, aksula, frlite, inter, phonehack, nuttymouse, plteam,
yoddos, tistorm, nucscan_a, grum, nucscan_b, darcaler,
kaotan, rdpopen, divine, minipg, dimbus, surfaccuracy, revenger, snapit,
safesaver, toto, ddv, vbsmw, yalove, proxybot, tubs, refer,
ulratt_8166, tapivat, orifice2k, bymot, suppad, hljacker, akuan, ursnif,
tapiconf, smuma, twobotkill, freequickkeylogger, eslog,
ciscokill, holar, tigisda, evader, masaustu, nettrash, shien, guapeton,
matrix_1_5, samsteal, luzia, cybercer, mysqlhack, crenufs,
blohi, ridnu, sethoc, colddeath, ilafor, netraider, mdm, cisum, symbab,
emperormailbomber, missu, dictator, minicommander, diskfill,
emesix, apher!tunneld, skuffbot, tourniq, drubot, carem, cloner, mice,
ptakks_xp, bundy, milkwed, nukeit, gataka, rago, tript,
darker, protlerdob, sckey, ab, ag, paradrop, ancev, vwealera, doublewm, ao,
watcher, aq, rads!epo, qqhijack, valcard, at, aw,
momaker, deludru, deatheye, az, screencut, beeinject, sking, collector,
anset@mm, rpclsa, nospace, filecoder, firtal, msgvb, puppy,
advanfer, kelihos, pestlogger, aimtoolz, pasdael, starsdoor, ms04031,
rasflooder, taskman, flics, natalie, aimpws, thorn, nerty,

snagger, mimic, masp, cayen, mutapager, senna, udpfl, afbot, gamqowi, zappa, resod, autooter, caomap, zmk, baupish, ts, arape, smile, homepage, ra, kuang!updater, isq_boom, explorezip, checkin, getaddress, projet, qqspy, hoazts, adpclient, emuvito, regform, shania, sany, snuke, bafi, icqbomb, small, hufysk, joiner, vbstupid, skowor, livup, obitel, dumaru, dmailbomber, weakas, xot, fulo, pwsteal, hase, cleanmbx, nonam, dapsol, miliam, fudnimp, jinmoze, syswebtelecomint, backlash, jeakail, vorwahl, littlebuster, sality, vlades_30208, cmailer, prior, salite, arhost, mapson, welder, catinea, weirder, spidor, corty, vig, sajdela, vih, vip, warray, ronop, zangosearchassistant, silva, krepper, under7, cardspy, morw, trivialbased, blathla, mmosteal, bedienks, fgdump, cbeplay, getadmin, caca, aleph, rkdice, winsys, eagleagent, nucleroot, bajar, knockex, tnedal, zbot!rfn, malu, mostime, tiberius, firehell, malf, rarcon, dongdor, sinred, mehm, downalbu, proget, santosa, subsar, dancerbot, dcomii, cabrotor, scar, underattack, drivalon!epo, axis, comscore, hearty, quetnek, dodo, blueang, togui, udp_vb, notre, speedz, messengerskinner, aolplop, sranda, proxyagent, deloder, inetwatch, mdmbot, izeburn, misoska, prolaco, dipnet, sonide, ieshow, logduck, dctoolbar, primat, gisayar, quicksearchbar, clickspring, dcpter, crypaft, cpvfeed, bdkerlen, aimchalex, zippy, aispai, windowskeyanalyst, kazeus, outwar, frenzy_1_10, sillybot, bypass, coledor, buchon, surila, fibimser, remplu, giku, fastfind, cih_killer, synfclick, delmyk, dedo, st, trhunter, hortiga_4800, chobi, tinit, tespid, fono_17152, chucha, fakeexe, daumy, hotkeys, vanti, planet2k, webnavi, basichell, gizer, apocalaps, mumu, ptsnoop, diabloii, begger, muma, doser, portuk, mumo, micreg, dllpatch, lanspy, syswin, syrutrk, slivwab, nutbus, scorpech, neroma, passkill, badgame, silka, iisindexserveroverflow, dollflort, medel, movida, syserv, chatscroll, dabew, blammis, anedl, clicktosearch, maniadoor, tagbot, belanit, thekiller, xiaoho!corrupt, pvstealth, pilsen, synflood, sillysharecopy, cd_open, momac, cookieupdater, klinge, mobilkiller, fiasco_2500, secondthought, girls, karabah, advo, rooter, kibik, thoper, guap, orbyt, wildek, jes, unexplained, acidshiver, valsday, iehlpr, speil, pspy, keyloggeronline, aimtroj, bispy, beastdoor, admedia, antiman, phreak, fakeflash, powerspider, hifihi, qukart, lypsacop, essgol, plurp, niklas, reur, grivlog, akez, swiran, kifie, formmail, eslac, rainsong_3925, codbot, dexple, bho, compux, folderfu, madtol, spookdoor, lendoor, msnmaker, qdel, avgold, vbnoet, selc, internal, comproc!rts, analogx, trufip!rts, krynos, thunker, shareme, roron, sysinst, prosti, winrc, virut, prosty, restud, trkshell, enzio, wisho, ext, gnotify, tps, bagsu!rfn, torun, cafeini, sctune, satir, goli, anirak, ojo, coldfuson, repus, indicator, asorl, zombkeylog, cimuz, funso, urldistract, fruit, drexonin, barkiofork, yurist, mailhacker, zephyrus, badboy, nvrdoc, banned, qozah, ieplugin, wiznuke, netcontroler, noxcape, lameweb, recto, hikjav, dupinject, doly_2_0, mutabor, dowgav, telserver, barrio_4_0, trilima, strysx, rfl, insane, sarhust, emar, exploider, boza, resmu, jade, bombita, sankey, redlof, azrael, froobot, shtheep, masspager, shower, zesrac, nachi, mymus, cabinfector, backconstructor, movie, myba, rebylt, upspider, knat, rakum, netsphere, muter, camserver, vabekon, bangsmoop, pirat, soder, death, cg, superspy, gamehack, dopedoor, killfiles, fratong, krikun, sadam, whitebait, natali, connection, amoeba, foxfox, luder, lash, vapsup, actem, grobim, leon, fibermil, soaliid, justjoke, dizer, syph, devil, filth, wping, escritorio, baigoo, belang, angping, cpush, psybot, synte, netcaptor, devir, pwg, ryapspy, kunta, iebatost, squirrel, restina, autonuke, lbuster, silencer, blocix, remoteconnection, phia, vaprop, ming, give4free, netmon, formatd, slydude, msnscan, formatc, fusic, ipfak, kurit!rts, error, duppy, klanei, kirsun, mota, lykov, scretbus, motd, bzub, maldldr, cerebrus, agsurbot, barok_2_0, faitypelf, epizak, dsklite, dynaweb, cih_1230, spysystem, decoy, rameater, lornuke, stigmador, nukeit_inferno, tapaoux, kifer, decon, wiedreh, inotofier, bubnix, wideman_8135, mastacash, netthief_xp, slackbot, padania, doal, organer, jolt2windows2000, joff, xalanga, dasener, spyanytime, rustyw,

nilson, johan, bancos, theef, salva, csysserv, meve, sima,
greenstuff, mxsender, sadoor, onaht, aimip, relbma, sillywr_204, dilna,
sysroot, kirbo, banjori, keylogmsnx3, deflo, shext, zofo,
randya, format, sssrv, vital, screenfly, bo2k, crashcoool, verweli,
passwordfox, milena, sping, eqstealer, ivaz, janker, netget,
portless, decrypter, lerma, netbus, vorofer, susftp, foxers, gagness, wuprad
, wordperf, mailspy, bounce, topantispyware, scelp,
greener, leniv, wactier, kavar, bendor, ratcracker, puebot, lamin, hackpass,
badtrans, rasare, plunix, getwrong, temple, haxdoor,
aolxinon, msnfun, netmetropolitan, bo, zengtu, bj, kizmagis, booreen,
cortheaper, santa, bs, mocmex!sys, bx, lastad, donetkrypt,
tnsrv, subot, vecex, hatchet, hatcher, gipneox, lethic, ambush, winur, iwing
, cmay, krakrues, arbinder, fakepriv, aolps, plsex,
gichtymessage, yahoopass, javbomber, ryex, zhymn, zlob, span, spam, cyclede,
purrer, rudov, fooject, spax, crackdown, dcrypter,
suit, atrar, spav, xombe, magiclink, upatre, peeper, ipcaller, eliles, jacky
, lostorin, feplec, dealply, flatsurfer,
smoothie_2_001, asniff, troxen!rts, wanhope, uy, exdis, cih, dnsbust, qqph,
trimore, billatan, galaxer, nmcrack, hornet,
lorez_1766, tarnid!rts, topguide, nulnuler, dog, dgssoftwarekeylogger,
aholic, freeone, chat, zhangpro, lazchen, flocker, phantom,
ixnuke, rapzo, exus, spaels, toha, fakedef, matman, votead, badiseso,
sappwort, manipulator, famudin, sidex, kreasy, iptakks, mood,
agm, land, fighter, vzm, lana, tufet, tohaz, niquim, noala, walker, mireye,
lizardbar, smadlpk, iemm, bvg, ganub, sillpy, mosuc,
nimda, bvm, slimbraju, yeltminky, ppapp, saiterec, teardrop, floxif,
yahsteal, derunsex, kapup, shipup, binres, kaput, resourcer,
easygen, pluder, zamelcat, alcan, darro, offkey, ainslot, uploader,
iexplorestrojan, reopener, magic, eva, dkshell, webdl, ms05039,
jbook, trp, qqdragon, aggrevator, lecivio, hinweis, wowcraft, ragterneb,
repka, exter, kolpak, carufax, macur, macup, sfone,
pirril, powerstrip, pnuke, indel, kiltex, quzah_3361, whinetroe, netmail,
dropegg, zegost, mariofev, bauka, ponmocup, exebind,
comitsproc, zmutex, ptakks, deshack, porndialer, panderson, rdr, opop,
controlrandom, titanic, extreme, lisiu, wansrog, maker,
althi, egolet, ldpinch, wtool, exe2vbs, rotdown, stresid, boxp, gaura,
crowter, francis, powerspy, aeon, sysrentor, btrumpet,
doorplus, oxon, cve19991317, sexdownloader, veediem, infamy, ehks, freetrip,
entangle, thiz, this, norcis, mtexer, hotworld, mrija,
thif, espion, zafi, xmahack, dreammon, renoper, benf, ip_protect, comronki!
rts, envolo, meliksah, toprebates, threepigs, petrale,
hookit, somthub, intraspy, frenzy_2_0, cws_startpage, lamb, npk, inethlp,
bionet, fuhd, society, hawk, bfcboom, afigen, chirem,
itsbar, remotedesk, luna_2724, finkmilt, borde, dansh, acnuz, buhound,
sereki, qname5, canahom, keysnitch, platcyber, wiessy,
bulta!rfn, aletc, ganipin, kamkam, gipad, decay, gnusan, lamicho, gcs,
elitec, danmec, babylonia!epo, backstreets, avogu, acoola,
rlsloup, stopin, tavosa, reateltex, kinster, snaky, newstick, goner,
doasearch, ultratt, illlog, fakeinit, vbbot, kangen, adstart,
pux, xine_1371, piptea, zippedfiles, banaw, mumador, massflood, virumulu,
qhosts, timsy, wuftpd, unzi, neveng, fratool, banan,
jethro, googite, matrix, shruggle, cabanas_3014, dnsdoor, ickiller, matrim,
venom, nsskill, envibr, dbit, reqletter, shut, easyon,
croman, shuq, digitul, qdown, icenipto, birdspy, tempx, zolder, david,
borges_8192, outsid, achtung, staro, anita, spilt, poganec,
softwar, spywin, feardoor, happyday, dronzho, agfest, lom, negett, neglemir,
iglamer, icecubes, huanot, start, ullysse, compchec,
kapart, fotish, systex, lefgroo, aolmovie, clep, clidak, fizzer, asianraw,
havex, fakemanga, argos, winmaximizer, elite, fatboy,
qqfake, quickfyre, gayb, aggressor, narcisus, vecnagen, bind, mabezat, lwsta
, winsy, angeldos, bogres_12888, regrejaz, kuto,
optixkiller, webhancer, neblso, datarap, toolbarpartner, dumador, bylist,
mrtype, clicktonet, benatic, ressdt, cryflas, ezaki,
petala, icqnuker, poetas, xark, brmonitor, smallrk, blackeyes, vnuke,
skyfire, fexacer, migmaf, kreeper, antilam, fiasco,
sagnusnagta, imvb, snoodsnip, fluxay, mulcss, kooblog, terminate, nikomac,
anaftp, tapiras, regrun, petador, apher!shelld, buttonf,

terminal, score, dirrename, umrena, milconsd, jaber, dudence, treemz, tidola, boonana, finaldo, sfiles, lucky, ldplog, slinger,
matsnu, kkmaka, prado, sillyfdc, sinus, wmfa, smash, nish, count2k, bolzano, wimpixo, hekdor, crazybull, aimfaker, yonsole,
fouldopner, voltaol, godwill, packes, lamar, andover, setez, yldiz, setex,
bleda, delshare, nuke99, destiny, platan, nuclear,
giwin, platak, sendworm, misun, clickpotato, yasv, icqgreets, yahooboot,
kotan, goldhl, faceless, darpa, plingky, spbit, progent,
bilay, badrite, zagaban, ms05051, stealthspytrojan, microyano, lioten,
spylanternkeylogger, beksnoc, betyunk, cih_1003, inviter,
botten, ch, poscal, scspy, ca, reatle, almir, frucake, cd, icub, harnig,
fakeaud, nenet, ataxia, cp, kilonepag, kiction,
linkoptimizer, vakcune, mailman, odsrootkit, kaht, uhop, butano, icasur,
apspz, feliz, pity, legacy!epo, felix, sinite, regmbu,
punad, remexp, nirky, hlpadd, neojit, klovbot, meteor, ikytoky, nullbnc,
banish, airbot, mauz, foryou, decept, lanc, sith, netsup,
vb, golin, winatch, tesoit, gobotools, sillyvb, fosforo, golid, ripgof,
taketeen, sub7logger, ransom, blitzy, lowzones, hostcont,
defun, outbreak, udsofo, finets, duso, upof, blackice, magef_4768, supeboy,
redevil, loveadot, netfutur, infinite, lastscene,
snilis, dock, byterage, mypower, tefor, feebs, schtuk, dap, hackyou, tonmye, rozp, xel, livewire, sytr, moiba, adclick, glue,
deathbot, freakazoid, combra, astaber, gwee, zhangyan, conhook, crack,
eclypse, crazycd, tuesoy, dest, tcmd, rumsoot, mirpn,
sniffor, tick, tirtas_5675, ainder, gcash, zangomediagateway, daurl, rankos, rungbu, isus, dafly, peiwah, boclay, rasaper, arhil,
ritacin, vsu, sinbat, thief, braban, miniasylum, flush, trenk!rts, auha,
glyph, shonk, poot, hadsruda!bit, neman, deathcorner,
siver, coolwebsearch, kloop, rifec, ravenpass, recerv, phorex, ayam,
aphexdoor, bymera, langex, crazvimes, stuxnet, provis!rts,
mmort, spudashup, shiznat, pacak, revise, condrag, magistr, surrogad, stator, pukich, luna, syzor, erect, autohax, pazus, autospy,
sander, testagent, oledbservice, nescan, stalni, userpatch, dcbot, tudo,
muerte, onalf, cheeshodi, chcod, sillyc_1536, ginwui,
friendgreet, togod, harvester, loxar, kenston_1895, notime, r2d2, veden!rts, kolabc, ebomb, apathy, bootmerlin, malamaged, lewor,
sukwidon, comame!gmb, licat, dichas, inc, erbot, luhn, hps, chubo, wozer,
cnnic, hpt, avanzado, chtong, igmpnuke, tooner, illusion,
estatic, monday, lwpw, delosc, kangkio, scapur, metibh, plexus, inspiration, iepassview, ducktest, aimaster, fanta@mm, relop,
possiblemalware, ertfor, maresa, lestat, xecat, rimod!gmb, sumoxt, dotf,
microjoiner, hepstax, mdabl, cookster, dynamer!ac,
autokeylogger, javel, stub, regap, virtumonde, liji, baxid, runux,
mental_10472, chifroms, bibei, bika, delfiles, cdargen, pedrp,
elitespyz, avril, eira, fursto, bizten, gvanto, cugirl, tofsee, redrival,
smog, adkuadu!rfn, inlev, acbot, topex, fpack,
winterlove, hadoken, samurai, roxrat, streespyer, infdr, difupat, chet,
wbesys, chep, pcacme, ntkiller, chex, xolxo, ms03043,
specx, yunsip, qqrob, ms03049, sidemax, amus, sendmail, pubavid, lizard,
partypooper, junkoil, vchain, mirsa, boomertrojan,
reversable, crunch, hackscr, appeldorn, rewindor, pluto, patchload, revell,
ipmail, fragglelite, chrobancos, clindestine, newon,
idele, nosys, avkiller, pstrojan, glu, mafchek, meltprox, limper, ifrasif,
nakuru, loower, simda!rfn, beenut, ultimatedialer,
fosorm, huopass, esimtipy, fotomoto, newjoke, instonarch, jestouch, ezu,
dyzap, kittrid, windupdates, cutdown, xobobus, ircgame,
loveyou, nkd, slanper, ideach, splad, nirbot, deborm, iprs, roram, adrotator, ardamax, zdesnasnet, ineudok, enterus, last2000,
tribyom, hideall, internetantivirus, eclipse, vldial, filk, xqdoor,
theripperz, seido, aolwhord30, scopiron, gloria, idtsys, uhil,
kript, arpoc, mvc, allight, nucku, yever, testworm, verify, dunik!rts,
axesoft, russianjep, aimmorph, wdoor, sillywr_184,
winpopup_feur, oderips, semisoft, pizza, rebaw, ixeshe, suidown, hilgild,
ladder, karsh, joulwo, thingy, msntv, tinyrun, botintin,
arch, sserver, aquafish, tsumi, hucsyn, berbew, passsender, netcat32, besys, chiller, buttman, backage, order, aritim, asergmen,
webinstall, rirc, tromax, advancedmailer, fantador, lorexp, exwin, hdbreaker

, fugor, lizgo, younga, nerphun, httptunnel, gnoewin,
protmin, hackboy, cartok, dmail, sadon, petspy, breat, linkdirect, chifrax,
sador, payback, woker, reclog, webnav, snowmoon, radja,
sisproc!gmb, rebenok, alavar, backsocket, telemot, snoopyt, roxy, pepex,
cabanas, elefant, webber, gatina, cputhief, multimsn,
alchemic, shellcode, backstealth, mailnuke, sekan, mohdem, trafix, hailiag,
pkeyspy, lolita, felver, daniel, hooker,
shopperreports, qtaz, neptaven, ohuru, fervis, maxyx, sakuz, syrius, icebrak
, blmoon, cabis, ircdeath, mutilate, lollipop, oskal,
macrocrypt, firekiller2000, maniac, picazen, peansen, hpmail, mnets,
cloudine, nuclearuploader, chengtot, tarifarch, gakivod, vqv,
cmdexer, rendez, predatortrojan, loselove, mtx!epo, cornfemo, unixon, whenu,
testmy, autorun, tostgos, expiorer, shoho, bombsquad,
alerternt, tiebho, fearso, mystalk, nowayout, walwas, perser, apbost, targer
, sodager, xanadu, iyus, faxu, digitala, devious, iroc,
bluanweb, blackangel, microspy, maylook, aelf, qqkiller, icqchat, tronic,
rammstn_12504, ditul, srendiv, indexer, ladex, popupper,
zimbo, nbe, cheeser, gatecrasher, dfg2002, vingard, hawey, stration, ihsix,
kcuf, sconato, atrojan, netboy, sykepager, r3power,
netbot, ayegent, antifw, numnom, aolnetboost, masyanya, atoaol, fivsec,
dopen, tishut, ircobus, zoder, firehost, rewin, sillip,
fasong, befast, medfos, aldy, stercogs, jortel, havar, sharke, hddl, gunz,
mellpon, dm, hostcontrol, db, egroupsexdial, woroshif,
mancsyn, dt, whal, clidem, mepf, rebirth, tcstrojan, ema, winos, wallormee,
begin2search, dantmil, t06, sachiel, diload, udps,
trion, steam, kanots, nukeattack, buplik, ciosor, bo2k!config, septer,
minikeylog, lowksoar, defmid, ostatok, bla_2_0, srvcmd,
eret, gentleman, subseven!speech, neaset, nathan_3476, apher, awak, miwavlen
, xauboy, aphex, neodfg, zeeker, danginex!rts, fidar,
rootodor, strexec, lamado, airostor, finetop, cow, tival, rdpbrute, cariez,
riberow, mytonel, rnrpctl, ucri, stealthmail, spear,
jifcapi, bizhor, overjoiner, gtbot, rodando, zetno, irritan, scparm, 2000
cracks, kheagol, falcone, callinghomebiz, qqfish,
procesemes, cromptui, julikz, fanys, wukill@mm, aolpics, systrace, cycbot,
sinis, attacker, comfoo, necurs, sysadmin, ident,
aesevin, nusump, vorus, ais, langeweile, aim, halflemon, mydoom, symten, aic
, aidid, setix, nifras, sticker, liciaa, cone, cong,
inido, darktrojan, caxnet, platen, isnev, singleweb, jacktron, cpuhog,
zensor, spion4, treden, colorer, plik, lithium, bunny,
kotef, raid, rain, harbag, hanz, creamos, sleepyone, skybag, kwdstd, thw,
eps, delsystem, wysotot, goldun, kanuk, pykse, thg,
ccproxy, udod, thc, sonic@mm, juegos, kala, extor, byshell, hagala, kalm,
smawis, icausur, lurker, mimail, waspy,
virtualhackingmachine, sivuxa, antes, zabdo, agentbypass, bigfly, sharaqq,
shador, viking, aras, maya, boxed, koblu, micetic,
iepassrecover, mysock, woripecs, mocket, fakehotmail, yabran_3087, mobibez,
xbot, baronnig, aibolit, supova, dahorse, jaqusim,
pocztylion, dilber, blinkom, soar, makecall, casbo, acset, masta, salie,
rile, wlock, steph, qqplug, ceda, bimstru, remotenc,
ypager, demiz, cmjdown, girigat, eicar_test_file, wangy, diatwo, phel,
msncrack, pher, dayek, gillver, cabby, kabak, kabal, herpes,
trojbrain, keepcar, clisser, demonkey, bropia, yubaceds, cabanas!msgbox,
banker, pelfpoi, hotmack, octopus, samex, blakage,
safadrew, multimail, zanayat, serok, shodinwat, mousedisable, febelneck,
exrec, mielit, delov, barkibloom, rscrt, tercesph, way,
nonov, dashvolex, zazorex, war, starae, mecool, becoming, bormex, veva,
fraggle, chalcol, klogger, nevsyn, antiaureatespy, yimfoca,
mtx, isis, qunap, mtv, crystal, reven, whyg, kenfa, bedril, seppuku, sdboter
, evidence, faceold, zalim, prayer, evul_8192, mole,
vwc, sinit, mooplids, roombuster, dlder, test, daiboo, jucons, laoshen,
update, roaller, updatr, omega, downexec, suomia, dialsnif,
coulomb, tapeworm, pacer, archivarius, ketch, podnok, nethack, mayhdoor,
ylang_1536, popica, igloo, trance, vefisi, whcrew, remoab,
pornik, sogou, datus, globar, dalton, gearclop, executant, delfobfus,
subattack, syspas, icekboy, flash, bolzano!epo, graps,
fakefolder, terror, musdie, eorezo, kittex, expoden, excalibur, doombot,
kazus, towshin, hirhir, blast, oanum!sys, scorfake, hanlo,

qpop, schoolbus, zeagle, smee, schoolbug, instiopen, cdgluk, hgweb, beomok,
critopex, hojan, nullnet, lacon, grump, vasnasea,
geniuscoupon, cusrev, cargo, asylum, offeragent, plutor, kimejkay, havoc,
sharer, chota, aimtheef, wote, unbeka, sharec,
poweredkeylogger, karagany, aoc_3649, botex, downloader, oblom, agping,
desktopscout, netbull, wolfst, aznime, remhack, winmx,
precursor, rabiggs, leebad, dmoniz, ppzombie, headline@mm, vanloi, rodecap,
ieinj, fakegina, bitaccelerator, bo2k!setup, lankiller,
cynto, seratin, farfrom, daosix, hackers_paradise, aufta, live, webdialler,
littboy, uppriv, bisty, torvil, masterparadise!tools,
dler, dasboot, qeds, caw, youaru, bstroj, yiha, keynet, purga, iframe,
vbsencrypt, buschtrommel, nadebanker, spy, xefes, chir,
tazi, sps, usteal, gigex, spl, ledor, perdex, recoder, mmcs, gattman,
helbsly, shadowphyre, verfst, freezer, emptybase, hexzone,
brontok@mm, htbot, tiptuf, oneclick, fallingdoor, msmh, aoc, pudorat, aol,
searchupgrade, aos, vagrnocker, algen, katar, wpakill,
jerwin, pld, unitemail, kgbkeylogger, avkillah, nowim, kabiyur, mdrop, brain
, braim, botao, mixus, glstorm, bumfa, defin, atirus,
cdenor, mendware, nyrobot, rorex, mooze, xbotor, qqcat, rores, ntrc, hunuke,
stridor, slacke, recal, evom, drathmot, tjm, perda,
exploter, zbomber, propho, adrenaline, nox, arsys, fakepowav, instantaccess,
droz, drox, xtra, mamov, vindor, drol, nuclearkeys,
afcore, eggdrop, newpic, fierads, ej, smgsrvp, anubis, backport, netcomp,
mutihack, ft, ev, yarner, mxc, algebra, fx, fakexpa,
rivon, recatl, cazdeg, steredir, conficker, cara, cuthem, lme, fakefreeav,
zupedoor, kreton, axent, msnworm, zomshc, waxi, drasher,
hackaject, kufgal, cmsound, skubur, blehdrop, blink, fakedigitalspawn,
exrand, rina, zap, psychard, sheer, spoolsploit,
invisiblekeyloggerstealth, lisfonp, freebid, hellbomber, dolan, tefil,
vicluder, driverbypass, icqcess, skreed, primativ, passmail,
volac, gobi, neetro, mydopam, dissed, elitebar, dissec, bsoder, ncast, prast
, allsum, slap, ainjo, verticity, bbsxp, adbehavior,
fonly, crazyworld, qrap, forpi, alieen, benuti, sticamint, suwener, greetyah
, killdisk, sledge, bsdi, hiddenrun, apocalyps,
soapspy, roxin, qqhacker, brutus, drastwor, forever, maya_4153, xingdoor,
iredor, ruchytwo, elgolf, kabwall, prorat, enumcreate,
mrc, osiris, anfiz, hddkill, batcrypt, copuper, briewots, headshot, peana,
webloit, cacogen, bafruz, malres, nemesis, themelt,
tilcun, dfma, picer, niojec, inpect, iorgut, danvee, kikz, veosma, icqsyke,
commonname, svost, antites, baglet, kika, leud,
vcell_3041, ircflood, udp_ipt, dummylock, servudaemon, hostil, dabrat, ramus
, spyeks, comroki!gmb, amanda, trialdest, moneytree,
netadmin, angelrevenge, crakreo, bukash, badcon, lnkhyd, ascienc, serubsit,
chksyn, darkddoser, yacra, mirle, muntsib, kevor,
easyserv, grenld, netvaiser, vrat, spindel, thenix, decryptor, spinder,
hallodoor, warhol, shermnar, mona, nuqel, cabornypt,
leviathan_3236, comet, battlepong, diamin, inttest, jeans, comotor,
helldriver, cevstn, vobfus, cadombi, orida, younmac, zipparch,
addlyrics, efno, sinresby, vucks, alcodor, mangtemi, amani, urbe, dkbits,
mefs, rubi, purplemood, facetwo, kplo, ishbot, trufout,
forcud, infinaeon, radoom, firejoiner, fireby, bo2k_plugin, winko,
enterring0, bibrog, ahgepad, clickelkite, chepvil, lanxie,
filunork, servstart, erah, redspider, drv32, mainserver, spooner, gecedoc,
labrus, iisaa, datdrain, epoch, riprova, retbar,
espoleo, lite, rysoft, flewon, paladin, secef, arctic, selcrypt, swisyn,
predator, secet, katomik, raleka, golember, sms_vb,
tiniresu, nonaco, flibot, barraf, fakebb, mapdimp, peopleonpage, parasmall,
thunderkiss, cd_die, meltor, zetronic, birdihuy,
malcole, neol, gabeerf, neos, neop, spiliwan, worton, yangxiay, sextracker,
dobom, verst, symop, cobsan, umbald, sheldor,
netmaster, envid, psn, psm, viset, ruler, ineta, orivion, visel, kextor,
skyflower, skopvel, fortune, appix, malux, oblivion, cx,
mediket, zgoo, zlug, dusvext, netject, logare, gansip, flvtube, kernelpatch,
nedky, xscript, smser, lorer, matchaldru, keyhook,
xinfect, kelopol, piaoyes, iishmd, squire, leniog, flood_2_1, flood_2_0,
botter, dindang, pistolar, vetib, aolbadboy, wiza,
mssqlhack, xbinder, neonet, qoologic, yaasf, screencontrol, dluca, nmb,

gamesteal, tds, delfinject, guapim, barrio, uwaga, chod,
detnat, chon, choo, talsab, wolf, hadache, cookiemo, viname, diablokeys,
tolsun, retxed, outlook, blueeye, dextro, liajred,
silentspy, youd, indor, kingbomber, gezak, kapa, morix, sadcase, gezad,
cinmag, kronical, nethief_xp, gojalda, moothie, stark,
lanxue, palukka, bornyame, stealth, derspeher, ares, rovnix, logici, tareger
, secondsight, comneop, darkshell, satur, maesorn,
iisstorm, podcast, bnlite, mailreap, setme, sednit, folpsy, freddy,
datkiller, hirs, funnyfile, eicardropper, dander, neuroid,
headtrip, ircnot, hanove, oanum, loops, sevensphere, sbytes, sheath, darkicq
, beyond, aclako, pswvg, collo, emuni, tcpportscan,
gyplit, qdel101, ciadoor, frentyks, nthunter, rbot!dam, gnuman, killsap,
spit_8192, lamebrk, linklooker, polan, kozog, repah,
cabdial, dexfom, timoha, repad, tebtair, rembomb, arianne, rumish, srecord,
ursap!rts, myst, chayka, rootbeer, bividon, messo,
correo, unis@mm, smasarch, bazuka, montp, kokomzn, stagol, carpet,
podcastbarmini, griffin, intexp, bobep, hellza, wmi, gaertob,
nuclearprank, aolbreak, hijacker, chopanez, wmb, tinxy, qqexplorer, mgcharm,
espamer, fakeremoc, titidoortrojan, rscdoor,
bombarder, mpx, finger, idyll_1556, trashdir, anar, eljefe, anap, hostposer,
chupa, youdon, drixed, abaddon, mousemunch, adson,
retun, rbt, apolipse, kyrdor, killqq, anonim, havkco, iconomon, temcry, mill
, floppymad, motepro, maslan, xiaoho, daoh, ceefor,
pc_ghost, hiberium, fakepay, bobax, sutsyb, stealther, excess, yewlo,
fakepav, xenozbot, dulom, zaurga, maul, soldier, daemonize,
ircsnuke, postalot, netdown, qqcv, pnbug, aimes, yerg, family, icqrelay,
20480, iefear, fatuous, praq, glitch, webaut, qxray,
hidwinrun, murkry_398, flashzero, taker, wineggdroponlinekeylogger, msposer,
zorg, bymbk, paytime, zori, pixar, linden, pazzky,
dp_dptrojan, dion, tumbi, hackerdefender, zokrim, fledul, excuse, kpager,
invisjoiner, brambul, pendix, vlogger, rodok, icqcrash,
aozo, sinredkeylogger, nutka, aupd, smurf, kather, coced, aftp, templar,
hidukel, rat_cracker, onever, heloag, sventore, celine,
sharat, lamirc, sharax, simouk, zevity, stepar, invisiblekeylogger, jezebel,
playgame, turown, cabanas_5116, meman, warpigs,
aflooder, botan, emseado, naprat, memas, fr, soltern, fv, maran, aimhance,
maral, sneak, achum, fa, fc, fd, fh, gast, morphine,
banka, bugs, amitis, ego, menajeto, hobbit, emcrk, kagratool, leebee, erom,
sood, minmal, indra, lindodia, apher!proxyd, lesper,
hibado, kollah, prowler, kogant, exact, redfox, ircspam, solvina, babot,
inikiller, systemp, meteit, duwork, targetsaver, hupigeon,
illwill, tinyrootkit, rws, douserv, fesber, tavibayers, qzap, proktu,
maradonaex, ifu, guzuloh, wunkay, loaderexedll, alfora,
storesky, nfwg, fakespypro, gismos, netrunner, crazyscr, efcommander,
handykeylogger, gopworm, senummy, gismod, halen, sinco,
fakeirc, overtron, sisron!rts, bocata, runeer, vcell, jolla, rewindftp,
spartadoor, turkojan, ftper, navidad, dexec, asu, dirt,
asy, swog, squida, lipler, whoisadm, glukonat, stuly, ilomo, wisdoor, dire,
enchanim, toolzy2k, vxi, ash, basi, tridmerc, launch,
falin, lemur, benju, fimanspi, dicomp, kung, voob, miso, shave, ursap,
emudbot, verbcq, synen, xandu, topbinder, midaddle, pykspa,
stoberox, mistery_2560, winxdef, jedobot, syner, bombxp, ees, horse,
udp_storm, mydealassistant, ghotex, horst, goyaby, pandos,
serelod, ncx, doxel, myspamce, tfd, balisdat, mad_2736, bogres, getter,
egroupinstantaccess, boredom, msnflood, hasist, ludbaruma,
chumpoke, pobreme, nurech, std, micronet, sysid, allaple, funfactory, gred,
sounli, greg, kishk, shotgun, dtr, fudor, bloored,
swrort, verind, wintrim, ultras, testspy, pechkin, toush@irc, antonio,
ftpnugry, gologger, lopelmoc, bboxet, wormex, fyeo, redplut,
seimon, hidep!gen, netcoach, usbalex, deathmin, russoturisto, alphx,
blackcore, lynder, silly_p2p, dccunf, kaz, naupointbar, ncw,
gremo, thetatic, hipo, tosep, greencode, vicety, drews, idwi, invisdoor,
coolfool, ccure, petick, codtree, merve, keytrap,
ruxtrojan, iisadonai, spartdoor, redspy, alanis, pdfjsc, donvog, notchod,
hankydor, minilash, foont, francette, botsinley,
paranoia, phantomftp, zaman, rewdulon, aolkaimx, vatintov, netres, cryptdru,
ordy, desktophijack, poit, synspy, bloodlust, regneva,

dinov, arwobot, utilman, asniffer, millenium, erser, hanserv, imiserv, costaro, iceroe, enumplus, datarape, ngvck, netconf, clost, kazaver, breacuk, newsagent, werle, delud, gamarue, rtms, keisan, douglaskeylogger, dumaru@mm, diemodem, habbo, shown_539, dursg, chevaloi, startpage, jinto, elicz, fakesmoke, coorat, zipinfect, oitorn, hatigh, tweder, jsgen, mytobot, offbot, cosol, fakesub7, chanser, zipper, corruptedlite, nervos, mega, ronater, winext, evil_953, webdialer, hvlrat!ip_stealer, sensitive, anon, seeearch, vesser, zinject, judora, protos, freecd, peobot, ejik, wholdor, nukit, bunitu, fordel, muzkas, haradong, abfewsm, ade, damn, icqpager, damm, trafog!rts, winsatan, graone, trinoo, reveton, suqin, msntrick, netcom, pinkle, irccontact, msinit, swlabs, eace, hackology, sagic, gendows, fluxdor, falraxco, bifrose, dedeymex, hegel, koceg, guide, loop, asune, conteo, mydoomer, profilestylez, shoerec, oberal, zener, odrtre, bestrevenue, kwak, quozah, kmky, eicrypt, edetok, shah, doomber, 12355, acillatem, optim, usem, showword, regkill, 3dstars, fenosy, boa, ntpacker, optix, mobi, fbhack, mintal, silentbanker, authstealer, jomloon, aoltony, dogkild, heantyk, mobs, hbr, kiescreen, syszsl, xpupnp, aim_vb, iconscroll, iddono, marque, faisal, ipcscan, ierk, winspy, zonsterarch, labirint, mesoum, inikill, oporto, injshell, stealtheye, orsam!rts, deadduck, caran, trovate, smartdove, grozlex, wingo, deberia, vodvit, ruff, kuksec, nytworx, aolfader, barriotrojan, darkftp, stabanarc, dycode, mykralor, mincer, donny, mypicview, agenttiny, virdrop, pestil, taripox, qmailer, taripos, buminpom, prier, subseven, sofra, savage, rumale, batzback, bbmail, butitil, welvirus, fxsvc, aecofil, proha, ootlt, chiprocks, lasta, dogstop, familykeylogger, donbot, dibos, infra, rux, crackedearth, frenzy_0_10, redesi, yewbmoat, padmin, checkesp, gnome, trats, parved, okupok, ygebanel, hvlrat!ip_scanner, inido!rts, gunbot, letum, slogod, srvdrop, flechal, intelirc, msks, kumsum, alpha, blebla, abresive, puxadoor, dynamer, visal, ghack, firehotcker, fileka, napsin, batosecu, spywad, hiloti, mothyfil, wildfire_3040, fedripto, zbot, wopla, harex, listonosz, serversockets, smokedownloader, vcryptoz, dfgbot, restarter, mmviewer, moncher, bumerang, dfind, irtih, fakeplayer, gecky, virtus, rashangup, apropos, killwin, jider, exemas, anuir, nan, nao, renater, disconnector, vemcas, limraps, dislex, racos, dasher, raideloz, drat, fobluda, bombthehandy, evar, vcktk, pabueri, warclone, sassdor, tropid!rts, amhathyari, yakoza, gamesprank, freemz, dremn, activitylogger, maxsearch, e, cmjspy, conedi, dealhelper, svc, sva, bamaboy, heckyebo, locproxy, hukle, sqlstorm, navrar, smeaglo, haxor, fono, pitke, gr, islafile, gz, gf, sillywr_166, psycho, sillywr_164, sillywr_162, gh, barok, webdavo, zhongsou, keebie, msnpass, votwup, blogchina, bluerain, rosyba, aris, spindest, pakabot, runescape, maax, ikslog, xircon, cashdeluxe, comame!rts, ppcore, rimcoss, gubed, aoldreamcatch, olan, exite, dandan, guber, contiva, unicle, raznew, remdobe, dabvegi, slhback, oroch, dnschanger, dpbot, click, hspam, smsxender, porchanspi, gizmo, sapi, vampiro, colonel, dci, sysbopt, playx, playb, vundo!dam, netsnooper, cleaniisweb, dotcomtoolbar, dabyrev, delarm, scythe, gamevance, sfind, becon, warece, convert, sckeylog, genu, banload, loaderexe, ajim, vidsase, sefbov, rokhshah, opalcot, chinbomb, wit, avpatch, bredolab, donips, burbul, kilie, crat, zigper, cray, semail, word6embedd, hydra, cashback, cheeky, altnadel, zexnus, xantvi, nukers, hateyou, murkry, ciccio, dragonbot, daurso, parvo, smitfraud, denyo, cosmu, starter, stafford, hacksoft, renos, wmdeath, cve−2006−3942, levelone, renol, amenby, nopadex, kenny, tvido, huntsou, hotmail, ffile, tiram, conscorr, pizbot, neurotik, dvbkd, poetry, neurotic, ranhidi, vuronmot, superlamer, drpupcdatamanager, blueball, tramal, fakeyahoo, taskplaner, glieder, xinxin, flexty, secretservice, crazer, purityscan, crysche, ipscan, lordly, bloack, popad, dahrwam, heres, cudsicam, biomac, ping, gillich, hrat, rootkitdrv, jubon, puzlice, vasdek, aolpwsteal, eps_166, smtpclient, ghostvoice, winfuck, enermix, qipi, madro, madri, pixel, chatspy, chmbuf, frone, webdown, nibu, clolo, darkmailer, rexx, zonebac, masteru, ppdoor, facker, dzyckz, ticlick,

banca, spamtool, etricks, ap901120, chansact,
bo2k!workspace, gedzac, cblade, ultimax, configloop, sumatrix, nukeit_krust,
hls, miji, ricta, olilat, bruhorn, skor, sms_bomber,
mircnew, reboot, taplak, revealerkeylogger, devsog, dosenjo, achis, xeigh,
mofei, theilllogger, adawar, intruder, leshiy, wincred,
nnkiller, hostile, aolstatus, fundoor, typon, keen, whiteboxmsn, aconti,
banof, lamejocker, mad, matit, kiray, powerful, ars,
lokup, fbsdhack, 27648, wabot, hermanuploader, cerebus, hvlrat!listmaker,
alcarys, ceekat, demina, purol, noname, puron, smym,
cih_1010, hebogo, flyagent, rsc, edior, winmite, ftpcentre, oicq, rsv, kdar,
cyn, soonheng, sambus, sambut, tocofob, netdog,
redghost, bee, sitebreaker, bex, sambud, cih_1019, ginop, bkill, wmfmaker,
crazypostman, artelad, audiodoor, delpbanc, upfudoor,
sacri, inetrack, udp_spam, sk, lolaweb, tirant, singu, champ_5714, switch,
parol, salrenmetie, singa, atadommoc, bagif_10090, coke,
mspawn, mouseloco, renall, gontu, yagoda, hookgina, chinoxy, filenail,
cysbar, msntwo, rcmos, masterparadise, wbecheck, portpro,
dervec, scrambler, babybear, zdesnado, gesa, nagram!rfn, netadvance, wow,
deface, setrix, ryknos, msverh, crackwm, tty, vroyan,
mutech, xtcp, rudoct, zenosearch, evidpatch, aolrun32, dmcast, darby,
patchreg, regcontrol, baser, shinny, vrodirb, b, wmhack,
cene2, grisch, ohmsrot, healsock, negamsa, norin, badcodor, newbiwo, djoiner
, comisproc!gmb, fix2001, druser, adialer_gen, alasrou,
imker, meltip, remotepwc, iisprinteroverflow, megasearch, grab, poisonivy,
grad, kamafe, neintab, desurou, sulunch!gmb,
neodurkjoiner, destroywin, freeze, sqlexec, crybot, mailff, tegsol, yosa,
playgames, webext, stats, comroki, nightmare, hawthief,
wumci, joyn, bjcg, hanacash, sabotage, induc, paykiller, sole, soddsat,
pluginman, dofoil, ratega, laphex, inost, dorando, hoepel,
vextor, tihack, sniff, qfz, dccfcker, coldfusion, purora, shoucast, opachki,
angryping, plankton, fragat, bluber, twores, gits,
danginex, avstral, orig, rotker, dogrobot, neomailer, yaz, yat, domcom,
webdial, yai, pebox, yak, tulu, quin, qqsender, fente,
silentlog, rysio, sqlcrack, winrat, renegad, comminet, fakesmav, doep,
iefeats, badrat, zuten, sabus, yayih, roomkiller, matama,
wkk, anonebomber, cve19990412, demo, zakahic, ulratt, ive,
webenhancementsmedia, remokil, sehtr, grobodor, zxman, sharecopy,
ginadoor, generic, kbluploader, baord, tinba, tscash, asank, vip_4309,
donoth, msidebar, kident, filemail, softsecure, netsnake,
parple, henged, ashley, hidedoor, orochi, gaslide, phile, skyboot, drone,
xtrat!rfn, rohbot, grenam, gnil, advertiser, dosh,
wotbot, antivrus, jaredex, inservc, aicore, ructo, msimbt, webcont, valentex
, gepys, autorooter, starskbot, vidc, loktrom, logged,
oogon, detar, adodb, logger, yabran_2226, tipikit, sashiel, botgor, tracur,
xmedia, bulilit, imel, colevo, smets, azirk,
napolar!rfn, wkysol, pointfree, lager, notfa, aolwps, rkproc, shen, mailpwl,
psyf, coced_236d, warezset, redrac, tuil, coced_236c,
fakeexplorer, mrmofie, hangup, draktor, wallon, towloh, badass@mm, hookinput
, swizzor, cachedump, jiwerks, mewlec, fawras,
netbus!portpatch, lazymin, icecubes@m, digund, lasttime, tvmediadisplay,
lamlite, repead, tonick, chonker, caznova, cleanb, silby,
ssonce, bebars, xplag, sipoo, wollf, bombat, frintorc, winexit, zilard,
chazer, devsis, backterra, naughter, adtomi, staster,
proagent, ghost, eak, phyiost, ultimx, nupic, blamine, yoyo, hj, melder, hl,
xerom, ha, timese, colmatch, dmb, datod, fatsee,
storiel, limin, cruel, calimocho, dmx, itis, lolyda, blingbling, nohoper,
bitar, buwah, delfdru, duddie, lohoboyshik, drach,
netsend, nanspy, folin, katien, zomrat, mofin, yabran, sepukku, fennarat, bo
!prot, netthief, seehole, lurpen!rts, star, sqlinject,
visilin, canvas, ultor, rebhip, bedienks_221, bedienks_224, kitrap, delflob,
chekafe, machime, bcb, siller, exeheader, adroar,
winemm, taterf, mailbomber, robofo, negt, antiqfx, pgpm, hacyayu, redsip,
pasana, prodoom, sideon, buddy, qurl, roingssearch,
mangwam, wirekeyview, aur, upload, gougou, selex, smorph, qqttff, yoohaa,
adfram, prodex, botmailer, saynob, monkif, connecti,
libdoor, realtens, inrar, metrag, winicab, hikiti, he4hook, realxj, cropo,
hermanagent, hupigon, comisproc, foxma, damailer,

internalrevise, spyhuq, easto, cachepassword, strarp, warno, ajan, remotesob
, methoss, proklog, horope, panama, asid, bodgy,
desktoplightning, kokegift, vavico, cheng, abot, mypo, eventor, spyda,
testapi, sjfs, fakeplus, aquar, sennaspy, webro, dreb,
dswebdownloader, neveg, dref, indofren, pimpco, fofeet, passwordmailerpro,
msnkeylog, neodurk, spacoom, croves, kraimer, xapo,
nineth, hunch, littlewitch, ardurk, weegexy, dripper, infotab, hahor,
intruse, karud, maltparco, musomar, newworm, akbot, calm,
forthgoer, pcrasher, chicons, killav, lecna!dha, lamfest, oscar, vlades,
joanap, loony, lazarus, frenzy, mustlove, heffer,
qqnukeall, netpass, apent, rightu, trojanman1, surldoe, goicu, winprof,
dynod, akim, bomka, icqvampa, mywife, masterlo, cyberdex,
xexe, oylecann, mtmpas, mitglieder, tusem, sathenvir, polif, nbname, wxpse,
fitmu, polip, infostlr, rukap, chatcrash, lurk, rejok,
fribet, subsari, xpaj, winshow, vfl, janis, aasp, mhtserv, sqlexp, configsys
, cramtoolbar, fakescanti, rusparail, zopt, zapemli,
gelsnopi, applog, mugly, batcloner, foxeyes, cyberjack, perfectoptimizer,
winltmpv, smallshare, tick_7936, whynxy, mail777, kolweb,
kukel, wun, freshbind, tibick, vbhide, srbj, ezoons, crew, remhead!gmb,
htrip, rootmon, grupong, neus, cred, brizol, sohlink,
hotlix, cylent, gestron, sybuex, tentacle, controltotal, prohax, mutny,
sibind, anis, srotgnat, tasmer, kazmor, redbind,
sfkeylogger, lazar, badby, bonk, kerio, tufik, anarxy, bone, flirtip,
perfectkeylogger, anarklik, navm, istbar, malagent, hectic,
24t, topsec, nemqe, memedia, rebooter, phishbank, jermsg, remdruk, sahay,
kies, talex, bo2k!embedtool, vidsrs, gippo_1030, kbind,
girlboy, pitchfork, nerte, lurka, msnkamuflao, kroter, tepille, exebundle_2x
, daserf, revenge, my123, keykeeper, massaker, keser,
medialoads, shaggy, tripian, messager, deemo, clickmaster, malamiko, sqlhuc,
murky, isa, delanaber, comrerop!gmb, hooy,
subseven_2_1, haiku@mm, spyboter, bionix, vividi, talkstocks, mmsassist,
enumerate, jjb, searchforfree, dalixy, goriadu, zerok,
sinisteruploader, rdiframe, ordy_1024, netdemon, infis_4608, taskplanner,
bcryptforcer, lovelorn, tmos, rasdialer, yildiz,
formmailbomber, sysrater, netcrack, anxiety, stupen, duramp, wuke, silly,
fedup, wintcpkill, winpwd, xmail, sdbot, yoursitebar,
mine, xmaib, ksniff, minx, seed, rufoll, servubased, alvoc, kility, busky,
lapka, lanxueqq, wealwedst, komut, hohack, escrit,
satcoiru, magania, rogobot, qdel92, wgetmo, koko, myset, vingrad, gbot,
mario, bildan, pulkfer, winad, christ, musht, dol, dom,
dob, meganuke, annoyer, cakeport, dod, porner, tumarc, shrin, look2me,
pukish, shrif, zaphal, dou, ultimatekeylogger, aenima,
helpth, speedup, nbspy, easyget, liza, hido, lacur, 52736, kaynutt, cabanas!
release, netspy_ii, opalcot!sys, arpa, pisatel, skypii,
renbang, dialripper, pmexe, arpspoof, badfree, adialer, himera, sunacha,
fignotok, hellzlittlespy, icqsnoofer, kubik, sskc,
diskadmin, velopsdru, trood, gabanbus, pesete, bugbear, navexcel, sulex,
padop, keylogit, adbreak, reload, cservsys, bac, sulee,
stinger, floodsave, virtualave, schoolbu, qqshou, ybad, dllload, bingle,
onitab, lexup, multidl, armageddon, snuff, rundis,
ogimant!rfn, simplese, blhouse, dialui, carfrin, rugzip, anonmail, arcer,
zget, warezx, tandfuy, injeven, relnek, azak, mlwatch,
coolsearch, azag, savles, eggnog, lammerlight, criminalmsn, mucks,
mirkillerv, murvb, runcrypt, mpaccess, dialdoor, gatak, winnie,
tiant, loader, tazry, wenper, cesca, xtob, perintal, fastsaveapp, smibag,
sms_chinas, icq2k, esepor, lovele, eggtest, yokoyou,
tiny, kolbot, saingat, riler, opensetream, citeary, gnot, ajja, spybo, mcf,
spoofbot, kotu, mcb, kispy, iisthcunreal, stratork,
mocip, passwordstealer, dateet, apikey, tank, melter, prizzy!epo, reechrot,
rawsekid, panamas, psychward, nethvost, womble, serai,
sockstun, aliser, serab, pedryak, spawl, freefall, subnix, fakev, ifnapod,
faker, sevec, rorpian, senekil, euthanasia,
deltasource_0_7, alamar, deltasource_0_5, mooma, fakea, avirtrojan, chooket,
kitro, belle, yeeha, darkbot, maka, msgate, broadoor,
zomjoiner, laorenshen, chafesi, overg, malagent!rts, offerboxbrowser, elly,
vhorse, gogogovb, atnas, expout, emissary, nedal,
jadtre, virila, obfp, pandora, roma, mapto, jkozd, jusp, xeplore, bo!

butttrumpet, zestlox, unfair, zom, grepage, melting@mm, hrvg, wergimog, netvaa, linuz, borges, borger, neher, xifos, malushka, nitol, depamd, vaklik, cryparie, hounddog, httserv, opa, spiga, cdudoor, daytonas, dozdl, afire, kgspy, chamel, vdx, kaze, insta, spitfire, poebot, nodom, eyesow, byzer, hlamc, kaliber, dead, ckiller, revel, fakecog, jupites, xpasslogger, fakegal, freekeylogger, ubriel, hocomrac, keke, carta, chimoz, nakter, zakiller, burm, burl, lamechi, promon, mocbot, astix, antigen, tetris, platrew, suslix, zux, relmony, orez, cobiterz, woned, zum, daur, mitglied, pexer, luladoor, litmus, noitap, dtray, chimo, nemsi, mushka, kolmat, daum, egroup, lanmen, carmapic, pager, vihu, flying, deadsock, broperk, burnox, mofksys, azero, sobomb, fakereprox, batman, gameplaylabs, fore, kipnot, berm, infos, alvabrig, alfool, zarp, lucuis, pameseg, banker!606e, tempex, passdumper, shiotob, trembler, kwin, selges, onescan, thunk, wiegrab, newweb, petik, blackcat, yobot, fradidr, fifesock, keenvalueperfectnav, steelrope, sofunny, mistfall, seetdoty, refoav, alerta, sizuma, ralex, rodal, dnstroj, ompnmagic, goptrojan, troj2k, fispids, ircbrute!gmb, hotbar, fakelogin, femad, femac, daily, axo, procinject, savno, winy2k, almanah, bluntman, radmin, milt, xoremal, wootbot, inteter, sysnom, tutosser, arman, bomber, xinkey, novadoor, markd, marke, gael, takit, calishoo, zepp, markj, pahkin, libie, backzat, hawawi, fimarker, qqpass, caftuli, shelcod, keco, etaclef, haxspy, subsys, sucon, widate, die, dia, goalweb, dix, webbath, riados, aimsnitch, kwbot, lookspy, lixy, aolsignon, psyrat, nulock, altice, dunpws, pikis, branvine, yodo, rutern, stealer, smwg, crashcool, sanjicom!sys, kuzhan, tds_se, bo2, aolspy, bartly, tikuffed, hackdef, yoybot, rpcdcom, sapade, vbinfer, deris, padic, blare, guardian, ositki, winnti!dha, afrootix, slow, y3kratpro, yibohbin, ordpea, invir, sillyw, sillyp, netdex, fakeants, dyflog, zengtwo, bom, sillyc, onlinegames, hobot, kolik, xooba, aolhosea, diewa170, superlogy, iani, g_door, godfly, lamebyte, andvb, habaku, pingdoor, qqpassover, swen, dude, connector, insider, drinet, nople, vakad, bindfile, sharpei, malat, remotesniffer, vidlo, bisar!rts, oob, aldebaran, golten, cashon, savior, olive, ziclfnk, vitero, bodsuds, inet20, soul, imper, kexject, antinod, aolshock, acidsena, cyjecter, zspy, romdil, agent, platum, t543, agend, drwup, pino, pinc, golroted, billrus, idim, skater, infector, nyg, evyl, karnos, soulclose, starcedor, lybsus, mav, grob, intmon, winerase, cxjnuke, secdow, rammstn, kasimorpi, msnspider, thedelphibomber, moerna, arurizer, blastit, wingater, exterminator, nsil, pimpol, yajing, tapan, basket, puver, cidra, almanahe, doji, lametoy, privacycop, soina, mirhook, liondoor, toviz, planetcracker, neorat, pointex, communicatortoolbar, tomoron, wardoor, damm_1537, shutall, kebede, panolis, ternanu, langly, auriemma, kibuv, tooncom, nvgra, enclave, killer, shatter, alician, kumer, ggdoor, music@mm, vxidl, mikcer, initx, lodap!rts, nepotemp, huskfim, rinscod, crypto, sillywr_180, isnup, dipti, mygel, ftpsend, wilba, softsix, lan, inbat, winshe, kerproc!rts, giri, zom_4096, stitch, uabapro, viologga, qqnum, ashcan, pamela, fitin, tecata, latinus, goti, zomby, monster, ckill97, netins, 14336, yurn, talwadig, phasma, winbiotools, schneck, sisron!gmb, perfiler, fanbot, livevids, reqlook, apagar, cbomb, filebundle, jusou, thonic, dzan, xtray, sevensaw, bladerunner, xtrat, cosisrop!rts, crix, blueadept, spampimp, metamorp, illnotifier, cospet, crazzywak, autocrat, netminis_10, lujer, inikill_32, murka, zombantiantivir, stap, rainsong!epo, remotepasswordstealer, uyours, evncil, zemac, mylife, kamikaze, bunkusa, meduel, bolife, smokejoiner, crackz, cybwar, ronki!rts, plage, privoxy, sicil, luge, liondumper, privateaccessplugin, mserver, stubby, axatak, hamaetot, chiki, moridin, boychi, aphexsniffer, aimclone, netcontrol2, k32, modtool, tramin, squad, kaboazo, sisproc!rts, ladmar, iwd, trace, diplugem, bagle, falhell, fakemplay, fmtdoor, brainspy, kallis, tmksoft, tabeci, voodoo, fiteli, kilah, partcom, justas, lunor, klogsil, polydl, salamdom, tonerok, shox, habrack, tinrepo, ftpattack, ringzero, poweroff, therapist, dadon, ahaid, mops, fropjor, msxmidi, rexer, matser, pluma, black, fanta, suc2k, ges,

mupbras, raser, intruzzo, mund, webdoor, newker, organiger, locha,
antidenial, snowdoor, gee, hippy, badluckreloaded, iiszang,
seleya, ecutant, kolilks, doser_4539, achar, rahitor, bozori, milen, mediar,
stupid, 53761, cgi_notify, emailspypro, asnifer,
fakeyah, apboot, fakeyak, yoader, sean, kermit, riccy, heised, sqlnuke,
coldlimit, servudos, zquest, fantibag, puma, jits, prevka,
cgsi, zhixingzhe, pyros, activekeylogger, troman, tetik, gaga, conspre,
avisa, tscout, screen, darkview, jx, spoobot, jv,
drogcatchaft, maguccipi, unsecure, gotecom, justjoke!tool, loper, coced_236,
insanenetwork, cyanure, ucsearch, dater, dks, poitard,
dreamscapekeylogger, destrukor, hitpop, uzbet, udpflood, dovan, swyque,
floopbot, dyfuca, sharall, lspp, deskpen, poxdar, zeriest,
penrox, mard, kotibu, pnew, bacros, reping, mari, mosucker, brajur, jb,
hdfixing, hftp, ridok, nimoo, savemoneyshop, alcobot,
ccinvader, clepa, bimoco, sluegot, quatro, sniperspy, toseven, homiak, wako,
nimos, evaman, sinkin, eplugin, fiansrch, twinny,
fenster, voober, shutit, keysave, wisell, institon, belocker, sahara,
searchcentrix, sexy_256, zonekiller, anonymail, bagz, samb,
tobor, frenzy2k, sadjique, dcpm, stormattack, tjspec, sudiet, lesbot,
clearlog, nephron, eps_104, amazex, otlard, knowlog,
delwinzip, bleemfake, parody, fumn, predec, gatez, enterak, preden, yorobun,
preder, losspass, cokegift, toshwire, wifer,
winspywareprotect, tinybar, paras, attkit, totalcontrol, aimpunt, imponex,
focelto, hetoub, chainsaw, kovirz, zule, phpbb,
scrawaked, servex, toolbar888, hiv_6380, zlo, lightning, mosuck, gwboy,
hackarmy, eayla, sneaker, thething, explodus, dold,
popuper, doly, doomjuice, sporkbot, sbaca, gimmiv, elitewrap, kme, spydance,
kaitex, loudmo, chuchelo, lunam, fxdoor, pioneer,
diskmaster, inject, homecrash, leoch, overlo, maoo, smilex, crazypc, ansis,
raptof, deltree, dihan, regexp, cresus, qscreen,
golsys, networkessentials, schwindler, qqbang, tserv, pornagent, msdivxdec,
udslee, mantas, zaka, puvbed, pitux, vbrabber,
cubeexpl, zakk, sst_kit, leguardi, partsiosity, hizballa, cuhmap, whiter,
nitter, wabrex, mark, yakad, fatee, nadiote, furitron,
hvlrat!patcher, oexsi, baronnight, gravit, broomops, brutu, retnecxp, pasich
, iyabot, keung, striker, misik, 28471, arranca,
hasmin, peper, gorm, leguardien, ketan, keyghost, iiswebcart, vhm, magcall,
pwrnut, blackbird, embed, ayospy, minigift, srdl, deev,
nekav, narith, autoattack, ircbrute, kuto_1468, pcghost, selfish, chatfuk,
bofang, aolexploiter, nflog, cicho, mirhack, zelda,
rsocks, odedem, iethief, huceqoo, paydex, badda, hackit, xanax, kadia,
delcommand, borler, diablo, astef, dropbox, cnaddare,
hortiga, wonip, goh, fakehadoc, windowsmite, skydance, javakiller, sosedka,
lamhav, kworum, banwarum, reyds, alrain, ambanguk,
rikitavi, zins, diesel, proyo, cuki, looper, lookup, zbot!ci, reipym, plesa,
darkmil, icmpcmd, hakan, juzkapy, telecommando, ninex,
netbetro, redroshex, igetnet, klgdiablo, fuzim, kaibi, bawmaq, arequipa,
emotion_4608, welher, modphip, rameh, bancodor,
wishmaster, alma, ikobur, ekan, totilix@mm, hyu, revaz, nivalid, selfcloner,
smsdos, mantice, lifewire, mespam, zero, callbox,
jasti, grameyoon, aphexspy, zerg, trafnit, visiotrol, ociyota, ribbon,
maneav, sickboy, aunps, zolpiq, dcom, gepost, fatfu,
thesojems, redhacker, malintent, coronex, showmypass, exitwin, tsunovest,
oicqsearch, flindir, servu_based, mircbased, farehacker,
daonol, ksiwin, beacon, simon, flvdirect, klez, actmon, iframer, drodos,
iezones, zeynep, zeynet, deadcow, evilgoat, difuca,
bliame, bunga, xorer, bombit, stupred, dorpiex, enar, texlock, darkirc,
delmed, easymode, busifom, sddrop, bober, fujacks,
savekeys, libertine, liber, elksoftfirewallkiller, splash, rxbot, viking!dll
, iisfpreg, keyspy, clixer, mozilla, zuso, briss, xobo,
httprat, coced_214, getmsncont, messbot, derium, wukill, limar, aolmeanda,
tuax, sehole, crystarsprotobomber, urbin, remotekeylog,
fosniw, sowsat, hlam, shark, doxin, closecd, ghosttoolz, qqhook, banealapay,
lolol, deltdstar, chimera, gewder, cyspetel, socay,
sisbug, ejtroj, homicide, ms0411, hackdoor, zbot!go, ntscan, saibo, gwghost,
vecnatool, shiba, gina, netbuster, garveep, bloon,
illic, pestaple, wazabre, viruswizard, emabi, visbak, sauratol, guorm,

pnguin, gagse, netexscan, roudvil, evadiped, cereg, blorso,
redosdru, munzter, pcx, indowing, newapt, younga_2384, rigel, hapick, balog,
dorkbot, dolly, epoe, epon, malex, goof, comame, gool,
nebuler, mobilebomb, ontarg, dufeva, wally, dnet, conducent, ruzmoil,
tlowdis, conip, gipla, flyswat, micro, sircam@mm, beavuh,
gedza, waber, icqrevenge, vextrac, mental, zmist, vtimrun, energy, rustock,
auryn, fisp, aolbss, mailspam, bereb, mentat, slackor,
ipccrack, angriff, magic_7045, bomak, skyfirenuker, crickticlener, pigeon,
gifnoc, pibi, tawsebot, devildor, demovol, bistro,
lustorm, netsph, emailbomb, sbg, sbd, netspy, remandson, refpron, aditer,
pusrac, myma, antibtc, prophet, hacarun, novlog, comando,
jimmyh, tsvt, phase_11, arpkiller, wecorl, naco, wpd, winfiles, pointad,
waltrodock, donn, keyrecorder, respup, kod, pleaz, jetto,
lapsavok, searchextender, killproc, aladinz, akak, teddybear, arcdoor, park,
conov, elfrit, aladino, eblaster, smsup, rip2003,
rikenar, banxpa, zwangi, nopir, peerat, msnsteal, lugiry, smf_example,
iisxploit, pirtes, oneclknetsrch, pahador, tetick,
begseabug, cavespy, sonick, zomfd, spiag, nussamoc, shockmailer, exploitie,
pihiker!rts, likpeh, mirkes, yitai, reder, nabony,
bayan, rayman, coremhead, adultchat, zaratustra, tsaisda,
thiefpasswordstealer, unisearch, trosup, jatodis, adson_1703, actualspy,
nostart, qhost!gen, sennaspone, taladrator, pitit, lanky_3153, xinia,
refdoro, bo_installer, lovesan, 007spy, bervod, vampire,
baidusobar, hicomma, likuner, proscks, finb, vbvx, fvcl, dasmin, stoplete,
cam2ftp, spynet, prpx, potao, cholera, express,
keylogger, shuckbot, wurldmedia, gewse, zaptusk, tupym, deltasource, mincese
, jlok, rampage, hlife, homac, doubled, voodoo_ii,
dlraser, vbload, cratorr, vine, senglot, mamianune, gdorm, aoltinurak,
offlinekey, exebundle, sysinfomailer, myxq, swimnag,
peepviewer, shakpics, jazuz, cewalk, mailbot, netpree, enumiacs_8192,
cableboo, xoox, infan, pepsi, fakeia, shaman, coiboa,
systrim, centim, hacktack, alop, willsienod, pilrurl, ciarin, gyner, simple,
hacolo, zeropopup, sexy, augudor, bearshow, xspy,
backend, triplix_3072, screener, systht, aoloscar, metrion_37204, doorspy,
roxor, gas, msbot, gao, flukan, inmota, dongdoor,
servese, nukem, reglo, bosseveryware, assiral, mbot, inuk, pclog, spycos,
buben, chekafev, remhead, probot, buzus, banwor,
aolhinter, pacenif, wedsnot, bytemagic, orbina!rts, propas, explge, snowplug
, southpark, trillident, spamma, msncookietrojan,
clagger, adclicker, plimrost, dunik, ws_ftp, stript, lopin, sym, sholta,
poshkill, injector, rockse, dowque, sumo, somex,
fakevimes, gol, zeprox, aolaph, gwelog, mesgra, ldl, netkey, drollit, hioles
, atomic, bddt, thespy, ubuster, antitcpa, tikfis,
snak, obfuscator, snag, ouper, junkcomp, punter, poinbag, ms05002, flatdag,
eesbinder, msconfig, sulunch, insom, phinot,
optixpager, parchood, koutodoor, bit, insect, mytob, mint, makeext, mispy,
fagot, freeinet, flux, ficoive, kenston, chuzy,
tepanele, cambot, braba, walrus, machinegun, sysman, storark, logsnif, duck,
back, schock, adlinks, winspybox, icqbomber, demowin,
xlbh, mostrar, pet, mansund, whomp, redblood, yoohoo, zaiba, adkubru,
gladgerown, hoptto, stealthbat, pec, qhelp, munga, clay,
bearote, hermweb, kryptonic, keylog, yobe, tswsvk, hvlrat, jpegeater, jimmy,
nsl, nso, elvdeng, iiscrash, phase, hapg, framex,
autoaccepter, brontok, tcpz, iparmor, boinberg, errno, dualdl, zapchast,
assasin, lexis, hallenger, fearlesswebdownloader,
setcrack, bored, keykiller, gifttroj, dinkdink, 7thspeer, heliosbinder, sdb,
delsha, smokedown, nohmelui, frame4, julius_1904,
mokser, regonid, evilsock, mk3, gotorm, mssctrojan, laboes, locotout,
sinipnotifier, bessal, cefyns, compain, tatch, absturz,
duduaccelerator, hacknuke, appserv, fiven, keysend, crypt, lipgame, loverspy
, erone, youzer, gametea, optixkill, arveye, streik,
anset, pussot, lestvoz, lenc, papi, feardor, guagua, musanub, agenthide,
cotan, smeet, smees, ginra_3570, xport, reda, zalivator,
i13, kirpich, phicik, jupir, kickin, oscarbot, eurasia, winnti, microjoin,
melare, godog, syspro, sanctuary, panddos, zoek, spion,
msole32, jukbot, maddis, outa, peanut, deepthroat, ouftap, ebtreporter,
treple, critical, dynamer!dtc, sniperne_2_2,

oneclicknetsearch, prondir, laqma, antiav, proctor, resdoc, acnthunter, dskeylogger, threxmond, coolvidoor, formatcy, ay, ripinip, krate, theals, instahp, subsearch, knight, shoco, holistyc, subwoofer, wanirc, android@mm, lmir, gatsorm, satray, crow, werther, ramnit, exehooker, telsa, rnaap, taskexpl, vmware, vcrypt, grador, fakedemic, slipro, wcgen, donut, pipes, piper, panoil, rafdy, myspy, subla, rjump, doomhunter, farfli, ahker, wexd, vigvam, sinn, julus_2702, tantixbot, impossiblekeylogger, sinf, mkar, safer, wadolin, remotexs, sharker, cero, skoob, sersam, icondance, lukicsel, delinf, relsy, nabfeign, chiton, qwhack, uboot, viahack, dryptdru, lamzan, zaoao, realvnc, fragled, darkomen, hakey, multsarch, injectxin, silentium, nuclrata, chernich, dupex, cunario, imagekiller, iis, wagroc, campsafe, aolpunttek, datheens, freeline, numrok, sekcja, ecoly, xsinct, tunnel, winreboot, procin, prodvin, pfexp, unreal, udp_fluck, donghe, biosada, winfavorites, realserver, funsoul, monstres, bahisho, diablocheat, matyas, zwqq, sigger, contempt, pogard, servudoor, mix, afn, lama, keenval, lame, iblisdoor, medianav, blonky, lamo, offcrypt, demekaf, myhar, lamt, stocc, multibinder, nicehello, liewar, mintop, vangeridze, klap, shter, lesword, danton, romario, extcreator, ahwsvr, fraz, themexp, opwin, nepinseft, autoworm, kuang, svcfake, wibimo, netshadow, mafod!rts, likun, zekneol, milol, superkod, arcanum, slugin, yumud, busan, dagger, ceeinject, craztone, galaxy, jiakong, alpoor, ld, labox, aniema, malbau, terzac, webex, ls, spyagent, pangu, shagbot, paema, lunatik, dat, progenic, kindal, gabpath, das, helldoor, dax, emotion, nagderr, icqflood, advoseven, fearmusk, xhack, sin, nuyap, video@mm, lespy, warner, srizbi, gpcode, igloo_1_5, winker, hilgild!gen, rex, kaogel, mailsender, dsnx, jinmozhe, fixerror, cws, dewin, smallagent, qbomber, delfer, ms02039, potomac, rem, highway_8192, drivethebus, cwg, rotor, gascript, vipgsm, fakenuker, careta, prolin, ocibt, vatack, bwg, kpsule, austral, druogna, yars, serinst, mortag, killapp, scudagent, alcalup, pgn, jitux, bodime, keylopws, pupper, spreter, bromead, ghostkeylogger, messah, audhi, scano, monkey, 2_0, winternights, vulgar, utrick, antimane, cabik, udp, ms06040, meseft, banuris, forethought, vostroy, kilonce, dabber, orzzo, rbot, sysag, loven, macker, podcite, carpediem, lotusoft, bnet, splashtrojan, tinykeylogger, qspy, fiwd, darknova, israz, hadefix, dickler, tetrikra, winmain, vbcrypt, jjoiner, poffer, hoavelu, ronathim, antimcafee, firefly, scranor, miscer, wnuker5, serot, sise, julius_1929, sillywr, keyhunter, sfc, jaan, keriop, surnova, porndial, netjoe, ceatrg, osapex, sentral, zfn, acccontrol, nsag, nags, upsodos, hamlet, vb@p2p, remoteagent, aolbill, maniador, gwgirl, grabber, satan, killmf, bezil, zayan, pitfall, sysclock, fragglerock, syst, ftapp, systemdebug, funflash, moriogu, vovan, rshell, gemax, secret_service, varun, anakha, refree, wmpatch, phidagem, baseaddress, jane2, ziconarch, readserv, karakum, furootkit, osirdoor, mirbot, womcodi, choochie, apoc, duptwux, complan, famus, pushbot, exploiter, apop, webprefix, formatall, eclipse_8192, hotmai, messenpass, janet, winfixer, pmail, naith, jerror, intpack, winsmurf, gosusub, chiviper, pahati, klone, nijrecy, yoksearch, jauxeer, sexer, cedocer, bustem, ws2k, faketask, pitfall!armlock, neededware, webcracker, denit, denis, fraudload, mafia, merinos, jini, ipgetter, af, nb, ai, pleted, gaze, msnhack, anomaly, nullsin, fulamer, aj, netbus_2_1, xmas2000mailbomber, fynloski, poter, vdgrop, hakuti, concon, batchlabs, homer, splitu, mutarol, kirbster, giza, ataka, psyme, iehack, msgghost, stopoffice, girlfriend, imvocet, rootkit, ultimspy, bo2k_stcpio, proxima, rapido, doftenlo, plexis, bcblade, unifyda, infex, racvacs, quazex, nytra, fabi, chimera_1542, jzone, clentil, dapato, fakems, bokill, banmailo, khaos, methoaf, tudasuda, shopathome, stepar!epo, fogels, rootcip, sett, uosproy, getdialer, flyvb, snoopy, hybrid, regprend, finodes, keystealer, deltad, pestdoor, darkmil!epo, adt, rimpa, ravex, inoot, bdirect, bladabindi, rebate, sabrow, ath0, sanctionedmedia, scrimp, raven, winspg, fakeaim, unicodeexploit, fys, aolbucop, cucum, realis, gspy, nimrod, hokeydaph, hybris, qwin, kulsibot, atho, chupik, chica, netcontroller,

exitwindows, ackenbel, chico, maliciousmsilloaderkazy,
tetas, webcamnow, diario, canbis, tony, otran, torxy, cefamon, tms, pasorot,
fakedos, corpse, desos, msdun, depthcharge, serpa,
noise, vataf, lastword, beebone, liveagentuvi, hosp, whacky, valfroc, expiro
, retribution, mypor, justin, mlfree, whoran, addendum,
usinec, vizim, mabutu, blobus, wingatekill, spyderweb, minigaway, bradop,
banito, delalot, cskey, testsniff, rserver,
onestepsearch, procaddress, mescalin, smil, lambot, maza, dontovo, prizzy,
rsbg, aimframe, glih, sequel, moonpie, foncsir,
aimscrname, skob, sperolz, freegate, bum, hacty, braidupdate, bubbel,
snakdos, snakdor, rc8, pospa, levil, flip, prettypark,
keyjoke, livuto, wimkilde, flib, mat_it, spyaxe, whisper, litmus!tool, epex,
baop, upnp, piw, availmetre, triplix, legendmir,
sepuf, tofger, infoaxe, permon, gamania, desktoppuzzle, sabia@m, namice,
ascreen, lalexa, gaewe, frubee, gigaspy, ikmet, rpclsass,
mixter, mokaksu, haxortool, lolit, dedipros, tibs, nwu, vck, comrerop, hatu,
blador, dipwit, hata, ducracker, delanab, liech,
patcher, pizload, moseneb, lamerx, patched, sigatariusspy, dalloc, flash3,
wideman_8225, darkscan, rarnmel, eva_4096, sexspeed,
icecast, mejdho, hermon, msnrat, ginseng, potentialdownloader, insaim, senik
, couponruc, meliks, dawin, secworm, school, gotit,
gichtyicqflood, parrot, gluer, sanpec, lovgate!rfn, xxxtoolbar, rcash, zonit
, harrier, liac, chadem, reklam, depi, nail, alogics,
dundun, naid, moniter, monator, wanex, enjof, zlxget, hidd, leetbot, kut,
poison, zasil, siboco, steplar, pswmon, bandeja, xfl,
zdown, imaler, starex, studio, sheker, gift!binder, kokodoor, sillywar,
digits, blackdream, philix, mantis, lastdoor, geycript,
moega, gimly, rejok_940, mudil, lamchi, wimain, wimail, qqnof, 1_01, idicaf,
proxydor, gotoone, bluback, skidlo, pernefed,
jokedoor, meenvi, idyll, perfcl, dllinjector, bonsws, starpack, aost,
phreaker, m2, locksky, gravad, phrostic, zhangpo, adonai,
mocmex, boomer, hackerheaven, passma, getdial, epident, crazytyping, fontra,
haharin, mj, wador, takill, malstart, neop!gmb,
nahibana, bomberman, netbuie, trillianrape, bigjack, mayhem, orion, ontonphu
, laocoon, groundbiz, santana, wmfap, consper, winnuke,
raampjes, xobnur, wcrat, smallfun, lingosky, vkhost, delexe, anaki, pokey,
ipamor, theug, beef_2110, zynet, losfondup, troodon,
fryeim, crysedru, mediaticketscdt, aimbot, brainwiper, relog, midap,
cabronator, rastax, rispere, enviar, arnger, zirgt, linong,
namaz, cefet, zayaho, phrovon, tradehack, iischinansl, stealthproxy, robodor
, runnelot, epaemy, mypics, enculator, vxskey, tfwb,
remserv, sendkey, sgww_one, demspy, dialxlite, gexin, zcrew, awfull, lamiun,
wincontrol, iowa, paramis, neysid, sojfuse, e404,
paqtoolkeylogger, xyligan, neworld, xpcspypro, sexip, sientok, spysend, alex
, aler, paramil, eliterab, noskey, caprobad, detroie,
adduser, kergez, mocdoha, pegajoiner, darkmoon, idyl, warkosh, lnkiebes,
fatroj, bodon, depress, regback, gibbon, vwealer,
wowstealer, adrevmedia, trogbot, mayday, lafon, siapag, gop, isq_sniffer,
cureicq, inor, gammes, aspam, busconquerer, gadugadu,
jeshex, evil, milcan, ambler, fakerean, lablank, filtek, explbkd, dumbcomp,
tooser, soda, totopf, webalta, pcik, puce, hamweq,
vaxlorne, oodoov, meetthelamer, zusha, sober, vihda, undernet, fantast,
remcom, vburses, sjkr, tinydl, arpkill, top, delfhost,
advancedkeylogger, zombaque, smokodoor, qupdate, incub, rortiem, gaobot,
ancev_512, nordon, spamchn, injecter, netsky, regswap,
servu, ennumi, sensode, sivis, petus, tapazom, sysdeb, diabolo, flame,
karnpir, petun, gaboc, yafive, izram, rsm, savnut, wowbar,
gaobot!config, pnimop, piorio, raw, rat, valvoline, ras, azaco_8192,
mediamotor, netkiller, killwifip, floodnet, zcom, cracksearch,
snow, pelotas, forput!rts, contact, derek, ultimaterat, phinit, jetil,
meteorshell, tibia, vbstat, flashtron, moldyow, coinminer,
gop@mm, glox, hocamin, dollarrevenue, bse, setha, peers, rwins, flor, metal,
yanz, dumarin, enterprise, radix, randon, pwdump,
mailspam_vb, grexon, pops, glacid, disketka, enumero, lamebot, apost, zomzap
, comsirig, ledap, neodurkbomber, institution,
allinone, relator, debploit, rimod, fepgul, kake, gespo, dupate_4180,
elfapault, andras, xpentert, johntheripper, niklaus,

devil_1_3, xzone, gogo, prosiak, svoy, guptachar, geef, syser, cutwail, ksare, ghost!binder, antiavg, cometsystems, apsob, paroc, semapi, netslayer, qhost, freesould, meatyip, newfuture, texagon, evul, nuj, formador, snova, felino, sniperne, sindoor, remscan, emeres, akcom, mua, lovemail, timpin, greenday, zenmaster, newk, pobtiz, adsearch, sunfo, mysearchpage, huntox, surferbar, mulod, gamec, zackfoo, ganda, ransirac, gamer, jubu, yellsob, emerleox, gamey, dvbbs, etymo, cekar, carberp, starline, sypak, aback, tmsd, yohoo, findstar, beer, fakesysdef, msntoolstrojan, olux, sprite, netdevil, blackhole, recodat, rinim, flobo, rusmgr, theripper, pinger, windup, remotecommands, fonob, cracker, bitcon, aphexlace, deept, norachs, wmremotekeylogger, floppymadness, emurbo, mader, mades, elfdown, cazdoor, tress, xilon, pwrhack, fanet, daekom, delikon, chebri, webind, niko, spreder, frenzy_2k, sacanph, bigdipper, cordmix, misery, theflu, crabox, hipak, nongmin, cycle, lopown, icmibs, omnitex, charlie, spelac, 18431, bustah, xposure, flator, zdownloader, keyloggergeneral, freity, sefnit, yspy, spytector, reversetrojan, deathpack, ntillusion, trasher, vbsencoder, mincli, ceysix, connapts, scafros, keyser, nistio, sealer, roacz, calcapov, sapaq, infecdoor, purstiu, pixnet, ovoxual, suixin, roach, donab, rufis, apart, bombim, streamto, ditto, gift, renamer, gatehell, kromber, geradorde, spawnt, peevet, yahmali, mescan, diskflood, amor, klexe, acercadeembozator, xicq, amprye, nosrawec, butool, sponge, adverbot, yabector, tolone, bandok, minebicoin, specrem, umex, yuner, lwasuzo, toprot, rhino, queen, ics, hydroleak, nobof, cnk, matrix_817, core, janeboot, dugenpal, corn, kelvibot, tagasaurus, wykcores, deleter, systray, centar, abetterinternet, choke, greenscreen, wolyx, genocide, belnow, spybox, spybot, hdek, ctx!enc, feri, synkal, ferq, dwonk, pecdav, hiddoor, desaldaf, mapanna, iisibk, kotwir, ideop, gentad, capwin, mydons, skintrim, iishack, kvex, hucline, slixt, solaf, icup, manex, heak, hotmailhacker, aimven, wingatecrash, ghostar, vibam, csrsshijack, sohanad, suph, rephlex, paging, warlor, olinger, fakerecy, duella, pereb, maryl, clariagain, smtptest, blockflip, icqsteal, darktima, trix, juntos, zombsmalltrojan, aolpass, sbome, sporke, tix, kipis, neverdie!epo, webot, dozmot, tip, autoupder, estufer, toal, nugache, ramnit!remnants, phoenix, toad, smoothie_1_1, dishigy, podjot, lsascan, axhuan, qqsendmess, vicenor, y3rat, perkesh, pipsek, nalki, moorest, usbwatch, frethog, malog, winroot, fpoman, imsoft, shypan, orbond, rod, tibser, sgww, trillima, tubrefum, nooler, pidief, doraah, buckhs, inverse, plimus, charge, chindoor, qqfun, redshad, advantage, tubby, illicq, schedan, kraze, nukemsn, ezkilla, haptk, cool, divxupdater, pugeju, darksma, bamital, yektel, refogkeylogger, brother, levex, enot, pimaf, lohav, posta, pmt, wildmedia, invader, porn, pmx, obsolete, wudfok, sockets, farnaz, sadhound, spetcrum, illmob, dusbunn, postman, ogimant, cecile, hscr, amighelo, alemod, globalkiller, del, opaserv, supcount, winsniffer, bumdoc, nagil, rseries, udepo, pgpme, sms_sharft, wisvereq, puper, amelg, ellikic, jord, colorbug, metrion, nucscan_sabine, autosipoc, banpaes, alvgus, jort, hcbkd, celofot, weird, henky, hiton, baxdrew, bytesv, pixo, dudra, abspics, 404search, tcpspeed, schaden, vivia, konik, vivic, autoec, smspager, reversebackdoor, konix, movingmouse, prex, migls, cocaine, chimpa, winock, pres, gotem, promail, urausy, mainline, casey, gemeindru, zaffi, kerlofost, oficla, remoteop, pramro, julius, ceel, valla, populf, perly, cinject, morto, lebugap, wintool, shatrix, stream, slenfblot, beoter, razac, silcer, aimspy, volage, baczback, fakemalard, sebak, seiblisima, spotcom, celebit, camking, darkportal, safemall, twospace, kespy, vatos, brunme, advkeylogger, crasher, yanot, levi, lohk, bebloh, icqsms, kervar, dxmsmtptrojan, xoro, projectnext, liquid, apropo, inform, berok, gych, julus, fallout, orter, canary, qingdl, trillmsn, fakeqq, liucale, evibo, funpop, munstre, xyproj, aluroot, sinodo, bloropac, apem, windowssyncroad, bouffe, heretic, gayol, arcadeweb, boomie, nomada, pagun, term, cjdra, vagen, daptdei, avone, psyokym, nbar, warspy, linra, pluzoks, codefish, can20020649, gaopro, persian, winshell, sclog, najort, twkcpl, eesdns,

factory, bube, undertaker_5036, adcliker, setfic, geweb,
siveras, httpdos, ruce, simer, obfame, satania, dilya, neurevt, hobble, eetu
, dispatcher, delude, lasiaf, merong, ducktoy, subzero,
uprootkit, sukill, gip, zombie!epo, kneel, lockscreen, ssiwg, comine,
malpayo, wincrash, paiteyello, sobetr, bedobot, aolpic,
voterai, supermm, elkern, blackeagle, flizi, aplore, virtualroot, buddylinks
, adject, nathan_3792, uplogger, mazpayn, white,
homekeylogger, pircher, unruy, fakemsn, pronic, kkiller, avrkill, fakemsa,
ulubione, millremtrojan, daodan, jiang,
perfectkeylogger_147, yahoxer, unjuz, wide, icogon, nathan, ritart, blueice,
mailspam_delf, lanaftp, sharft, spolash, rafta,
hungryhands, octamus, cards, trojandelf, imagethief, campla, reni, bo2k!rfn,
svchost, therat, m2trojan, ant, renocide, killhelper,
rodvir, subseven!tool, klik, comson, perfloger, hoygunver, redjunk, gawime,
realgz, fmail, darkpus, backattack, boiling, iistorm,
asper, fakeaol, mediyes, porex, sensive, sohs, later, sddownloader, hungry,
yourde, sohu, pavsee, veedna, anker, inhoo, jiospy,
armaged_2, satanzcrewnotifier, vboxador, lalus, tron, cheat, sobit,
dlstwoyle, spaces, socnet, hack, sobig, neter, lalul, hustler,
elidex, kirasha, landis, skrin, desex, dupator, crime, desec, kesk, paduds,
litar, recspa, fickle, mypic, getin, stata, fratele,
sst, inetspy, kbrdspy, minild, diskfiller, pexnod, zdemon, dyf, buptenda,
preald, afgan, dwolf, sbot, tofumanics, egghead,
imsproad, joker, kaiserdown, barbarie, headout, blaver, gypet, dinoxi, afgar
, mewey, xdel, rmt, boupke, nagyo, rmc, dtrv, mumawow,
nuctibet, makedirs, sms_attacker, infiltrat, killxp, outside, puntol,
autohack, darsing, fabi_15930, oocooc, fearless, atomic2,
boaxxe, prosiak_070b, coma, jumptrojan, can20030466, hopalon, darif,
drivalon, starfield, lorie, anonymailer, freeweb, smarpiyasa,
jammer, haltura, inspir, agramat, awjer, insephor, zzsoft, crsgate, harwig,
pof, pod, bindweb, necast, skyrat, gichtychatflood,
noknok, qqmima, icqnotify, hvlrat_5312, lindose_2132, orgame, aimpwf, pilon,
witer, messer, waardaat, tiger, adultlinksqbar,
avkill, facepass, thefinger, kaos, spirit, tamin, amend, krass, spymail,
kobot, hilin, antivirusxp, aimjaker, privacycenter,
yydoor, remotekeylogger, mierun, alylum, alphabet, fender, bowl, enviserv,
beadmin, naparb, spadeace, dorn, tsupdate, iisdos,
libido, enerlam, vbclone, loadfox, submariner, easycrack, incommander,
kelvir, gunbound, cahyna, vedex, fobot, nofear, iyeclore,
ranck, srvcp, icqmouse, searchmeup, sinmis, taxifolia, ramgad, zupshero,
dodgemon, pifbit, theefdl, amadya, portcon, lybdoor,
irckill, gavir, clop, kilo, elivoco, killzone, uncap, kidterror, invictus,
petrolin, phdet, hine, ycracker, blol, hino, searchhelp,
banbra, killr, pontoeb, kongrid, scode, kblup, winhalt, infexor, warmup,
mtorrent, snorm, vbinject, cfour, elkmil, krotten,
hemtray, batdetour, redhors, cratpro, msworld, mauthy, freez, netpunk, xinf,
xine, epip@m, die3nt, mook, reccaz, dodoor, consik,
monatortrojan, myparty, svint, flopico, jakuz, zmorph, nythug, toledorz,
runfile, meibu, falseqq, tepv, firebird, titog, lohack,
pointup, jeem, stang, fcrypter, kittridge, blortios, paltus, amber, jaedus,
toren, fireruk, pakernat, peguese, gara, smvss,
bigshot, warfair, hooon, ystl, abotus, shbot, dooxud, budu, netbus!ripper,
house, rirlged, checkinet, yah, gibe, relic, eraser,
frontover, yopper, aolcoprono, aldax, relis, cosiam, vulcano, aixftpd, aname
, mydown, godev, netol, feuer, windowlivepot,
clearsearch, blazgel, hauntpc, cassidy, netop, eniac, rigtoy, syhotran,
hyflid, appbundler, tagrecall, blakan, oneclicknets,
teppto, obvesa, porkodio, howjey, mike, faltbang, felic, rvp, miptrojan,
energyfactor, netmagik, winskiller, ircbot, idiot, fipeg,
rawim, weavun, heathen, hellfire, alco, 4096, fatalex, weiameia, telhack,
novelce, chepdu, finalx, fakenap, poson, menti, kof2002,
exquein, screenroses, teleb, folstart, eyestye, pcspy, duon, demiurg, kyzbot
, wecykler, crobftp, newacc, sepro, soduc, snid,
optixdll, tontark, merkur, codword, stardrop, gentee, pccontrol, kslogger,
sisits!rts, crugup, favoriteman, cocoazul, xorpix,
weird_10240, aglvus, awful, remrochor, ticker, elpro, caesium, zyaku, retroy
, bablo, netgrisch, porga, serman, mudrop, taptrap,

tamenoc, webdav, sanjicom, nucledor, rybot, microwinsearch, syslog, haan,
infober, frastron, nba, pvbswg, acidoor, babylonia,
smeagol, niden, pulpit, phopaiz, safemail, nehata, wkill, proteboy, demes,
octoark, aimevildoer, udp_neonix, mjoiner, transponder,
oicqdover, pochi, xema, demig, snur, salitre, hoaveldoor, warespy, ebeg,
turkahn, lightmoon, smal, thamcower, vbswgbased, evolmi,
hermes, govdi, saddamme, iroffer, mellon, yaha, wowsteal, assill, pcinvader,
antires, verano, dadobra, sillywr_136, lohet, sysmono,
hatevba, bumaf, delwin, smbdie, obvod, crazynet, hayqu, randir, fireming,
cqma, balyz, fibermil_3137, xoxer, vlight, roomdestroyer,
ledos, norinc, strpasseal, autinject, antipc, olmarik, zhang, multijoiner,
moopidoop, smaira, hoaxer, sabine, nums, bumper,
pcagent, mickdo, apsiv, hscan, rowmuny, kuterativ, susan, ecopic, henod,
festi, cryptorstub, freescratchandwin, directblaster,
shorm, disabler, coremhead!gmb, quopax, peerfit, qlod, seveneleven, 50000,
toauta!rfn, chcb, fakespyguard, runar, diakey, tafix,
argentino, chatflood, sysrer, wuloit, erdin, runae, porsmi, netcontrol,
iisiiscrack, vinger, shydood, kext, hunter, clickme,
botnet, can20030112, wogue, losabel, uniskit, gamov, fooldad, valha, mahs,
zotob, tiltee, trobobo, truiq, gosocks, pawned, heidex,
sandesa, undertake, fakeformat, eldycow, donk, lola, vkont, bithaw, pfinet,
iesearchtoolbar, kegotip, ninunarch, crazybilets,
starms, rshots, windaus, cslam, procexe, benjamin, radix_402, clort,
ginapass, spybuddy, pennytools, goldfake, miniglitch, hdkill,
kexqoud, msnsmall, virton, feidin, whog_878, wlf, dinf, dina, pinguide,
urelas, gtkftpd, asylum_web, dialupas, dfm, subcra,
kameral, ashlt, swarm, kingsolaris, storm, molesto, kedad, lemir, aspcode,
broser, donalddick, gvuz, aa2, nullpos, sycomder,
blondie, miewer, avexp, fintas, iespy, rotarran, king, kino, cavitate, gobot
, bohu, dani, remod, viretuam, eps_130, bonuscash,
aimb, rugo, maroto, romelp, argech, gobind, efx, temir, nethief, zimenok,
zegost!rfn, zawex, huhk, nachhat, cocar, toxo, copia,
regate, webath, conspy, comquab, adder, exes, assalaut, prestige, gilp,
tirtas, zombprat, yaher, duiskbot, barbie, mateusxxx,
lashplay, tsgrinder, scofted, wadnock, hkit, vcladru, mifeng, rtb, winboot,
umod, advancedhack, holycrypt, barten, superad,
suckspro, traux, blarul, woodelf, can20030349, trafficadvance, bfs, kyga,
ringbeam, moses, ied, winkill, atendo, rapita, hiv,
alicia, nefutur, ramdile, paglst, naukatr, dormer, wurgan, rob, xalnaga,
egapel, bladi!rts, ofreayo, discoball, barisot, cipvon,
serveme, dedream, capiruf, dump, huwankata, delall, esmeralda, azha,
robinkiller, defense, bare, kweeni, arg, susanin, rebo,
msncrackstore, quickbrowser, histboader, mvold, rslocal, mslagent, larvagen,
fohoma, vidcash, euniverse, wincheck, ieiex, druvops,
oderoor, aimrealmccoy, ptrojan, msnlogthief, urcs, bazofia, faybox,
singaraja, viled, kapod, duaspi, proix, litmusii, cowor,
dataspy, annil, yenik, btngdoor, webin, sweet, surfsidekick, ravs, webdor,
sakao, redrosdru, tgf, tiul, ranbyus, ps, icabdi,
deftcode, swc, swz, netpocalipse, downhoax, cleaman, whog, pm, knewk,
flystudio, feljina, ukash, simbolos, rrestart, noxjasm,
yahlover, wzbrute, ztrypz, taek, crackerbox, mewii, lamnet, frozen, redhack,
pereban, hropac, phantu, sennaonemaker, adpower,
packetstorm, shutdowner, halen_2596, kilfno, rit, interlac, rip, grenp,
safbot, zebroxy, sirefef, huigezi, ric, banselso,
prast!rts, qaz, minix, bregol, lechiket, minit, minkir, guangwaigirl,
mpsundetector, opasoft, fastore, remoteexploit, lethic!rfn,
keyboardthief, easysearchbar, noodle, deanom, rat_ii, onerin, darksky, miam,
bleepingmouse, sycra, almaeda, icenuker, russkill,
nevereg, passripperpro, becna, perrun, poltergeist, blaxe, winhelp, redro,
pepatch, drdos, acnet, eqtonapt, bedrill, vscrt,
icqfuer, mygly, rofmohe, lovit, winupdt, vosfigh, stawin, faledel, yisou,
rotate, hantaner, barlf, ositdoor, vbdrive, liame,
inservice, gypthoy, lamespy, partriot, netax, necrole, louda, fernando,
djupd, genteekiller, evilhttp, popspy, proy, bdurl,
kloharn, keyjack, prop, proj, rapnod, levona, iisssl, darto, sheng, wotron,
zombie, ahbomb, rudeflate, kraddare, dktoybox,
starcross, parakid, dicsor, doneltart, onlinelogger, paroxysm, hatred,

cabreck, detarmal, smitdof, yupfil, hulstor, assame, momma,
leymo, elitekeylogger, delfsnif, passoner, xvgl, lovgate, remoter, rebbew,
fast, spaces_1445, daymay, henky_1448,
magicpsyahoo!messenger, simbot, dados, bankpatch, winsessionlogger, auric,
genasom, flita, eter_7168, delta, juntador, cranb,
nodfu, icomman_1_4, doomran, duksten, kingos, habar, pirt, svenx, spycredit,
freak, skyper, chiboy, firepass, lywer, rally,
meredrop!rts, rainbow, evilbot, vesotup, chyup, wantvi, eurosol, hacdef,
lovecamel, cacfu, thutani, skatanbot, kamaj, kuoog,
cupsop, novabula, daboom, deadbolt, vorbeld, kaiten, dalt, bonding, sknet,
fendu, clickit, fregdll, istall, gosup, teta, claretore,
scrolo, poinback, enfal, 2search, gant, calacreo, alien, sploiter, kanav,
cargao, mitglider, fireox, sparsay, spector, sms_stefan,
tofam!rts, veslorn, strigy, brid, breach, tecvil, sulpex, unsored, spool,
vestic, sksocket, clucsplic, lyusane, tyqui, leodon,
mirat, noter, sovbot, cih_816, tefuss, syfi, cetis, skin98, goldeneye,
myagent, vecnoit, iflar, shuq@mm, dkangel, acint,
easyanonmail, glecia, gotalk, acid, reppop, polip!dam, toxaic, nilob, vybab,
kriz, daromec, difeqs, ethersh, rosya, compan,
traceboy, kazaaspreader, ilmx, antilamerlight, vbmest, sexo, insultmedia,
miser, kanyak, dominador, xorph, netlist, mettemar,
winwebsec, datasneak, honeypot, crumpet, msnnick, adultchk, marburg_8590,
intelliadvert, alcaul@mm, wk, serpro, crypter, alobtoe,
gsmcrack, wesoten, fed, popwin, lottery, iced, frog, stealthspy, icer, inex,
yabo, dreamad, nitvea, urat, padmer, korgo, wargam,
keypanic, url2dword, goaway, suton, vuder, rabbiz, antinuke, fodvorus,
favadd, zirit, duster, aolpok, firewar, lamewin, score_3072,
redbinder, muldrop, executor, aimvision, caphaw, morgoth, portterminator,
taz, kagra, keatep, wingate, picsys, netsec, cyberspy,
insteax, kerspin, spydoor, resurel, zues, panik, bantool, theefle, protoride
, pcclient, sytro, lsass, netcontr, outbrowse, rolog,
killav!rfn, thorin, hangping, waizo, paukor, pirathack, ragedoor, usnet,
sibleep, ranonemail, pboexploit, jolic, clustinex, vtub,
demmo, gara_888, getfund, quake, potar, screengrab, eliz, netterrorist,
midgare, sinique, haktek, sydo, imort, dermon, ccinject,
abndog, horsamaz, calelk, listas, costrat, antirar, aquadoor, crash, nonnel,
khesanh, deves, wintecor, aimrat, loror, andriod,
nopti, flee, mmci, webdownloader, newdotnet, mirbaby, trab, powerfull, haed,
trah, qqdrop, hangame, billy, logonui, teclass, champ,
fourta, randex, cherat_1933, fatcon, anxiety_1358, apophis, cotfuser,
forcedentry, deser, minodacek, nesgohn, subivix, msnbigbot,
rutispud, vcr, whirep, ninjaspy, rff, upadmin, farli, molly, riprox, control
, vai, karachun, zzmm, yorc, kober, testsp, raber,
lanagent, mbdis, caid, gforce, wininfo, killsfiles, analox, infdll, xuma,
iwanywhere, eiunpach, glocker, densmail, likseput,
begman, lodrypt, sankei, pingbed, weko, deskwizz, lapre, mrinfo, priority,
boid, locksky@mm, c2lop, cadux, shell, homutex, julk,
transistor, gibon, slides, lemerul, flymux, berstray, eleye, netghost,
goblin, doomer, pirpi, shelp, wofopey, reposin, taesb,
ghcif, nihilit, duload, koretek, can20030003, nbpass, seb, udp_shockwave,
rcserv, alerter, elfnet, exploitdoor, benny, mogul,
ruledor, alexa, roopirs, bizex, delfdrop, eter, rous, lopy, servidor, figbox
, drateam, teaper, newbiero, malev, exebinder, baiso,
illnotif, jannetremotekeylogger, fddos, mong, otwycal, masteseq, mutter,
delvi, clodpuntor, botanvec, lames, lee, odsy, nexz,
murlo, samnuk, looked, wpepro, mex, infectus, capsfin, dacryptic, logex,
xploitgen, gain, irapture, taris, familywatcher, logrelaz,
fanop, lassab, olafre, skudex, muvdos, perlay, slenping, subseven!finder,
katux, killall, lemmy, micron, melp, hostisver, helios,
aolmgk, mmtask, bluwin, unchained, screenlog, systemsurveillance, cubi,
vexral, cogebot, vaite, cutout, sergnt, bumat!rts,
toga!rfn, shodabot, leviathan, manyw, manyx, blackmonay, timegluk, devil5,
atmader, killavlo, biweaver, freebird, urlsnoop, hdfill,
smarcar, melkosoft, winfig, kefy, oposum, muska, oblibion, crazzynet, xrat,
mads, p2e, beaugrit, yvoice, exal, lacrec, flatei, job,
voidozer, hidrun, coced_235c, gensteal, swift, hobo, wenga, kodorjan,
fireanvil, clogger, netav, wudoo, bizdup, longvi, heoyon,

celebite, waledac, rpv, lashtorm, enegi, kidlogger, bbn, bbe, compor, ligsetrac, siteno, mailpassview, safesurfing, dssdoor, cx2, htool, blitznuker, bugexploit, epldr, wabbin, ponfoy, spysender, wuzak, dialplatform, sheepgoat, kimex, p2load, vimes, kimes, habudru, whirlpool, trasbind, sinowal, katherdoor, crashsystem, gisafloader, xlog, dbomber, ghostra, puppetzombie, ziyazo, evol, customie, halint, icqnewq3, giri_4937, eyeveg, lamost, unite, siluhdur, sonitro, hisbucken, tometa, websearch, wlog, itsproc, diego, shockmark, yoof, beast, frusion, livelist, lsky, dlrhifrem, cdilla, rochap, arsd, hala, undertaker, barum, hotmailhack, g_spot, gutted, rsbot, msnnightmare, rocket, pexmor, camec, xmiler, wren, floid, rathead, skrat, dtr_1_6, krni386, getad, panther, genome, smith, byweird, trucha, ska, haless, gupboot, frethem, lineage, porlist!rts, sircam, loah, webrecommended, darkside, knox, embhit, udp_mrudp, feral, verital, seapig, loser, dros, vortex, ferat, dsocks4, stonari, amores, reopt, turtuk, scareg, kullan, wenru, necmko, cradolep, parite, gohotlisttrojan, opanki, japedoor, prefsap, xdialer, iiscrack, fakedoor, lear, disager, tibdef, bublik, rohu, tedroo, sicirc, bo!beeone, tesefo, sidetab, warin, sillywr_152, kalash, mite, pesdear, remotespy, kbldown, povgon, regger, comneop!gmb, closemouse, vinject, aybo, twinny_16384, shinop, swapmouse, jeefo, hamtacker, jtram, ixas, delemon, cocker, promise, petlil@mm, disablemouse, rhapsody, brabot, nuke, begemot, remoper, dihallo, followme, meciv, ireul, hilat, giftbinder, shang, cloverplus, conbatib, xconsole, belod, temu, emold, gspot, recory, cakl, zsyangel, bysmd, aolwinspy, meduna, cmon, regdll, ailis, buffy, icmprep, fyngen, mneah, sendfake, aphidma, nilage, ih_infector, demonize, nukensnuke, apher!httpd, sassrye, debllama, drowor, boom, axam, badco, floodrefu, ghostdog, tapitroj, puram, vbpnt, juny, shero, saeeka, ranky, zytric, xenor, ddoag, inactivedesktop, mediapass, xenon, pibus, xenof

E.3 Architecture Distribution in PE headers

Samples	Architecture
231445	PE32 executable (GUI) Intel 80386, for MS Windows
57012	PE32 executable (DLL) (GUI) Intel 80386, for MS Windows
50608	PE32 executable (GUI) Intel 80386, for MS Windows, UPX compressed
11009	PE32 executable (GUI) Intel 80386, for MS Windows, Nullsoft Installer self—extracting archive
9349	PE32 executable (GUI) Intel 80386, for MS Windows, PECompact2 compressed
8668	PE32 executable (DLL) (GUI) Intel 80386, for MS Windows, UPX compressed
8507	PE32 executable (native) Intel 80386, for MS Windows
8117	PE32 executable (GUI) Intel 80386 (stripped to external PDB), for MS Windows
5004	PE32 executable (console) Intel 80386, for MS Windows
4758	PE32 executable (GUI) Intel 80386 Mono/.Net assembly, for MS Windows
1604	PE32 executable (DLL) (GUI) Intel 80386, for MS Windows, PECompact2 compressed
1314	PE32 executable (DLL) (console) Intel 80386, for MS Windows
1134	PE32 executable (GUI) Intel 80386, for MS Windows, Petite compressed
1061	PE32 executable (GUI) Intel 80386 (stripped to external PDB), for MS Windows, UPX compressed
825	PE32 executable (console) Intel 80386 (stripped to external PDB), for MS Windows
748	PE32+ executable (GUI) x86—64, for MS Windows
743	PE32 executable (DLL) (native) Intel 80386, for MS Windows
733	PE32 executable (GUI) Intel 80386 (stripped to external PDB), for MS Windows, Nullsoft Installer self—extracting archive
625	PE32 executable (DLL) Intel 80386, for MS Windows
547	PE32 executable (DLL) (GUI) Intel 80386 (stripped to external PDB), for MS Windows
520	PE32 executable (console) Intel 80386, for MS Windows, UPX compressed
431	PE32 executable (GUI) Intel 80386, for MS Windows, InnoSetup self—extracting archive
336	PE32 executable (GUI) Intel 80386, for MS Windows, UPX compressed, RAR self—extracting archive
220	PE32 executable (GUI) Intel 80386, for MS Windows, RAR self—extracting archive
202	PE32 executable (GUI) Intel 80386 (stripped to external PDB), for MS Windows, PECompact2 compressed
185	PE32 executable (DLL) (console) Intel 80386 Mono/.Net assembly, for MS Windows
178	PE32 executable (console) Intel 80386 Mono/.Net assembly, for MS Windows
169	PE32 executable (DLL) (console) Intel 80386 (stripped to external PDB), for MS Windows
141	PE32 executable (GUI) Intel 80386, for MS Windows, InstallShield self—extracting archive
139	PE32+ executable (native) x86—64, for MS Windows
124	PE32+ executable (console) x86—64, for MS Windows
116	PE32 executable (console) Intel 80386 (stripped to external PDB), for MS Windows, UPX compressed
115	PE32 executable (GUI) Intel 80386, for MS Windows, UPX compressed, Nullsoft Installer self—extracting archive
80	PE32 executable (console) Intel 80386, for MS Windows, PECompact2 compressed
70	PE32 executable (DLL) (GUI) Intel 80386, for MS Windows, Petite compressed
68	PE32 executable (GUI) Intel 80386, for MS Windows, MS CAB—Installer self—extracting archive
67	PE32 executable (DLL) (console) Intel 80386, for MS Windows, UPX compressed
62	PE32 executable (Unknown subsystem 0x0) Intel 80386, for MS Windows
61	PE32 executable (DLL) (GUI) Intel 80386 (stripped to external PDB), for MS Windows, UPX compressed
61	PE32 executable (GUI) Intel 80386 (stripped to external PDB) system file, for MS Windows, UPX compressed
41	PE32 executable (native) Intel 80386 (stripped to external PDB), for MS Windows
38	PE32 executable (GUI) Intel 80386, for MS Windows, UPX compressed, PECompact2 compressed
37	PE32 executable (GUI) Intel 80386 (stripped to external PDB) system file, for MS Windows
33	PE32 executable (GUI) Intel 80386 (stripped to external PDB), for MS Windows, MS CAB—Installer self—extracting archive
30	PE32 executable (GUI) Intel 80386, for MS Windows, AIN 2.x self—extracting archive
29	PE32 executable (DLL) (GUI) Intel 80386 Mono/.Net assembly, for MS Windows
25	PE32 executable (console) Intel 80386, for MS Windows, Petite compressed
20	PE32 executable (GUI) Intel 80386 system file, for MS Windows
17	PE32 executable (GUI) Intel 80386, for MS Windows, ZIP self—extracting archive (WinZip)
16	PE32+ executable (GUI) x86—64 Mono/.Net assembly, for MS Windows
14	NULL
14	PE32+ executable (DLL) (GUI) x86—64, for MS Windows

```
14      PE32+ executable (GUI) x86-64 (stripped to external PDB), for MS Windows
12      PE32 executable (GUI) Intel 80386, for MS Windows, WISE installer self-extracting archive
12      PE32 executable (GUI) Intel 80386 (stripped to external PDB) Mono/.Net assembly, for MS Windows
12      PE32 executable (GUI) Intel 80386 (stripped to external PDB), for MS Windows, Petite compressed
11      PE32 executable (console) Intel 80386 (stripped to external PDB), for MS Windows, PECompact2 compressed
11      PE32 executable (DLL) (GUI) Intel 80386, for MS Windows, AIN 2.x self-extracting archive
10      PE32 executable (DLL) (console) Intel 80386 (stripped to external PDB), for MS Windows, UPX compressed
10      PE32 executable (native) Intel 80386 Mono/.Net assembly, for MS Windows
9       PE32 executable (GUI) Intel 80386, for MS Windows, UPX compressed, InstallShield self-extracting archive
9       PE32 executable (GUI) Intel 80386 Mono/.Net assembly, for MS Windows, COFF
9       PE32 executable (DLL) Intel 80386, for MS Windows, PECompact2 compressed
7       PE32 executable (GUI) Intel 80386 system file, for MS Windows, UPX compressed
7       PE32 executable (GUI) Intel 80386, for MS Windows, Petite compressed, ACE self-extracting archive
6       PE32 executable (DLL) (console) Intel 80386, for MS Windows, PECompact2 compressed
5       PE32 executable (DLL) (GUI) Intel 80386, for MS Windows, AIN 1.x self-extracting archive
5       PE32 executable (DLL) (GUI) Intel 80386, for MS Windows, UPX compressed, PECompact2 compressed
5       PE32 executable (GUI) Intel 80386, for MS Windows, LHa self-extracting archive
5       PE32 executable (GUI) Intel 80386 system file, for MS Windows, PECompact2 compressed
5       PE32 executable (GUI) Intel 80386, for MS Windows, UPX compressed, InnoSetup self-extracting archive
5       PE32 executable (Unknown subsystem 0xeb02) Intel 80386, for MS Windows
4       PE32 executable (GUI) Intel 80386, for MS Windows, AIN 1.x self-extracting archive
4       PE32 executable (GUI) Unknown processor type 0x100, for MS Windows
4       PE32 executable (Windows CE) ARM, for MS Windows
4       PE32 executable (DLL) (GUI) Intel 80386 (stripped to external PDB) system file, for MS Windows, UPX compressed
4       PE32 executable (DLL) Intel 80386 Mono/.Net assembly, for MS Windows
4       PE32 executable (DLL) Intel 80386, for MS Windows, UPX compressed
3       PE32 executable (DLL) (native) Intel 80386 (stripped to external PDB), for MS Windows
3       PE32 executable (DLL) (console) Intel 80386 (stripped to external PDB), for MS Windows, Petite compressed
3       PE32 executable (DLL) (GUI) Intel 80386 (stripped to external PDB) system file, for MS Windows
2       directory
2       PE32 executable (DLL) Intel 80386 (stripped to external PDB), for MS Windows
2       PE32 executable (GUI) Intel 80386, for MS Windows, UPX compressed, RAR self-extracting archive, InstallShield self-
        extracting archive
2       PE32 executable (console) Intel 80386, for MS Windows, UPX compressed, RAR self-extracting archive
2       PE32+ executable (GUI) x86-64, for MS Windows, InstallShield self-extracting archive
2       PE32+ executable (console) x86-64 (stripped to external PDB), for MS Windows
2       PE32 executable (console) Intel 80386, 32rtm DOS extender
2       PE32+ executable (console) Intel Itanium, for MS Windows
1       PE32 executable (Unknown subsystem 0x9a) Intel 80386, for MS Windows
1       PE32 executable (DLL) (native) Intel 80386, for MS Windows, UPX compressed
1       PE32 executable (DLL) (GUI) Intel 80386, for MS Windows, UPX compressed, AIN 2.x self-extracting archive
1       ERROR: cannot open `/home/malwarelab/malware_collection/windows/PE/03659B39969AC5191B08C23256D755BB' (No such file or
        directory)
1       ERROR: cannot open `/home/malwarelab/malware_collection/windows/PE/0365AC13E50F3E7A629F45226FFB4EE7' (No such file or
        directory)
1       PE32 executable (DLL) (Windows CE) ARM, for MS Windows
1       PE32 executable (Windows CE) ARM Thumb, for MS Windows
1       PE32 executable (DLL) (GUI) Intel 80386 (stripped to external PDB), for MS Windows, Petite compressed
1       PE32+ executable (DLL) (console) x86-64, for MS Windows
1       PE32 executable (GUI) Intel 80386, for MS Windows, UPX compressed, COFF
1       PE32+ executable (DLL) (console) x86-64 (stripped to external PDB), for MS Windows
1       PE32 executable (DLL) (console) Intel 80386 (stripped to external PDB), for MS Windows, PECompact2 compressed
1       PE32 executable (DLL) Intel 80386 (stripped to external PDB) Mono/.Net assembly, for MS Windows
1       PE32 executable (console) Intel 80386, for MS Windows, RAR self-extracting archive
1       PE32 executable (GUI) Intel 80386 (stripped to external PDB) Mono/.Net assembly, for MS Windows, Petite compressed
1       PE32 executable (native) Intel 80386, for MS Windows, COFF
1       PE32+ executable (DLL) (GUI) x86-64 (stripped to external PDB), for MS Windows
1       PE32 executable (Unknown subsystem 0x2476) Intel 80386, for MS Windows
1       PE32 executable (console) Intel 80386, for MS Windows, Nullsoft Installer self-extracting archive
1       PE32 executable (GUI) Unknown processor type 0x13c, for MS Windows, UPX compressed
1       PE32 executable (console) Intel 80386 (stripped to external PDB) Mono/.Net assembly, for MS Windows
1       PE32 executable (GUI) Intel 80386, 32rtm DOS extender, UPX compressed
1       PE32 executable (console) Intel 80386 (stripped to external PDB), for MS Windows, Petite compressed
1       PE32 executable (Unknown subsystem 0x1002) Intel 80386 system file, for MS Windows
1       PE32 executable (Unknown subsystem 0x102) Intel 80386, for MS Windows
1       PE32 executable (GUI) Intel 80386 (stripped to external PDB), for MS Windows, InstallShield self-extracting archive
1       PE32 executable (GUI) Unknown processor type 0x4c, for MS Windows
1       PE32+ executable (console) x86-64 Mono/.Net assembly, for MS Windows
1       PE32 executable (DLL) (GUI) Intel 80386 (stripped to external PDB), for MS Windows, PECompact2 compressed
1       PE32 executable (GUI) Intel 80386, 32rtm DOS extender
1       PE32 executable (console) Intel 80386 (stripped to external PDB), for MS Windows, COFF
1       PE32 executable (GUI) Intel Itanium (stripped to external PDB), for MS Windows, UPX compressed
1       PE32 executable (GUI) Intel 80386, for MS Windows, PECompact2 compressed, RAR self-extracting archive
1       PE32 executable (Unknown subsystem 0x4601) Intel 80386, for MS Windows
1       PE32 executable (native) Intel 80386, for MS Windows, PECompact2 compressed
1       PE32 executable (DLL) (GUI) Intel 80386 (stripped to external PDB) system file, for MS Windows, PECompact2 compressed
1       PE32 executable (GUI) Unknown processor type 0x144, for MS Windows
1       PE32+ executable (GUI) Intel Itanium, for MS Windows
1       PE32 executable (console) Intel 80386, for MS Windows, UPX compressed, ARJ self-extracting archive
1       PE32 executable (DLL) (GUI) Intel 80386 system file, for MS Windows
1       PE32 executable (GUI) Intel 80386 (stripped to external PDB) system file, for MS Windows, PECompact2 compressed
1       PE32 executable (GUI) MIPS R10000, for MS Windows
```