CLAIMS

1. A computer-implementable method for detecting malicious files in non-isolated environment, the method comprises:

a preparatory step, including:

- creating a collection of files, the collection of files comprises at least one malicious executable file and at least one non-malicious executable file;
- analyzing at least one executable file from the collection of files, including:
  - extracting data from a binary form and/or a disassembled form of the at least one executable file,
  - determining parameters based on the extracted data,
  - determining parameters specific to malicious files and/or vice versa to non-malicious files by a statistical method, and
  - extracting a first data flow graph and a second data flow graph;
- constructing a first feature vector and a second feature vector based on the parameters;
- training an ensemble of classifiers including: a first trained classifier trained on the basis of the first feature vector, a second trained classifier trained on the basis of the second feature vector; a third classifier trained on the basis of the first data flow graph, a fourth classifier trained on the basis of the second data flow graph; and
- predefining a decisive priority for each of the ensemble of classifiers; and

an operating step, including:

- obtaining at least one executable file;
- detecting whether the at least one executable file is a malicious executable file using the ensemble of classifiers trained at the preparatory step; and
- outputting the result of the analyzing at least one executable file.

2. The method of claim 1, wherein the first feature vector comprises all numerical parameters from the parameters, and the second feature vector comprises all string parameters from the parameters.

3. The method of claim 1, further comprising extracting at least the following file features from the at least one executable file:

the first data flow graph,

the second data flow graph,

Metadata,

an Import Table,

an Export Table, and

a Byte/Entropy Histogram.

4. The method of claim 1, wherein for the binary form of the at least one executable file the parameters

include:

a Byte n-grams,

information from the executable file fields,

a file section entropy,

metadata, and

line length distribution histograms.

5. The method of claim 4, wherein the information from the executable file fields includes:

a number of sections in the file,

a size of headers, a code size,

an availability of digital signature,

an availability of imported dll and functions where these libraries have been used,

names of exported functions,

data from a resource catalog, and

an assembler version.

6. The method of claim 4, wherein the file section entropy is calculated using a sliding-window method followed by a number of statistical measures to characterize the obtained entropy measure vector.

7. The method of claim 4, wherein the metadata is extracted from a header of the binary form of the at least one executable file, wherein the metadata includes:

a number of sections in the file,

a size of sections,

an availability of a checksum,

a file size,

a creation time,

an operating system version, and

an assembler version.

8. The method of claim 1, wherein for the disassembled form of the at least one executable file the parameters include:

characters,

registers,

operation codes,

a system interface (Windows APIs) access rate,

sections, and

metadata.

9. The method of claim 8, further comprising defining a frequency of occurrence of the characters

extracted from the disassembled form of the at least one executable file, wherein the characters include: “-“, “+”, “*”, ”]”, ”[”, ”?”, ”@”.

10. The method of claim 8, further comprising additionally calculating a ratio between a number of known registers extracted from the disassembled form of the at least one executable file and a number of unknown registers extracted from the disassembled form of the at least one executable file.

11. The method of claim 8, wherein the sections extracted from at least one disassembled executable file are predefined and the sections include: .text, .data, .bss, .rdata, .edata, .idata, .rsrc, .tls and .reloc.

12. The method of claim 8, wherein the metadata extracted from the disassembled form of the at least one executable file includes the file size and a number of code lines.

13. The method of claim 1, wherein the analysis result is binary or probabilistic.

14. The method of claim 1, wherein the decisive priority of each of the ensemble of classifiers is defined based on a test sample, wherein at least the following is used for estimation: accuracy, completeness, or F-measure.

15. A system for detecting malicious files in a non-isolated environment, the system comprises:

a data extraction module configured to implement the preparatory step in the method of any one of claims 1-14;

a computer device configured to implement the operating step in the method of any one of claims 1-14; and

an output module configured to output the result of the analyzing at least one executable file in the method of any one of claims 1-14.