GitHub - packing-box/awesome-executable-packing: A curated list of awesome resources related to executable packing

By dhondta

Archived: 2026-04-05 20:22:06 UTC

- README
- Code of conduct
- Contributing
- CC0-1.0 license

> A curated list of resources related to executable packing (including Portable Executable, Executable and Linkable Format and others) containing references to books, papers, blog posts, and other written resources but also packers and tools for detecting packers and unpacking executables.

Packing is the action of modifying an executable in a way that does not modify its purpose. It is generally one or a combination of the following operations:

- bundling: makes a single executable with multiple files
- compression: compresses the executable to reduce its original size
- encoding: obfuscates the executable by encoding it
- encryption: obfuscates the executable by encrypting it
- mutation: alters the executable's code so that it uses a modifided instruction set and architecture (e.g. using oligomorphism)
- protection: makes the reversing of the executable harder (i.e. using anti-debugging, anti-tampering or other tricks)
- virtualization: embeds a virtual machine that allows to virtualize executable's instructions

Contents

📚 Literature

Documentation

- 🌎 a.out (FreeBSD manual pages)
- 🌎 A.out binary format
- 🌎 About anti-debug tricks
- 📊 Android packers: Separating from the pack
- 📌 Anti debugging protection techniques with examples
- 📓 Anti-unpacker tricks
- 📄 Anti-unpacker tricks - Part 14 (and previous parts)
- 📊 API deobfuscator: Resolving obfuscated API functions in modern packers
- 🌎 Armouring the ELF: Binary encryption on the UNIX platform
- 📗 The art of memory forensics: Detecting malware and threats in Windows, Linux, and mac memory
- 📊 The art of unpacking
- 🌎 Awesome executable packing
- 🌎 Awesome LLVM security
- 📌 Cloak and dagger: Unpacking hidden malware attacks
- 📖 Cluster analysis
- 🌎 Clustering algorithms
- 🌎 COM binary format
- 🌎 Common object file format (COFF)
- 🌎 Comparison of executable file formats
- 📰 A complexity measure
- 📰 Cyclomatic complexity density and software maintenance productivity
- 📊 Dealing with virtualization packers
- 🌎 Defacto2
- 📰 Do we need hundreds of classifiers to solve real world classification problems?
- 📊 Dynamic binary analysis and obfuscated codes
- 🌎 elf (FreeBSD manual pages)
- 📌 Entropy and the distinctive signs of packer PE files
- 📓 Evading machine learning malware detection
- 🌎 Executable and linkable format (ELF)
- 📋 Executable and linking format (ELF) specification
- 🌎 Executable file formats
- 📌 Explained: Packer, crypter, and protector
- 🌎 FatELF: Universal binaries for Linux (HALTED)
- 📰 Feature selection: A data perspective
- 📓 Gunpack: Un outil générique d'unpacking de malwares
- 📰 How to use t-SNE effectively
- 📋 Hyperion: Implementation of a PE-Crypter
- 📜 Implementing your own generic unpacker
- 🌎 Learn symbolic execution and angr
- 📊 LIEF: Library to instrument executable formats

- 📌 Mach-O - A look at apple executable files
- 🌎 Mach-O file format reference
- 📊 Mach-O internals
- 📖 Machine learning
- 📌 Making our own executable packer
- 🌎 The malware analyst's guide to aPLib decompression
- 📰 The matthews correlation coefficient (MCC) should replace the ROC AUC as the standard metric for assessing binary classification
- 📋 Microsoft portable executable and common object file format specification
- 🌎 MITRE ATT&CK | T1027.002 | obfuscated files or information: Software packing - Enterprise
- 🌎 MITRE ATT&CK | T1406.002 | obfuscated files or information: Software packing - Mobile
- 🌎 MZ disk operating system (DOS)
- 📊 NotPacked++: Evading static packing detection
- 🌎 OllyDbg OEP finder scripts
- 🔖 On the worst-case complexity of timsort
- 📊 One packer to rule them all: Empirical identification, comparison and circumvention of current antivirus detection techniques
- 📜 One packer to rule them all: Empirical identification, comparison and circumvention of current antivirus detection techniques
- 📰 Packer analysis report debugging and unpacking the NsPack 3.4 and 3.7 packer
- 📌 Packer detection tool evaluation
- 📄 Packers
- 📌 Packers/Protectors for Linux
- 📊 Packing-box: Breaking detectors & visualizing packing
- 📊 Packing-box: Improving detection of executable packing
- 📊 Packing-box: Playing with executable packing
- 📌 Parsing mach-O files
- 📗 Pattern recognition and machine learning (Information science and statistics)
- 🌎 PE format - Win32 apps
- 📜 PinDemonium: A DBI-based generic unpacker for Windows executables
- 🌎 Portable executable (PE)
- 📗 Practical malware analysis: The hands-on guide to dissecting malicious software
- 📌 ProtectMyTooling - Don't detect tools, detect techniques
- 📊 Qualitative and quantitative evaluation of software packers
- 📊 Reverse engineering malware: Binary obfuscation and protection
- 📊 Runtime packers testing experiences
- 📊 Runtime packers: The hidden problem?
- 📰 Standards and policies on packer use
- 📗 Surreptitious software: Obfuscation, watermarking, and tamperproofing for software protection
- 🔖 A survey of dimensionality reduction techniques
- 📊 TitanMist: Your first step to reversing nirvana
- 📌 Tuts 4 you - UnPackMe (.NET)

- 📌 Tuts 4 you | unpackme
- 📗 The "Ultimate" anti-debugging reference
- 📄 Unpacking binary 101
- 📌 Unpacking the potential of "Packing box"
- 📌 Unpacking, reversing, patching
- 📊 Virtual machine obfuscation
- 📊 WaveAtlas: Surfing through the landscape of current malware packers
- 📊 We can still crack you! General unpacking method for Android Packer (NO ROOT)
- 📊 When malware is packing heat
- 📋 Win32 portable executable packing uncovered
- 📌 Writing a packer
- 📌 Writing a PE packer
- 📌 Writing a simple PE packer in detail
- 🌎 x86 disassembly/Windows executable files
- 🌎 YARA - The pattern matching swiss knife for malware researchers.

Back to top

Scientific Research

- 📰 2-SPIFF: A 2-stage packer identification method based on function call graph and file attributes (December 2021) ⭐
- 📰 Absent extreme learning machine algorithm with application to packed executable identification (January 2016)
- 📰 An accurate packer identification method using support vector machine (January 2014)
- 📓 Adaptive unpacking of Android Apps (May 2017)
- 🎓 Advanced feature engineering for static detection of executable packing (June 2024) ⭐
- 📰 Advanced preprocessing of binary executable files and its usage in retargetable decompilation (December 2014)
- 📰 Adversarial attacks against Windows PE malware detection: A survey of the state-of-the-art (May 2023) ⭐
- 📰 Adversarial EXEmples: A survey and experimental evaluation of practical attacks on machine learning for windows malware detection (September 2021) ⭐
- 🎓 Adversarial learning on static detection techniques for executable packing (June 2023) ⭐
- 📓 Adversarial malware binaries: Evading deep learning for malware detection in executables (September 2018) ⭐
- 🎓 Adversarial tool for breaking static detection of executable packing (August 2024) ⭐
- 📓 Adversarially robust assembly language model for packed executables detection (November 2025) ⭐
- 📰 All-in-one framework for detection, unpacking, and verification for malware analysis (January 2019) ⭐
- 📰 Analysis of machine learning approaches to packing detection (October 2023) ⭐ ⭐
- 📰 Anti-emulation trends in modern packers: A survey on the evolution of anti-emulation techniques in UPA packers (May 2018)

- 📰 API-MalDetect: Automated malware detection framework for windows based on API calls and deep learning techniques (September 2023) ⭐
- 🎓 An application of machine learning to analysis of packed mac malware (May 2022) ⭐
- 📓 Application of string kernel based support vector machine for malware packer identification (August 2013)
- 📰 The application research of virtual machine in packers (August 2011)
- 📓 AppSpear: Bytecode decrypting and DEX reassembling for packed Android malware (November 2015)
- 📰 The arms race: Adversarial search defeats entropy used to detect malware (October 2018)
- 📕 Assessing static and dynamic features for packing detection (October 2024) ⭐
- 📄 Assessing the impact of packing on machine learning-based malware detection and classification systems (September 2025) ⭐
- 📰 Auditing static machine learning anti-Malware tools against metamorphic attacks (March 2021) ⭐
- 🎓 Automated static analysis of virtual-machine packers (August 2013)
- 📰 Automatic analysis of malware behavior using machine learning (December 2011)
- 📰 Automatic generation of adversarial examples for interpreting malware classifiers (March 2020)
- 📓 Automatic static unpacking of malware binaries (October 2009)
- 📰 BareUnpack: Generic unpacking on the bare-metal operating system (December 2018)
- 📰 Benchmark for filter methods for feature selection in high-dimensional classification data (March 2020) ⭐
- 📰 Beyond the sandbox: Leveraging symbolic execution for evasive malware classification (February 2025) ⭐
- 📰 Binary-code obfuscations in prevalent packer tools (October 2013)
- 📰 BinStat tool for recognition of packed executables (September 2010)
- 📰 Birds of a feature: Intrafamily clustering for version identification of packed malware (September 2020) ⭐
- 📓 BitBlaze: A new approach to computer security via binary analysis (December 2008)
- 📓 BODMAS: An open dataset for learning based temporal analysis of PE malware (May 2021) ⭐
- 📓 Boosting scalability in anomaly-based packed executable filtering (November 2011)
- 🎓 Building a malware mutation tool (June 2024)
- 🎓 Building a smart and automated tool for packed malware detections using machine learning (June 2020)
- 🎓 Building high-quality datasets of packed executables - Enhancing static detection models via curated packed binary datasets (August 2025) ⭐
- 📰 Bypassing anti-analysis of commercial protector methods using DBI tools (January 2021) ⭐
- 📰 Bypassing heaven’s gate technique using black-box testing (November 2023) ⭐
- 📓 BYTEWEIGHT: Learning to recognize functions in binary code (August 2014)
- 📓 ByteWise: A case study in neural network obfuscation identification (January 2018)
- 📓 Certified robustness of static deep learning-based malware detectors against patch and append attacks (November 2023) ⭐
- 📓 Challenging anti-virus through evolutionary malware obfuscation (April 2016)
- 📓 Chosen-instruction attack against commercial code virtualization obfuscators (April 2022) ⭐
- 📰 Classification of malware by using structural entropy on convolutional neural networks (April 2018)

- 📰 Classification of packed executables for accurate computer virus detection (October 2008)
- 📓 Classifying packed malware represented as control flow graphs using deep graph convolutional neural network (March 2020) ⭐
- 📓 Classifying packed programs as malicious software detected (December 2016)
- 📰 A close look at a daily dataset of malware samples (January 2019) ⭐
- 🎓 Code obfuscation techniques for software protection (April 2012)
- 📓 Collective classification for packed executable identification (September 2011)
- 📰 A compact multi-step framework for packing identification in portable executable files for malware analysis (February 2024) ⭐
- 📓 A comparative analysis of classifiers in the recognition of packed executables (November 2019) ⭐
- 📰 A comparative analysis of software protection schemes (June 2014)
- 📓 A comparative assessment of malware classification using binary texture analysis and dynamic analysis (September 2011)
- 📓 Comparing malware samples for unpacking: A feasibility study (August 2016)
- 🎓 Complexity-based packed executable classification with high accuracy (December 2008)
- 📓 A comprehensive solution for obfuscation detection and removal based on comparative analysis of deobfuscation tools (October 2021) ⭐
- 🎓 Computational-intelligence techniques for malware generation (October 2015)
- 📰 Conceptual and empirical comparison of dimensionality reduction algorithms (PCA, KPCA, LDA, MDS, SVD, LLE, ISOMAP, LE, ICA, t-SNE) (May 2021)
- 📰 A consistently-executing graph-based approach for malware packer identification (April 2019) ⭐
- 📰 Construction and evaluation of the new heuristic malware detection mechanism based on executable files static analysis (August 2018)
- 📓 A control flow graph-based signature for packer identification (October 2017) ⭐
- 📰 Control flow-based opcode behavior analysis for malware detection (July 2014)
- 📓 Countering entropy measure attacks on packed software detection (January 2012)
- 📓 Cryptographic function detection in obfuscated binaries via bit-precise symbolic loop mapping (May 2017)
- 🔖 Deceiving end-to-end deep learning malware detectors using adversarial examples (January 2019) ⭐
- 📓 Deceiving portable executable malware classifiers into targeted misclassification with practical adversarial examples (March 2020)
- 📄 Decoding the secrets of machine learning in malware classification: A deep dive into datasets, feature extraction, and model performance (November 2023) ⭐
- 📓 Denial-of-service attacks on host-based generic unpackers (December 2009)
- 🎓 Deobfuscation of packed and virtualization-obfuscation protected binaries (June 2011)
- 📕 Deobfuscation of virtualization-obfuscated code through symbolic execution and compilation optimization (April 2018)
- 📓 Deobfuscation of virtualization-obfuscated software: A semantics-based approach (October 2011)
- 📓 Design and development of a new scanning core engine for malware detection (October 2012)
- 🎓 Design and implementation of a modular executable packer - Experimenting with packing techniques and static detection (June 2025) ⭐

- 📓 Design and performance evaluation of binary code packing for protecting embedded software against reverse engineering (May 2010)
- 📰 Detecting obfuscated malware using reduced opcode set and optimised runtime trace (May 2016)
- 📓 Detecting obfuscated viruses using cosine similarity analysis (March 2007)
- 📓 Detecting packed executable file: Supervised or anomaly detection method? (August 2016)
- 📰 Detecting packed executables based on raw binary data (June 2010)
- 📓 Detecting packed executables using steganalysis (December 2014)
- 🎓 Detecting packed PE files: Executable file analysis for the Windows operating system (June 2021) ⭐
- 📓 Detecting traditional packers, decisively (October 2013)
- 📰 Detecting unknown malicious code by applying classification techniques on opcode patterns (February 2012)
- 📓 Detection of metamorphic malware packers using multilayered LSTM networks (November 2020) ⭐
- 📓 Detection of packed executables using support vector machines (July 2011)
- 📓 Detection of packed malware (August 2012)
- 📓 DexHunter: Toward extracting hidden code from packed Android applications (September 2015)
- 📓 Disabling anti-debugging techniques for unpacking system in user-level debugger (October 2019)
- 📰 DroidPDF: The obfuscation resilient packer detection framework for Android Apps (July 2020)
- 📓 Dynamic binary instrumentation for deobfuscation and unpacking (November 2009)
- 📓 Dynamic classification of packing algorithms for inspecting executables using entropy analysis (October 2013)
- 📓 A dynamic heuristic method for detecting packed malware using naive bayes (November 2019) ⭐
- 📰 Effective, efficient, and robust packing detection and classification (May 2019) 🌟🌟🌟
- 📰 An efficient algorithm to extract control flow-based features for ioT malware detection (April 2021) ⭐
- 📓 Efficient and automatic instrumentation for packed binaries (June 2009)
- 📰 Efficient automatic original entry point detection (January 2019)
- 📰 An efficient block-discriminant identification of packed malware (August 2015)
- 📓 Efficient malware packer identification using support vector machines with spectrum kernel (July 2013)
- 📰 Efficient SVM based packer identification with binary diffing measures (July 2019)
- 📰 ELF-Miner: Using structural knowledge and data mining methods to detect new (Linux) malicious executables (March 2012)
- 📓 EMBER2024 - A benchmark dataset for holistic evaluation of malware classifiers (August 2025) ⭐
- 🔖 EMBER: An open dataset for training static PE malware machine learning models (April 2018) ⭐⭐
- 📓 An empirical evaluation of an unpacking method implemented with dynamic binary instrumentation (September 2011)
- 📓 Encoded executable file detection technique via executable file header analysis (April 2009)
- 📰 Enhanced metamorphic techniques-A case study against havex malware (August 2021) ⭐
- 📓 Enhancing machine learning based malware detection model by reinforcement learning (November 2018)
- 📓 Entropy analysis to classify unknown packing algorithms for malware detection (May 2016) ⭐
- 📰 An entropy-based distance measure for analyzing and detecting metamorphic malware (June 2018)

- 📓 Entropy-driven visualization in gview: Unveiling the unknown in binary file formats (September 2024) ⭐
- 📰 ERMDS: A obfuscation dataset for evaluating robustness of learning-based malware detection system (May 2023)
- 📓 ESCAPE: Entropy score analysis of packed executable (October 2012)
- 📓 Ether: Malware analysis via hardware virtualization extensions (October 2008)
- 📓 Eureka: A framework for enabling static malware analysis (October 2008)
- 📰 Evading anti-malware engines with deep reinforcement learning (March 2019) ⭐
- 📓 Evading packing detection: Breaking heuristic-based static detectors (July 2024) ⭐
- 📓 Experimental comparison of machine learning models in malware packing detection (September 2020) ⭐
- 📓 An experimental study on identifying obfuscation techniques in packer (June 2016)
- 📓 Experimental toolkit for manipulating executable packing (June 2024) ⭐ ⭐
- 🎓 Experimental toolkit for studying executable packing - Analysis of the state-of-the-art packing detection techniques (June 2022) ⭐
- 📓 Exploring adversarial examples in malware detection (May 2019) ⭐
- 📰 Fast and robust fixed-point algorithms for independent component analysis (May 1999)
- 📓 A fast flowgraph based classification system for packed and polymorphic malware on the endhost (April 2010)
- 📓 A fast randomness test that preserves local detail (October 2008)
- 📰 Feature selection for malware detection based on reinforcement learning (December 2019)
- 📰 Feature selection for packer classification based on association rule mining (August 2024) ⭐
- 📓 Feature set reduction for the detection of packed executables (June 2014)
- 📰 File packing from the malware perspective: Techniques, analysis approaches, and directions for enhancements (December 2022) ⭐ ⭐
- 📓 Fileprints: Identifying file types by n-gram analysis (June 2005)
- 📓 A fine-grained classification approach for the packed malicious code (October 2012)
- 📰 A framework for metamorphic malware analysis and real-time detection (February 2015)
- 📰 Functionality-preserving black-box optimization of adversarial windows malware (May 2021) ⭐
- 📰 G3MD: Mining frequent opcode sub-graphs for metamorphic malware detection of existing families (December 2018)
- 🔖 Generating adversarial malware examples for black-box attacks based on GAN (February 2020) ⭐
- 📓 A generic approach to automatic deobfuscation of executable code (May 2015) ⭐
- 📓 Generic black-box end-to-end attack against state of the art API call based malware classifiers (September 2018) ⭐
- 📰 Generic packing detection using several complexity analysis for accurate malware detection (January 2014)
- 📓 Generic unpacker of executable files (April 2015)
- 📓 Generic unpacking method based on detecting original entry point (November 2013)
- 🔖 Generic unpacking of self-modifying, aggressive, packed binary programs (May 2009)
- 📓 Generic unpacking techniques (February 2009)
- 📓 Generic unpacking using entropy analysis (October 2010)

- 📓 GUARD: Generic API de-obfuscation and obfuscated malware unpacking with sIAT (March 2025) ⭐
- 📰 Hashing-based encryption and anti-debugger support for packing multiple files into single executable (February 2018)
- 📓 A heuristic approach for detection of obfuscated malware (June 2009)
- 📰 A heuristics-based static analysis approach for detecting packed PE binaries (October 2013)
- 📓 Highlighting the impact of packed executable alterations with unsupervised learning (April 2025) ⭐
- 📰 Hunting for metamorphic engines (November 2006)
- 📰 Identifying malware packers through multilayer feature engineering in static analysis (February 2024) ⭐
- 📓 An implementation of a generic unpacking method on Bochs Emulator (September 2009)
- 📰 An improved method for packed malware detection using PE header and section table information (September 2019)
- 📰 Improving malware detection using multi-view ensemble learning (August 2016) ⭐
- 📜 Incremental clustering of malware packers using features based on transformed CFG (November 2022) ⭐
- 📓 Information theoretic method for classification of packed and encoded files (September 2015)
- 📓 Instructions-based detection of sophisticated obfuscation and packing (October 2014)
- 🔖 Intriguing properties of adversarial ML attacks in the problem space (March 2020) ⭐
- 🔖 Intriguing properties of neural networks (February 2014)
- 📰 A learning model to detect maliciousness of portable executable using integrated feature set (January 2017)
- 🔖 Learning to evade static PE machine learning malware models via reinforcement learning (January 2018) ⭐
- 📓 Limits of static analysis for malware detection (December 2007)
- 📗 Longitudinal study of the prevalence of malware evasive techniques (December 2021) ⭐
- 🔖 MAB-Malware: A reinforcement learning framework for attacking static malware classifiers (April 2021) ⭐
- 📓 A machine-learning-based framework for supporting malware detection and analysis (September 2021) ⭐
- 🎓 Maitland: Analysis of packed and encrypted malware via paravirtualization extensions (June 2012)
- 📓 Mal-EVE: Static detection model for evasive malware (August 2015)
- 📰 Mal-flux: Rendering hidden code of packed binary executable (March 2019)
- 📰 Mal-XT: Higher accuracy hidden-code extraction of packed binary executable (November 2018)
- 📰 Mal-xtract: Hidden code extraction using memory analysis (January 2017)
- 📰 MaliCage: A packed malware family classification framework based on DNN and GAN (August 2022) ⭐
- 📰 The MALICIA dataset: Identification and analysis of drive-by download operations (February 2015)
- 📰 Malware analysis using multiple API sequence mining control flow graph (July 2017)
- 📰 Malware analysis using visualized images and entropy graphs (February 2015)
- 🎓 Malware detection through opcode sequence analysis using machine learning (June 2015)
- 📓 Malware family classification method based on static feature extraction (December 2017)
- 📓 Malware images: Visualization and automatic classification (July 2011)

- 📓 Malware makeover: Breaking ML-based static analysis by modifying executable bytes (May 2021) ⭐
- 📓 Malware obfuscation techniques: A brief survey (November 2010)
- 📓 Malware obfuscation through evolutionary packers (July 2015)
- 📰 Malwise - An effective and efficient classification system for packed and polymorphic malware (June 2013)
- 📓 McBoost: Boosting scalability in malware collection and analysis using statistical classification of executables (December 2008)
- 📕 Measuring and defeating anti-instrumentation-equipped malware (June 2017)
- 📓 Memory behavior-based automatic malware unpacking in stealth debugging environment (October 2010)
- 📓 MetaAware: Identifying metamorphic malware (December 2007)
- 📓 Metadata recovery from obfuscated programs using machine learning (December 2016)
- 📰 Metamorphic malware detection based on support vector machine classification of malware sub-signatures (September 2016)
- 📰 Metamorphic malware identification using engine-specific patterns based on co-opcode graphs (August 2020) ⭐
- 📰 Mimicking anti-viruses with machine learning and entropy profiles (2019-05-21)
- 📓 MLxPack: Investigating the effects of packers on ML-based malware detection systems using static and dynamic traits (May 2022) ⭐
- 📓 Modern Linux malware exposed (June 2018)
- 📰 MSG: Missing-sequence generator for metamorphic malware detection (March 2025) ⭐
- 📓 MutantX-S: Scalable malware clustering based on static features (June 2013)
- 📓 The new signature generation method based on an unpacking algorithm and procedure for a packer detection (February 2011)
- 🔖 Novel feature extraction, selection and fusion for effective malware family classification (March 2016)
- 📰 A novel framework for image-based malware detection with a deep neural network (October 2021) ⭐
- 📓 Obfuscation-resilient executable payload extraction from packed malware (August 2021) ⭐
- 📰 Obfuscation: The hidden malware (August 2011)
- 📓 Obfuscation: Where are we in anti-DSE protections? (a first attempt) (December 2019)
- 📓 Obfuscator-LLVM: Software protection for the masses (May 2015)
- 📓 OmniUnpack: Fast, generic, and safe unpacking of malware (December 2007)
- 📰 On deceiving malware classification with section injection (August 2022) ⭐
- 🔖 On evaluating adversarial robustness (February 2019) ⭐
- 📓 On the (Im)possibility of obfuscating programs (August 2001)
- 📰 On the (im)possibility of obfuscating programs (2) (April 2012)
- 📰 On the adoption of anomaly detection for packed executable filtering (June 2014)
- 📓 On the feasibility of malware unpacking via hardware-assisted loop profiling (August 2023) ⭐
- 📰 Opcode sequences as representation of executables for data-mining-based unknown malware detection (May 2013) ⭐
- 📰 Opcodes as predictor for malware (January 2008)
- 📓 OPEM: A static-dynamic approach for machine-learning-based malware detection (September 2012)
- 📰 Original entry point detection based on graph similarity (April 2024) ⭐

- 📰 An original entry point detection method with candidate-sorting for more effective generic unpacking (January 2015)
- 📓 Packed code detection using shannon entropy and homomorphic encrypted executables (October 2024) ⭐
- 📰 Packed malware detection using entropy related analysis: A survey (November 2015)
- 📰 Packed malware variants detection using deep belief networks (March 2020)
- 📓 Packed PE file detection for malware forensics (December 2009)
- 📰 Packer classification based on association rule mining (July 2022) ⭐
- 📓 Packer classifier based on PE header information (April 2015)
- 📰 Packer detection for multi-layer executables using entropy analysis (March 2017) ⭐
- 📓 Packer identification based on metadata signature (December 2017) ⭐
- 📓 Packer identification method based on byte sequences (November 2018)
- 📓 Packer identification method for multi-layer executables with k-Nearest neighbor of entropies (October 2020) ⭐
- 📓 Packer identification using byte plot and Markov plot (September 2015)
- 📓 Packer identification using hidden Markov model (November 2017)
- 🎓 Packer-complexity analysis in PANDA (January 2018)
- 📓 PackGenome: Automatically generating robust YARA rules for accurate malware packer detection (November 2023) ⭐
- 🔖 PackHero: A scalable graph-based approach for efficient packer identification (July 2025) ⭐
- 🎓 Packing detection and classification relying on machine learning to stop malware propagation (December 2021) ⭐
- 🎓 Pandora's Bochs: Automatic unpacking of malware (January 2008)
- 📓 Pattern recognition techniques for the classification of malware packers (July 2010)
- 📰 PE file features in detection of packed executables (January 2012)
- 📓 PE file header analysis-based packed PE file detection technique (PHAD) (October 2008)
- 📓 PE-Miner: Mining structural information to detect malicious executables in realtime (September 2009)
- 📓 PE-Probe: Leveraging packer detection and structural information to detect malicious portable executables (June 2009)
- 📓 PEAL - Packed executable analysis (January 2012)
- 📰 Performance evaluation of filter-based feature selection techniques in classifying portable executable files (January 2018) ⭐
- 📰 PEzoNG: Advanced packer for automated evasion on Windows (December 2022) ⭐
- 📰 Pitfalls in machine learning for computer security (October 2024)
- 📓 PolyPack: An automated online packing service for optimal antivirus evasion (August 2009)
- 📓 PolyUnpack: Automating the hidden-code extraction of unpack-executing malware (December 2006)
- 📰 Potent and stealthy control flow obfuscation by stack based self-modifying code (April 2013)
- 📰 Practical attacks on machine learning: A case study on adversarial windows malware (September 2022) ⭐
- 📓 Preprocessing of binary executable files towards retargetable decompilation (July 2013)
- 📓 Prevalence and impact of low-entropy packing schemes in the malware ecosystem (February 2020) ⭐
- 📓 Program obfuscation by strong cryptography (February 2010)

- 📓 RAMBO: Run-Time packer analysis with multiple branch observation (July 2016)
- 🎓 REFORM: A framework for malware packer analysis using information theory and statistical methods (April 2010)
- 📓 Renovo: A hidden code extractor for packed executables (November 2007)
- 📓 RePEconstruct: Reconstructing binaries with self-modifying code and import address table destruction (October 2016)
- 📓 RePEF — A system for restoring packed executable file for malware analysis (July 2011)
- 📰 Replacement attacks against VM-protected applications (September 2012)
- 📓 Research and implementation of compression shell unpacking technology for PE file (May 2009)
- 📰 Research and implementation of packing technology for PE files (January 2013)
- 📓 Research of software information hiding algorithm based on packing technology (September 2020)
- 📰 Resurrecting anti-virtualization and anti-debugging: Unhooking your hooks (March 2021) ⭐
- 📰 Revealing packed malware (September 2008)
- 📓 Reverse engineering self-modifying code: Unpacker extraction (October 2010)
- 🎓 Robust static analysis of portable executable malware (December 2014)
- 📓 SATURN - Software deobfuscation framework based on LLVM (November 2019)
- 📰 SCORE: Source code optimization & reconstruction (July 2020)
- 📓 SE-PAC: A self-evolving packer classifier against rapid packers evolution (April 2021) ⭐
- 📰 Secure and advanced unpacking using computer emulation (August 2007)
- 📓 Semi-supervised learning for packed executable detection (September 2011)
- 📓 Semi-supervised learning for unknown malware detection (April 2011)
- 📰 Sensitive system calls based packed malware variants detection using principal component initialized multilayers neural networks (September 2018)
- 📰 Sequential opcode embedding-based malware detection method (March 2022) ⭐
- 📰 Singular value decomposition and metamorphic detection (November 2015)
- 📰 SMASH: A malware detection method based on multi-feature ensemble learning (August 2019)
- 📰 Software protection through anti-debugging (May 2007)
- 📓 SoK: (state of) the art of war: Offensive techniques in binary analysis (May 2016)
- 📓 SoK: Automatic deobfuscation of virtualization-protected applications (August 2021) ⭐
- 📓 SoK: Deep packer inspection: A longitudinal study of the complexity of run-time packers (May 2015) ⭐
- 🎓 Source-free binary mutation for offense and defense (December 2014)
- 📓 SPADE: Signature based packer detection (August 2012)
- 📓 Static analysis method on portable executable files for REMNUX based malware identification (October 2019)
- 📓 Static analysis of executables to detect malicious patterns (August 2003)
- 🎓 Static features exploration for executable packing with unsupervised learning (June 2023) ⭐
- 📰 Static malware detection & subterfuge: Quantifying the robustness of machine learning and current anti-virus (June 2018)
- 📓 A static, packer-agnostic filter to detect similar malware samples (July 2012)
- 📰 Structural analysis of binary executable headers for malware detection optimization (May 2017)
- 📰 Structural entropy and metamorphic malware (November 2013)

- 📓 Structural feature based anomaly detection for packed executable identification (June 2011)
- 📓 The study of evasion of packed PE from static detection (June 2012)
- 📓 A study of the packer problem and its solutions (September 2008)
- 📰 A survey on adversarial attacks for malware analysis (December 2024) ⭐
- 📰 A survey on automated dynamic malware-analysis techniques and tools (March 2008)
- 📓 A survey on machine learning-based detection and classification technology of malware (September 2021) ⭐
- 📰 A survey on malware analysis techniques: Static, dynamic, hybrid and memory analysis (September 2018)
- 📓 Survey on malware evasion techniques: State of the art and challenges (February 2012)
- 📰 A survey on run-time packers and mitigation techniques (November 2023) ⭐ ⭐
- 📓 Symbolic deobfuscation: From virtualized code back to the original (July 2018)
- 📓 Symbolic execution of obfuscated code (October 2015) ⭐
- 📓 Syntia: Synthesizing the semantics of obfuscated code (August 2017)
- 📰 A systematical and longitudinal study of evasive behaviors in windows malware (February 2022) ⭐
- 🔖 Technical report on the cleverhans v2.1.0 adversarial examples library (June 2018) ⭐
- 📓 Things you may not know about Android (Un) packers: A systematic study based on whole-system emulation. (February 2018)
- 📓 Thwarting real-time dynamic unpacking (January 2011)
- 📓 A token strengthened encryption packer to prevent reverse engineering PE files (January 2015)
- 📓 Toward generic unpacking techniques for malware analysis with quantification of code revelation (August 2009)
- 📓 Towards paving the way for large-scale Windows malware analysis: Generic binary unpacking with orders-of-magnitude performance boost (October 2018) ⭐
- 📓 Towards static analysis of virtualization-obfuscated binaries (October 2012)
- 🔖 Transcending transcend: Revisiting malware classification in the presence of concept drift (December 2021) ⭐
- 📓 Tutorial: An overview of malware detection and evasion techniques (December 2018)
- 📰 Two techniques for detecting packed portable executable files (June 2013)
- 📓 Unconditional self-modifying code elimination with dynamic compiler optimizations (October 2010)
- 📓 Understanding Linux malware (May 2018)
- 📓 Unknown malcode detection using OPCODE representation (December 2008)
- 📓 A unpacking and reconstruction system-agunpacker (January 2009)
- 🎓 Unpacking framework for packed malicious executables (July 2013)
- 📕 Unpacking malware in the real world: A step-by step guide (July 2024) ⭐
- 📰 Unpacking techniques and tools in malware analysis (September 2012)
- 📓 Unpacking virtualization obfuscators (August 2009)
- 🎓 Unsupervised clustering machine learning on packed executable (June 2022) ⭐
- 📰 UnThemida: Commercial obfuscation technique analysis with a fully obfuscated program (July 2018) ⭐
- 📰 Using entropy analysis to find encrypted and packed malware (March 2007)

- 📓 VABox: A virtualization-based analysis framework of virtualization-obfuscated packed executables (June 2021) ⭐
- 📓 VMAttack: Deobfuscating virtualization-based packed binaries (August 2017)
- 📓 VMHunt: A verifiable approach to partially-virtualized binary code simplification (October 2018) ⭐
- 📓 VMRe: A reverse framework of virtual machine protection packed binaries (June 2019)
- 📰 Watermarking, tamper-proofing, and obfuscation - Tools for software protection (August 2002)
- 📰 Wavelet decomposition of software entropy reveals symptoms of malicious code (December 2016)
- 📓 When malware is packin' heat; limits of machine learning classifiers based on static analysis features (January 2020) ⭐ ⭐
- 📰 WYSINWYX: What you see is not what you execute (August 2010)
- 📰 x64Unpack: Hybrid emulation unpacker for 64-bit Windows Environments and detailed analysis results on VMProtect 3.4 (July 2020) ⭐
- 📓 Xunpack: Cross-Architecture unpacking for Linux IoT malware (October 2023) ⭐

Back to top

🗂️ Datasets

- BODMAS - Code for our DLS'21 paper - BODMAS: An Open Dataset for Learning based Temporal Analysis of PE Malware.
- Contagio - Collection of the latest malware samples, threats, observations, and analyses.
- CyberCrime - C² tracking and malware database.
- Dataset of Packed ELF - Compilation of packed ELF samples.
- Dataset of Packed PE - Sanitized version of the original dataset, PackingData, removing packed samples from the Notpacked folder but also samples in packer folders that failed to be packed (having a same hash as the original unpacked executable).
- Ember - Collection of features from PE files that serve as a benchmark dataset for researchers.
- Ember2024 - Update to the EMBER2017 and EMBER2018 datasets.
- FFRI Dataset Scripts - Make datasets like FFRI Dataset.
- MaleX - Curated dataset of malware and benign Windows executable samples for malware researchers containing 1,044,394 Windows executable binaries and corresponding image representations with 864,669 labelled as malware and 179,725 as benign.
- Malfease - Dataset of about 5,000 packed malware samples.
- Malheur - Contains the recorded behavior of malicious software (malware) and has been used for developing methods for classifying and clustering malware behavior (see the JCS article from 2011).
- Malicia - Dataset of 11,688 malicous PE files collected from 500 drive-by download servers over a period of 11 months in 2013 (DISCONTINUED).
- MalShare - Free Malware repository providing researchers access to samples, malicious feeds, and Yara results.
- Malware Archive - Malware samples, analysis exercises and other interesting resources.
- The Malware Museum - Collection of malware programs, usually viruses, that were distributed in the 1980s and 1990s on home computers.

- MalwareBazaar - Project operated by abuse.ch aimed to collect and share malware samples, helping IT-security researchers and threat analysts protecting their constituency and customers from cyber threats.
- MalwareGallery - Yet another malware collection in the Internet.
- MalwareSamples - Bringing you the best of the worst files on the Internet.
- MalwareTips - Community-driven platform providing the latest information and resources on malware and cyber threats.
- OARC Malware Dataset - Semi-public dataset of 3,467 samples captured in the wild from Sep 2005 to Jan 2006 by mail traps, user submissions, honeypots and other sources aggregated by the OARC, available to qualified academic and industry researchers upon request.
- Open Malware Project - Online collection of malware samples (formerly Offensive Computing).
- PackingData - Original dataset with sample PE files packed with a large variety of packers, including ASPack, BeRoEXEPacker, exe32pack, eXpressor, FSG, JDPack, MEW, Molebox, MPRESS, Neolite, NSPack, Pckman, PECompact, PEtite, RLPack, UPX, WinUpack, Yoda's Crypter and Yoda's Protector.
- Packware - Datasets and codes that are needed to reproduce the experiments in the paper "When Malware is Packing Heat".
- RCE Lab - Crackme's, keygenme's, serialme's ; the "tuts4you" folder contains many packed binaries.
- Runtime Packers Testset - Dataset of 10 common Malware files, packed with about 40 different runtime packers in over 500 versions and options, with a total of about 5,000 samples.
- SAC - Slovak Antivirus Center, non-commercial project of AVIR and ESET companies ; contains packers, detectors and unpackers.
- SOREL - Sophos-ReversingLabs 20 Million dataset.
- theZoo - Project created to make the possibility of malware analysis open and available to the public.
- ViruSign - Another online malware database.
- VirusSamples - Best of the worst kind of files on the Internet.
- VirusShare - Virus online database with more than 44 millions of samples.
- VirusSign - Giant database dedicated to combating malware in the digital world.
- VirusTotal - File analysis Web service for detecting malware.
- VX Heaven - Site dedicated to providing information about computer viruses.
- VX Underground - PL-CERT based open source MWDB python application holding a malware database containing every APT sample from 2010 and over 7.5M maliciousbinaries.
- VXvault - Online malware database.
- WildList - Cooperative listing of malwares reported as being in the wild by security professionals.

Back to top

📦 Packers

After 2010

- Alienyze - Advanced software protection and security for Windows 32-bit executables.
- Alternate EXE Packer - Compression tool for executable files (type EXE) or DLL's relying on UPX 3.96.
- Amber - Position-independent(reflective) PE loader that enables in-memory execution of native PE files(EXE, DLL, SYS).

- Andromeda - Custom packer used in malware campaigns using RunPE techniques for evading AV mitigation methods.
- APKProtect - APK encryption and shell protection supporting Java and C++.
- Armadillo - Incorporates both a license manager and wrapper system for protecting PE files.
- ASM Guard - Packer utility for compressing and complicating reversing compiled native code (native files), protecting resources, adding DRM, and packing into an optimized loader.
- ASPack - Advanced solution created to provide Win32 EXE file packing and to protect them against non-professional reverse engineering.
- ASProtect 32 - Multifunctional EXE packing tool designed for software developers to protect 32-bit applications with in-built application copy protection system.
- ASProtect 64 - Tool for protecting 64-bit applications and .NET applications for Windows against unauthorized use, industrial and home copying, professional hacking and analysis of software products distributed over the Internet and on any physical media.
- Astral-PE - Low-level mutator (Headers/EP obfuscator) for native Windows PE files (x32/x64).
- AutoIT - Legitimate executable encryption service.
- AxProtector - Encrypts the complete software you aim to protect, and shields it with a security shell, AxEngine, best-of-breed anti-debugging and anti-disassembly methods are then injected into your software.
- Backpack
- BangCle - Protection tool using the second generation Android Hardening Protection, loading the encrypted DEX file from memory dynamically.
- Bero - BEP (Bero EXE Packer) for 32-bit windows executables.
- BIN-crypter - EXE protection software against crackers and decompilers.
- BoxedApp Packer
- Code Virtualizer - Powerful code obfuscation system for Windows, Linux and macOS applications that helps developers to protect their sensitive code areas against Reverse Engineering with very strong obfuscation code, based on code virtualization.
- ConfuserEx - An open-source, free protector for .NET applications.
- Crinkler - Compressing linker for Windows, specifically targeted towards executables with a size of just a few kilobytes.
- DarkCrypt - Simply and powerful plugin for Total Commander used for file encryption using 100 algorithms and 5 modes.
- DexGuard - Android app obfuscation & security protocols for mobile app protection.
- DexProtector - Multi-layered RASP solution that secures your Android and iOS apps against static and dynamic analysis, illegal use and tampering.
- DotBundle - GUI tool to compress, encrypt ad password-protect a .NET application or embed .NET libraries.
- DotNetZ - Straightforward and lightweight, command-line piece of software written in C that allows you to compress and pack Microsoft .NET Framework executable files.
- ElecKey - Suite of software and tools that offer a complete solution for software protection, copy protection, and license management.
- ELF Packer - Encrypts 64-bit elf files that decrypt at runtime.

- ELF-Encrypter - Collection of programs to encrypt ELF binaries using various algorithms.
- ELF-Packer - Simple Polymorphic x86_64 Runtime Code Segment Cryptor.
- ELFCrypt - Simple ELF crypter using RC4 encryption.
- ELFkickers - A collection of programs that access and manipulate ELF files.
- ELFuck - ELF packer for i386 original version from sk2 by sd.
- Enigma Protector - Professional system for executable files licensing and protection.
- Enigma Virtual Box - Application virtualization system for Windows.
- Eronona-Packer - This is a packer for exe under win32.
- EXE Bundle - Bundles application files into a single PE32 file.
- EXE Stealth - Anti-cracking protection and licensing tool for PE files featuring compression and encryption polymorphic technology.
- Ezuri - A Simple Linux ELF Runtime Crypter.
- GzExe - Utility that allows to compress executables as a shell script.
- hXOR-Packer - PE packer with Huffman compression and XOR encryption.
- Hyperion
- LIAPP - Easiest and most powerful mobile app security solution.
- LM-X License Manager - Lets you protect your products against piracy by enforcing various levels of security, save time, and reduce business risks.
- m0dern_p4cker - Just a modern packer for elf binaries (works on Linux executables only).
- MidgetPack - ELF binary packer, such as burneye, upx or other tools.
- MPRESS - Compresses (using LZMA) and protects PE, .NET or Mach-O programs against reverse engineering.
- NetCrypt - A proof-of-concept packer for .NET executables, designed to provide a starting point to explain the basic principles of runtime packing.
- .netshrink - Executable compressor for your Windows or Linux .NET application executable file using LZMA.
- NPack - Can compress 32bits and 64bits exe, dll, ocx, scr Windows program.
- Obsidium - Feature-rich professional software protection and licensing system designed as a cost effective and easy to implement, yet reliable and non-invasive way to protect your 32- and 64-bit Windows software applications and games from reverse engineering.
- oplzkwp - Library for ELF obfuscation ; it uses PRESENT and blake244 to encrypt your payload on the fly.
- Origami - Packer compressing .net assemblies, (ab)using the PE format for data storage.
- OS-X_Packer - Binary packer for the Mach-O file format.
- Pakkero - Binary packer written in Go made for fun and educational purpose.
- Pakr - In-memory packer for macOS Mach-O bundles.
- Papaw - Permissively-licensed packer for ELF executables using LZMA Zstandard or Deflate compression.
- PE-Packer - Simple packer for Windows 32-bits PE files.
- PE-Toy - A PE file packer.
- PELock - Software protection system for Windows executable files ; protects your applications from tampering and reverse engineering, and provides extensive support for software license key management, including support for time trial periods.

- PePacker - Simple PE Packer Which Encrypts .text Section I release a simple PE file packer which encrypts the .text section and adds a decryption stub to the end of the last section.
- PEShield - PE-SHiELD is a program, which encrypts 32-bit Windows EXE files, leaving them still executable.
- PESpin
- PEtite - Free Win32 (Windows 95/98/2000/NT/XP/Vista/7/etc) executable (EXE/DLL/etc) compressor.
- PEzoNG - Framework for automatically creating stealth binaries that target a very low detection rate in a Windows environment.
- PEzor - Open-Source Shellcode & PE Packer.
- pocrypt - Naive Proof of Concept Crypter for GNU/Linux ELF64.
- ProtectMyTooling - Multi-Packer wrapper letting us daisy-chain various packers, obfuscators and other Red Team oriented weaponry.
- ps2-packer - Create packed ELF files to run on the PS2.
- RapidEXE - Simple and efficient way to convert a PHP/Python script to a standalone executable.
- sherlocked
- Silent-Packer - Silent Packer is an ELF / PE packer written in pure C.
- Simple-PE32-Packer - Simple PE32 Packer with aPLib compression library.
- SimpleDPack - A very simple windows EXE packing tool for learning or investigating PE structure.
- Smart Packer - Packs 32 & 64bit applications with DLLs, data files, 3rd party run-time into one single executable that runs instantly, with no installs or hassles.
- Squishy - Modern packer developed for 64kb demoscene productions, targets 32bit and 64bit executables.
- theArk - Windows x86 PE Packer In C++.
- Themida - From Renovo paper: Themida converts the original x86 instructions into virtual instructions in its own randomized instruction set, and then interpret these virtual instructions at run-time.
- UPX - Ultimate Packer for eXecutables.
- VirtualMachineObfuscationPoC - Obfuscation method using virtual machine.
- VMProtect - Protects code by executing it on a virtual machine with non-standard architecture that makes it extremely difficult to analyze and crack the software.
- Ward - Simple implementation of an ELF packer that creates stealthy droppers for loading malicious ELFs in-memory.
- Woody Wood Packer - ELF packer - encrypt and inject self-decryption code into executable ELF binary target.
- xorPacker - Simple packer working with all PE files which cipher your exe with a XOR implementation.
- XyrisPack
- zELF - A modular ELF64 packer for Linux x86_64 featuring 22 compression codecs, ML-based codec selection, and support for both static and PIE binaries.
- ZProtect - Renames metadata entities and supports advanced obfuscation methods that harden protection scheme and foil reverse engineering altogether.

Back to top

Between 2000 and 2010

- 20to4 - Executable compressor that is able to stuff about 20k of finest code and data into less than 4k.
- ACProtect - Application that allows to protect Windows executable files against piracy, using RSA to create and verify the registration keys and unlock code.
- AHPack - PE and PE+ file packer.
- Application Protector - Tool for protecting Windows applications.
- AT4RE Protector - Very simple PE files protector programmed in ASM.
- AverCryptor - Small and very handy utility designed to encrypt notes in which you can store any private information - it helps to hide your infection from antiviruses.
- BurnEye - ELF encryption program, x86-linux binary.
- ByteBoozer - Commodore 64 executable packer.
- cryptelf - Modifies binary by appending code to handle runtime decryption, changing the program EP and changing the .note segment to LOAD ; encrypts the .text section by XORing its bytes with a key.
- CryptExec - Next-generation runtime binary encryption using on-demand function extraction.
- EXE Guarder - Licensing tool for PE files allowing to compress and specify a password notice.
- EXE Wrapper - Protects any EXE file with a password from non-authorized execution.
- Exe32Pack - Compresses Win32 EXEs, DLLs, etc and dynamically expands them upon execution.
- EXECryptor - Protects EXE programs from reverse engineering, analysis, modifications and cracking.
- ExeFog - Simple Win32 PE files packer.
- eXPressor - Used as a compressor this tool can compress EXE files to half their normal size.
- FSG - *Fast Small Good*, perfect compressor for small exes, eg.
- GHF Protector - Executable packer / protector based on open source engines Morphine and AHPack.
- HackStop - EXE and COM programs encrypter and protector.
- Kkrunchy - Small exe packer primarily meant for 64k intros.
- Laturi - Linker and compressor intended to be used for macOS 1k, 4k and perhaps 64K intros.
- mPack - Mario PACKersimple Win32 PE Executable compressor.
- NSPack - 32/64-bits exe, dll, ocx, scr Windows program compressor.
- NTPacker - PE file packer relying on aPlib for compression and/or XOR for encryption.
- PECompact - Windows executable compressor featuring third-party plug-ins offering protection against reverse engineering.
- RDMC - DMC algorithm based packer.
- RLPack - Compresses your executables and dynamic link libraries in a way that keeps them small and has no effect on compressed file functionality.
- RSCC - ROSE Super COM Crypt ; polymorph cryptor for files greater than 300-400B and smaller than 60kB.
- RUCC - ROSE Ultra COM Compressor ; COM and EXE compression utility based on 624.
- Sentinel HASP Envelope - Wrapping application that protects the target application with a secure shield, providing a means to counteract reverse engineering and other anti-debugging measures.
- sePACKER - Simple Executable Packer is compressing executables' code section inorder to decrease size of binary files.
- Shiva - Tool to encrypt ELF executables under Linux.
- tElock - Practical tool that intends to help developers who want to protect their work and reduce the size of the executable files.

- TTProtect - Professional protection tool designed for software developers to protect their PE applications against illegal modification or decompilation.
- UPack - Compresses Windows PE file.
- UPX-Scrambler - Scrambler for files packed with UPX (up to 1.06) so that they cannot be unpacked with the '-d' option.
- WinUpack - Graphical interface for Upack, a command-line program used to create self-extracting archives from Windows PE files.
- x86.Virtualizer - x86 Virtualizer.
- XComp - PE32 image file packer and rebuilder.
- Yoda Crypter - Supports polymorphic encryption, softice detection, anti-debug API's, anti-dumping, etc, encrypts the Import Table and erases PE Header.
- Yoda Protector - Free, open source, Windows 32-bit software protector.

Back to top

Before 2000

- 32Lite - Compression tool for executable files created with Watcom C/C++ compiler.
- 624 - COM packer that can compress COM programs shorter than 25000 bytes.
- ABK Scrambler - COM file scrambler and protector recoded from ABKprot.
- AEP - Addition Encode-Protective for COM and EXE file.
- AINEXE - DOS executable packer (part of the AIN Archiver suite).
- aPack - 16-bit real-mode DOS executable (.EXE and .COM) compressor.
- AVPack - Encrypts EXE or COM files so that they'll be able to start on your PC only.
- AXE - Program compression utility.
- BIN-Lock - COM file scrambler for preventing reverse engineering.
- BitLok - COM and EXE file protector.
- CauseWay Compressor - DOS EXE compressor.
- CC Pro - COM and EXE executable file compression utility.
- CEXE - Compresses an input EXE into a smaller executable (only runs on WinNT, Win2000 and above - won't run on Win95 or Win98).
- COMProtector - Adds a security envelope around DOS .COM files by randomly encrypting it and adding several anti-debugging tricks.
- CrackStop - Tool that creates a security envelope around a DOS EXE file to protect it against crackers.
- Crunch - File encryptor for COM and EXE files.
- EPack - EXE and COM file compressor ; works with DOS/Windows95 files.
- ExeGuard - DOS EXE files free protector using anti-debugging ticks to prevent hacking, analysis and unpacking.
- EXELOCK 666 - Utility for protecting .EXE files so no lamers can hack out the copyright.
- Fire-Pack
- FSE - Final Fantasy Security Envelope freeware for protecting COM and EXE progams.
- Gardian Angel - COM and EXE encrypter and protector using a variety of anti-debugging tricks.
- JMCryptExe - DOS EXE encrypter.

- LGLZ - DOS EXE and COM file compressor using modified LZ77.
- LzExe - MS-DOS executable file compressor.
- Mask - Tool that prevents COM program from being cracked by using encryption and anti-debugging tricks.
- Megalite - MS-DOS executable file compressor.
- Mess - This tool does the same as HackStop, with the exception that it is freeware for non-commercial use.
- Morphine - Application for PE files encryption.
- Neolite - Compresses Windows 32-bit EXE files and DLLs.
- PACK - Executable files compressor.
- Pack-Ice
- PCShrink - Windows 9x/NT executable file compressor relying on the aPLib compression library.
- PE Diminisher - Simple PE packer relying on the aPLib compression library.
- PE-Protector - Encrypter/protector for Windows 9x/ME to protect executable files PEagainst reverse engineering or cracking with a very strong protection.
- PEBundle - Physically attaches DLL(s) to an executable, resolving dependencies in memory.
- PEPack - PE compression tool based on the code of a newer version of PE-SHiELD.
- PKlite - Easy-to-use file compression program for compressing DOS and Windows executable files.
- Pro-Pack - DOS executable file compressor.
- RERP - ROSE's EXE Relocation Packer.
- RJCrush - EXE and COM files compressor with the ability to compress overlays.
- Scorpion - EXE and COM file encrypter and protector.
- SecuPack - Win32 executable compressor.
- Shrinker - Compresses (up to 70%) 16 and 32 bit Windows and real mode DOS programs.
- SPack
- $PIRIT - COM/EXE executable files polymorphic encryptor.
- SysPack - Device drivers compressor.
- T-Pack - Executable COM-FILE compressor (LZ77) optimized for small files like BBS-Addys or similar files.
- TinyProg - EXE and COM programs compressor.
- TRAP - EXE and COM files encrypter and protector.
- Vacuum - Runtime Compressor for DOS32 executables.
- VGCrypt - PE crypter for Win95/98/NT.
- WinLite - Compresses Windows executables (such as Pklite, Diet or Wwpack) for executables programs under DOS.
- WWPack - Squeezes EXE files, compresses relocation tables, optimizes headers, protects EXE files from hacking.
- XE - PE32 image file packer and rebuilder.
- XorCopy - COM file XOR-based encrypter.
- XORER - COM file XOR-based encrypter.
- XPA - DOS executable packer.
- XPack - EXE/COM/SYS executable file compressor.

Back to top

🔧 Tools

- Android Unpacker - Presented at Defcon 22: Android Hacker Protection Level 0.
- Angr - Platform-agnostic binary analysis framework.
- APKiD - Android application Identifier for packers, protectors, obfuscators and oddities - PEiD for Android.
- aPLib - Compression library based on the algorithm used in aPACK.
- AppSpear - Universal and automated unpacking system suitable for both Dalvik and ART.
- Assiste (Packer) - Assiste.com's example list of packers.
- AVClass - Python tools to tag / label malware samples.
- Bintropy - Prototype analysis tool that estimates the likelihood that a binary file contains compressed or encrypted bytes.
- BinUnpack - Unpacking approach free from tedious memory access monitoring, therefore introducing very small runtime overhead.
- Binutils - The GNU Binutils are a collection of binary tools for Linux (it namely includes Readelf).
- BitBlaze - Analysis platform that features a novel fusion of static and dynamic analysis techniques, mixed concrete and symbolic execution, and whole-system emulation and binary instrumentation, all to facilitate state-of-the art research on real security problems.
- Capa - Open-source tool to identify capabilities in PE, ELF or .NET executable files.
- Capstone - Lightweight multi-platform, multi-architecture disassembly framework.
- Cave-Finder - Tool to find code cave in PE image (x86 / x64) - Find empty space to place code in PE files.
- CFF Explorer - PE32/64 and .NET editor, part of the Explorer Suite.
- ChkEXE - Identifies almost any EXE/COM packer, crypter or protector.
- Clamscan Unpacker - Unpacker derived from ClamAV.
- COM2EXE - Free tool for converting COM files to EXE format.
- de4dot - .NET deobfuscator and unpacker.
- de4js - JavaScript Deobfuscator and Unpacker.
- Defacto2 Analyzers Archive - Collection of 60 binary files analysers for MS-DOS and Windows32 from the 1990s and the 2000s.
- Defacto2 Packers Archive - Collection of 460 binary and data file packers for MS-DOS and Windows32 from the 1990s and 2000s.
- Defacto2 Unpackers Archive - Collection of 152 binary files unpackers for MS-DOS and Windows 32 from the 1990s and 2000s.
- DIE - Detect It Easy ; Program for determining types of files.
- DSFF - DataSet File Format for exchanging datasets and converting to ARFF (for use with Weka), CSV or Packing-Box's dataset structure.
- DynamoRIO - Runtime code manipulation system that supports code transformations on any part of a program, while it executes.
- Emulator - Symantec Endpoint Protector (from v14) capability to create a virtual machine on the fly to identify, detonate, and eliminate malware hiding inside custom malware packers.

- EtherUnpack - Precision universal automated unpacker (successor of PolyUnpack).
- Eureka - Binary static analysis preparation framework implementing a novel binary unpacking strategy based on statistical bigram analysis and coarse-grained execution tracing.
- EXEInfo-PE - Fast detector for executable PE files.
- ExeScan - Executable file analyzer which detects the most famous EXE/COM Protectors, Packers, Converters and compilers.
- EXETools - Forum for reverse engineering and executale packing related topics.
- FUU - Fast Universal Unpacker.
- GetTyp - File format detection program for DOS based on special strings and byte code.
- GUnpacker - Shell tool that performs OEP positioning and dumps decrypted code.
- Gym-Malware - This is a malware manipulation environment for OpenAI's gym.
- IDR - Interactive Delphi Reconstructor.
- ImpREC - This can be used to repair the import table for packed programs.
- Justin - Just-In-Time AV scanning ; generic unpacking solution.
- Language 2000 - Ultimate compiler detection utility.
- LIEF - Library to Instrument Executable Formats ; Python package for parsing PE, ELF, Mach-O and DEX formats, modifying and rebuilding executables.
- Lissom - Retargetable decompiler consisting of a preprocessing part and a decompilation core.
- LordPE - PE header viewer, editor and rebuilder.
- Malheur - Tool for the automatic analysis of malware behavior (recorded from malicious software in a sandbox environment).
- MalUnpack - Dynamic unpacker based on PE-sieve.
- Manalyze - Robust parser for PE files with a flexible plugin architecture which allows users to statically analyze files in-depth.
- MRC - (Mandiant Red Curtain) Free software for Incident Responders that assists with the analysis of malware ; it examines executable files (e.g., .exe, .dll, and so on) to determine how suspicious they are based on a set of criteria.
- .NET Deobfuscator - List of .NET Deobfuscators and Unpackers.
- NotPacked++ - Attack tool for altering packed samples so that they evade static packing detection.
- Oedipus - A Python framework that uses machine learning algorithms to implement the metadata recovery attack against obfuscated programs.
- OEPdet - Automated original-entry-point detector.
- OllyDbg Scripts - Collection of OllyDbg scripts for unpacking many different packers.
- OmniUnpack - New technique for fast, generic, and safe unpacking of malware by monitoring the execution in real-time and detecting the removed layers of packing.
- PackerAttacker - Tool that uses memory and code hooks to detect packers.
- PackerBreaker - Tool for helping unpack, decompress and decrypt most of the programs packed, compressed or encrypted using advanced emulation technology.
- PackerGrind - Adaptive unpacking tool for tracking packing bahaviors and unpacking Android packed apps.
- PackerID - Fork of packerid.py using PEid signatures and featuring additional output types, formats, digital signature extraction, and disassembly support.

- PackID - Packer identification multiplatform tool/library using the same database syntax as PEiD.
- Packing-Box - Docker image gathering many packing-related tools and for making datasets of packed executables for use with machine learning.
- PANDA - Platform for Architecture-Neutral Dynamic Analysis.
- PANDI - Dynamic packing detection solution built on top of PANDA.
- Pandora's Bochs - Extension to the Bochs PC eumlator to enable it to monitor execution of the unpacking stubs for extracting the original code.
- PCjs - Uses JavaScript to recreate the IBM PC experience, using original ROMs, CPUs running at their original speeds, and early IBM video cards and monitors.
- PE Compression Test - List of packers tested on a few sample executables for comparing compressed sizes.
- PE Detective - This GUI tool can scan single PE files or entire directories (also recursevely) and generate complete reports.
- PE-bear - Freeware reversing tool for PE files aimed to deliver fast and flexible “first view” for malware analysts, stable and capable to handle malformed PE files.
- PEdump - Dump windows PE files using Ruby.
- Pefeats - Utility for extracting 119 features from a PE file for use with machine learning algorithms.
- Pefile - Multi-platform Python module to parse and work with Portable Executable files.
- PEFrame - Tool for performing static analysis on PE malware and generic suspicious files.
- PEiD - Packed Executable iDentifier.
- PEiD (CLI) - Python implementation of PEiD featuring an additional tool for making new signatures.
- PEiD (yara) - Yet another implementation of PEiD with yara.
- PeLib - PE file manipulation library.
- PEPack - PE file packer detection tool, part of the Unix package "pev".
- PEscan - CLI tool to scan PE files to identify how they were constructed.
- PETools - Old-school reverse engineering tool (with a long history since 2002) for manipulating PE files.
- PEview - Provides a quick and easy way to view the structure and content of 32-bit Portable Executable (PE) and Component Object File Format (COFF) files.
- PExplorer - Most feature-packed program for inspecting the inner workings of your own software, and more importantly, third party Windows applications and libraries for which you do not have source code.
- Pin - Dynamic binary instrumentation framework for the IA-32, x86-64 and MIC instruction-set architectures that enables the creation of dynamic program analysis tools.
- PINdemonium - Unpacker for PE files exploiting the capabilities of PIN.
- PolyUnpack - Implemention attempt of the general approach for extracting the original hidden code of PE files without any heuristic assumptions.
- PortEx - Java library for static malware analysis of PE files with a focus on PE malformation robustness and anomaly detection.
- PROTECTiON iD - PE file signature-based scanner.
- ProTools - Programmer's Tools, a web site dedicated for all kinds of tools and utilities for the true WinBloze programmer, including packers, crypters, etc.
- PyPackerDetect - Small Python script/library to detect whether an executable is packed.
- PyPackerDetect (refactored) - A complete refactoring of the original project to a Python package with a console script to detect whether an executable is packed.

- PyPeid - Yet another implementation of PEiD with yara-python.
- Quick Unpack - Generic unpacker that facilitates the unpacking process.
- RDG Packer Detector - Packer detection tool.
- Reko - Free decompiler for machine code binaries.
- REMINDer - Packing detection tool based on the entropy value of the entry point section and the WRITE attribute.
- REMnux - Linux toolkit for reverse-engineering and analyzing malicious software.
- Renovo - Detection tool built on top of TEMU (dynamic analysis component of BitBlaze) based on the execution of newly-generated code and monitoring memory writes after the program starts.
- ResourceHacker - Resource editor for 32bit and 64bit Windows applications.
- RetDec - Retargetable machine-code decompiler based on LLVM.
- RTD - Rose Patch - TinyProt/Rosetiny Unpacker.
- RUPP - ROSE SWE UnPaCKER PaCKaGE (for DOS executables only).
- SAFE - Static Analyzer For Executables (available on demand).
- SecML Malware - Create adversarial attacks against machine learning Windows malware detectors.
- ShowStopper - Tool to help malware researchers explore and test anti-debug techniques or verify debugger plugins or other solutions that clash with standard anti-debug methods.
- StudPE - PE viewer and editor (32/64 bit).
- SymPack - Safe, portable, largely effective but not generic library for packing detection and unpacking ; part of the Norton Antivirus solution.
- Titanium Platform - Machine learning hybrid cloud platform that harvests thousands of file types at scale, speeds threat detection through machine learning binary analysis, and continuously monitors an index of over 10B files for future threats.
- TrID - Utility for identifying file types from their binary signatures.
- Triton - Dynamic binary analysis library.
- Tuts 4 You - Non-commercial, independent community dedicated to the sharing of knowledge and information on reverse code engineering.
- Unipacker - Automatic and platform-independent unpacker for Windows binaries based on emulation.
- UnpacMe - Automated malware unpacking service.
- Unpckarc - Packed executables detection tool relying on several heuristics.
- UU - Universal Unpacker.
- Uundo - Universal Undo - Universal Unpacker.
- Uunp (IDA Pro plugin) - IDA Pro debugger plug-in module automating the analysis and unpacking of packed binaries.
- UUP - Universal exe-file UnPacker.
- VMHunt - Set of tools for analyzing virtualized binary code ; now only supports 32 bit traces.
- VMUnpacker - Unpacker based on the technology of virtual machine.
- Winbindex - An index of Windows binaries, including download links for executables such as EXE, DLL and SYS files.
- yarGen - Generator for YARA rules - The main principle is the creation of yara rules from strings found in malware files while removing all strings that also appear in goodware files.

Back to top

Contributing

Contributions are welcome! Please read the contribution guidelines first.
