

ҚАЗАҚСТАН РЕСПУБЛИКАСЫ ҒЫЛЫМ ЖӘНЕ ЖОҒАРЫ БІЛІМ МИНИСТРЛІГІ

«Л.Н. ГУМИЛЕВ АТЫНДАҒЫ ЕУРАЗИЯ ҰЛТТЫҚ УНИВЕРСИТЕТІ» КЕАҚ

**Студенттер мен жас ғалымдардың
«GYLYM JÁNE BILIM - 2023»
XVIII Халықаралық ғылыми конференциясының
БАЯНДАМАЛАР ЖИНАҒЫ**

**СБОРНИК МАТЕРИАЛОВ
XVIII Международной научной конференции
студентов и молодых ученых
«GYLYM JÁNE BILIM - 2023»**

**PROCEEDINGS
of the XVIII International Scientific Conference
for students and young scholars
«GYLYM JÁNE BILIM - 2023»**

**2023
Астана**

УДК 001+37
ББК 72+74
G99

«GYLYM JÁNE BILIM – 2023» студенттер мен жас ғалымдардың XVIII Халықаралық ғылыми конференциясы = XVIII Международная научная конференция студентов и молодых ученых «GYLYM JÁNE BILIM – 2023» = The XVIII International Scientific Conference for students and young scholars «GYLYM JÁNE BILIM – 2023». – Астана: – 6865 б. - қазақша, орысша, ағылшынша.

ISBN 978-601-337-871-8

Жинаққа студенттердің, магистранттардың, докторанттардың және жас ғалымдардың жаратылыстану-техникалық және гуманитарлық ғылымдардың өзекті мәселелері бойынша баяндамалары енгізілген.

The proceedings are the papers of students, undergraduates, doctoral students and young researchers on topical issues of natural and technical sciences and humanities.

В сборник вошли доклады студентов, магистрантов, докторантов и молодых ученых по актуальным вопросам естественно-технических и гуманитарных наук.

УДК 001+37
ББК 72+74

ISBN 978-601-337-871-8

**©Л.Н. Гумилев атындағы Еуразия
ұлттық университеті, 2023**

Бұл әсіресе агроөнеркәсіптегі жер ресурстарын басқару мәселесінде өзекті, өйткені саланың және әрбір шаруашылық жүргізуші субъектінің жалпы жағдайы осы процестің тиімділігіне байланысты. Инновациялық технологиялар ауыл шаруашылығының ресурстық әлеуетін толық іске асыруға, сондай-ақ оны көбейту, жердің сапалық сипаттамаларын жақсартуға мүмкіндік береді.

Ауылшаруашылық ресурстарын басқарудың цифрлық трансформациясының оң әсері бұл — дақылдардың өнімділігінің өсуінен, еңбек өнімділігінің өсуінен, аграрлық өндіріске экологиялық жүктеменің төмендеуінен және т. б. көрініс табады.

Қорытынды. Біздің елімізде ауыл шаруашылығының жер-ресурстық әлеуетін басқаруда ақпараттық технологияларды пайдалану өндіріс көлемін ұлғайту және саладағы еңбек өнімділігі мен сапасын арттыру қажеттілігімен байланысты. Жүргізілген зерттеу негізінде аграрлық салада инновациялық цифрлық технологияларды әзірлемей және іске асырмай, қазіргі жағдайда саланың жер-ресурстық әлеуетін пайдаланудың тиімділігі мен тиімділігін арттыру мүмкін емес деген қорытынды жасауға болады. Сонымен қатар, ауыл шаруашылығында тиімсіз жер пайдалану жекелеген экономикалық субъектілердің де, жалпы саланың да тұрақты жұмыс істеуіне кері ықпал етеді, бұл сөзсіз өңірлер мен елдің экономикалық қауіпсіздігін қамтамасыз етуге кедергі келтіреді. Ауыл шаруашылығындағы жер ресурстарын басқару механизмінің цифрлық трансформациясы аграрлық өнім өндірудің инновациялық процестерін енгізуге және саланың табиғи ресурстық әлеуетінің жай-күйі мен перспективалары туралы өзекті ақпаратты пайдалануға негізделеді.

Пайдаланылған әдебиеттер

1. Баутин В. М., Мычка С. Ю. Формирование механизма управления ресурсным потенциалом предприятий АПК в условиях реализации политики импортозамещения // Век качества. 2017. № 2. С. 54—70.
2. Гаджиева К. Р. Механизм совершенствования системы управления воспроизводством земельных ресурсов регионального сельского хозяйства // РППЭ. 2018. № 10 (96). С. 28—35.
3. Булычева Е. А. Особенности цифровизации управления земельными ресурсами в муниципальном образова-нии // Материалы XIV Межд. студ. науч. конф. «Студенческий научный форум». URL: <https://scienceforum.ru/2022/article/2018028777> (дата обращения: 21.12.2021).
4. Мулярец С. А. Специфика и проблемы цифровой трансформации предприятий российского агропромышленного комплекса // Инновации и инвестиции. 2021. № 4. С. 315—320.
5. Хисамутдинова Ю. А. Особенности процесса цифровизации управления земельными ресурсами в муниципальном образовании // Стратегии развития социальных общностей, институтов и территорий : материалы VII Межд. науч.-практ. конф. Екатеринбург, 19–20 апреля 2021 г. : в 2 т. Екатеринбург: Изд-во Урал. ун-та, 2021. Т. 1. С. 319—323.
6. Samygin D., Baryshnikov N., Vinnichuk L., Glasunov I. Strategic models of optimization of support of farmers // Strategic models of optimization of support of farmers. Ponte. 2017. Vol. 73. Issue 4. Pp. 146—157. DOI: 10.21506/j.ponte.2017.4.44

УДК 004.056.57

УПАКОВКА ПРОГРАММ КАК МЕТОД ЗАЩИТЫ ВРЕДНОСНОГО ПРОГРАММНОГО ОБЕСПЕЧЕНИЕ

Жаксыбек Диас Мейрбекулы, Ахметова Жанар Жумановна
zhaksibek.dias@icloud.com

Магистрант факультета информационных технологий ЕНУ им. Л. Н. Гумелева,
Астана, Казахстан

Введение

Вредоносное программное обеспечение охватывает широкий спектр программных средств, предназначенных для нарушения конфиденциальности, целостности или доступности

компьютерных систем. С ростом распространенности киберугроз понимание механизмов, используемых авторами вредоносных программ для защиты и сокрытия своих творений, стало критически важным как для специалистов по кибербезопасности, так и для исследователей. Одним из таких механизмов является "упаковка", метод, который запутывает и шифрует вредоносный код, делая его более устойчивым к обнаружению и анализу [1, 2].

Основная цель этой статьи - представить углубленный анализ упаковки как метода защиты от вредоносных программ; с помощью всестороннего анализа, выяснить основные принципы этого механизма защиты и его последствия для обнаружения и устранения вредоносного ПО. При этом будут изучены различные типы упаковщиков, каждый из которых предлагает различные функции, такие как сжатие, шифрование и запутывание.

Эта статья призвана внести вклад в коллективные знания экспертов по кибербезопасности, тем самым обеспечивая более эффективные стратегии борьбы с вредоносным программным обеспечением.

Упаковка вредоносных программ

Упаковка в контексте вредоносных программ — это метод, используемый авторами вредоносных программ для защиты и сокрытия своего вредоносного кода от обнаружения и анализа программами обеспечения безопасности и исследователями. Используя упаковщики, они могут уменьшить цифровой след своего вредоносного ПО, повысить его шансы на обход мер безопасности и усложнить процесс обратной разработки [3].

Цель упаковщика – это преобразование исходного кода ПО в более компактный формат с сохранением функциональности. Обычно это достигается за счет комбинации методов сжатия, шифрования и запутывания. Сжатие уменьшает размер вредоносного ПО, ускоряя его передачу по сети и затрудняя обнаружение. Шифрование включает в себя кодирование данных для защиты их от несанкционированного доступа, что еще больше затрудняет анализ. Обфускация — это изменение кода таким образом, что его трудно понять или интерпретировать, добавляя вредоносному ПО еще один уровень защиты.

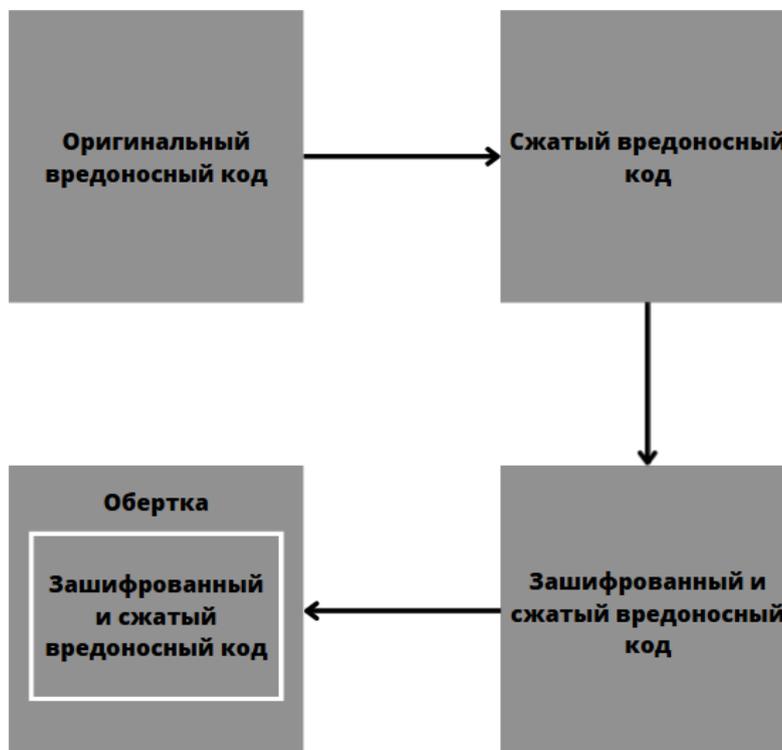


Рисунок 1. Принцип работы упаковщиков

Существует три основных типа упаковщиков: упаковщики среды выполнения (runtime packers), упаковщики во времени установки и гибридные упаковщики. Каждый тип служит для разных целей и предлагает различные уровни защиты от вредоносных программ.

1. Упаковщики среды выполнения: эти упаковщики защищают вредоносное ПО во время его выполнения. Упакованная вредоносная программа остается зашифрованной и скрытой до тех пор, пока не будет запущена в целевой системе. После запуска упаковщик расшифровывает и распаковывает код вредоносного ПО в памяти, позволяя ему функционировать должным образом. Данный вид упаковщиков широко используются из-за их способности обходить методы статического анализа, которые сосредоточены на изучении вредоносного ПО без его выполнения.

2. Упаковщики во времени установки защищают вредоносное ПО на этапе установки. В отличие от упаковщиков среды выполнения, упакованное вредоносное ПО расшифровывается и распаковывается перед записью на диск целевой системы. Такой подход затрудняет обнаружение вредоносного ПО программным обеспечением безопасности в процессе установки, но менее эффективна от анализа во время выполнения.

3. Гибридные упаковщики: как следует из названия, гибридные упаковщики сочетают в себе элементы двух предыдущих видов упаковщиков. Они могут использовать несколько уровней шифрования, сжатия и обфускации, требуя расшифровки и распаковки вредоносного ПО на разных этапах выполнения или установки. Гибридные упаковщики обеспечивают более высокий уровень защиты, что затрудняет обнаружение и анализ вредоносного программного обеспечения.

Использование методов упаковки значительно усложняет процесс обнаружения и анализа вредоносного ПО. Упаковщики эффективно скрывают истинный функционал вредоносного кода, вынуждая антивирусное программное обеспечение разрабатывать более сложные методы выявления и противодействия этим угрозам. Основная проблема заключается в том, что упакованное вредоносное ПО может обойти традиционное обнаружение на основе сигнатур, основанное на выявлении известных шаблонов во вредоносном коде [4].

Для решения проблем, связанных с упакованными вредоносными программами, программное обеспечение безопасности использует различные методы обнаружения, включая статический анализ, эвристический анализ и анализ, основанный на поведении.

- Статический анализ. Этот метод включает в себя изучение вредоносного ПО без его запуска. Основное внимание уделяется структуре кода, заголовкам файлов и другим метаданным для выявления аномалий или известных сигнатур упакованных файлов. Хотя статический анализ полезен для обнаружения известных упаковщиков, он затрудняет выявление нестандартных или ранее неизвестных методов упаковки.

- Эвристический анализ. Эвристический анализ опирается на набор правил и алгоритмов для выявления потенциальных вредоносных программ на основе их характеристик и моделей поведения. Этот метод является более гибким, чем статический анализ, поскольку он может обнаруживать ранее неизвестные или модифицированные упаковщики. Однако эвристический анализ может давать ложные срабатывания и по-прежнему не может обнаруживать сложные методы упаковки.

- Анализ на основе поведения. Этот подход отслеживает поведение файла во время выполнения в контролируемой среде, например в песочнице. Наблюдая за действиями файла, поведенческий анализ может идентифицировать упакованное вредоносное ПО на основе его поведения во время выполнения, даже если метод упаковки неизвестен. Хотя этот метод является мощным средством обнаружения упакованных вредоносных программ, он может быть ресурсоемким и может не подходить для защиты в режиме реального времени.

Несмотря на достижения в методах обнаружения, упаковщики продолжают создавать серьезные проблемы для антивирусного программного обеспечения. Авторы вредоносных программ постоянно совершенствуют свои методы упаковки и создают собственные упаковщики, чтобы избежать обнаружения. Чтобы опережать эти угрозы, исследователи безопасности и разработчики антивирусов должны постоянно обновлять и улучшать свои алгоритмы

обнаружения и внедрять новые методы для борьбы с постоянно развивающимися методами упаковки вредоносных программ.

Недавние исследования на разработке методов машинного обучения и искусственного интеллекта для улучшения обнаружения и анализа упакованных вредоносных программ, показывают многообещающие результаты [5]. Эти методы могут значительно расширить возможности программного обеспечения для защиты и позволить им адаптироваться к постоянно меняющейся среде вредоносных программ.

Популярные инструменты и методы упаковки

Авторам вредоносных программ доступны различные инструменты упаковки и сокрытия своего вредоносного кода. Некоторые из наиболее часто используемых инструментов упаковки это UPX, Themida и VMProtect. Каждый инструмент предлагает уникальные функции и различные уровни защиты от вредоносных программ, что способствует их популярности среди злоумышленников.

UPX (Ultimate Packer for eXecutables [6]) — это упаковщик с открытым исходным кодом, который в основном ориентирован на сжатие. Он поддерживает несколько исполняемых форматов и совместим с несколькими операционными системами. UPX прост в использовании и широко распространен, но его популярность также означает, что большинство средств защиты могут легко обнаруживать и распаковывать вредоносные программы, упакованные с помощью UPX.

Themida — это мощный и сложный коммерческий упаковщик, предлагающий несколько уровней защиты, включая шифрование, сжатие и обфускацию. Он специально разработан для предотвращения обратной разработки за счет включения методов защиты от отладки и создания дампа. Авторы вредоносных программ предпочитают Themida за ее способность защищать как от статического, так и от динамического анализа [7].

VMProtect — еще один продвинутый коммерческий упаковщик, который использует виртуализацию для защиты от вредоносных программ. Он преобразует исходный код в серию зашифрованных инструкций, которые могут выполняться только на пользовательской виртуальной машине [8]. Такой подход сильно усложняет анализ вредоносных программ исследователями и значительно усложняет распаковку.

В дополнение к этим популярным инструментам упаковки авторы вредоносного ПО часто создают собственные упаковщики, чтобы еще больше увеличить шанс избежать обнаружения и анализа. Пользовательские упаковщики могут сочетать различные методы шифрования, сжатия и запутывания. Разрабатывая собственные упаковщики, авторы вредоносного ПО могут опередить антивирусное программное обеспечение и исследователей, которые могут еще не иметь необходимых инструментов или знаний для обнаружения и анализа этих новых методов упаковки.

Распаковка вредоносного ПО

Обратная разработка (реверс-инжиниринг) — это процесс деконструкции и анализа части программного обеспечения, чтобы понять его функциональность. В контексте упакованного вредоносного ПО реверс-инжиниринг обычно включает в себя распаковку вредоносного ПО для раскрытия его истинного содержимого, что позволяет исследователям в области безопасности изучить угрозу и противодействовать ей.

Работа с упакованными вредоносными программами сопряжена с многочисленными проблемами и ограничениями для реверс-инженеров. Эти проблемы включают идентификацию используемого упаковщика, обход методов защиты от отладки и анализа, а также преодоление сложностей пользовательских упаковщиков и виртуализации. Поскольку авторы вредоносных программ продолжают совершенствовать свои методы упаковки, реверс-инженеры также должны совершенствовать свои навыки и инструменты, чтобы опережать угрозу.

Существует три основных подхода к распаковке вредоносных программ: статическая распаковка, динамическая распаковка и ручная распаковка.

- Статическая распаковка: этот метод означает анализ упакованного вредоносного ПО без его запуска. Цель состоит в том, чтобы определить процедуру распаковки и извлечь исходный код без запуска вредоносного ПО. Статическая распаковка может быть эффективна против

известных упаковщиков, но имеет проблемы с уникальными или сильно запутанными упаковщиками.

- **Динамическая распаковка.** Динамическая распаковка включает запуск вредоносного ПО в контролируемой среде, такой как песочница или виртуальная машина. Отслеживая процесс выполнения, исследователи могут определить, когда процедура распаковки завершена и исходный код загружен в память. Этот метод эффективен против неизвестных и пользовательских упаковщиков, но может быть ресурсоемким и рискованным из-за необходимости выполнения вредоносного ПО.

- **Ручная распаковка.** Ручная распаковка — трудоемкий процесс, требующий высокого уровня знаний в реверс-инжиниринге. Он включает в себя использование отладчика для пошаговой процедуры распаковки и ручное извлечение исходного кода. Ручная распаковка может быть эффективной даже против самых опытных упаковщиков, но требует много времени и значительных навыков.

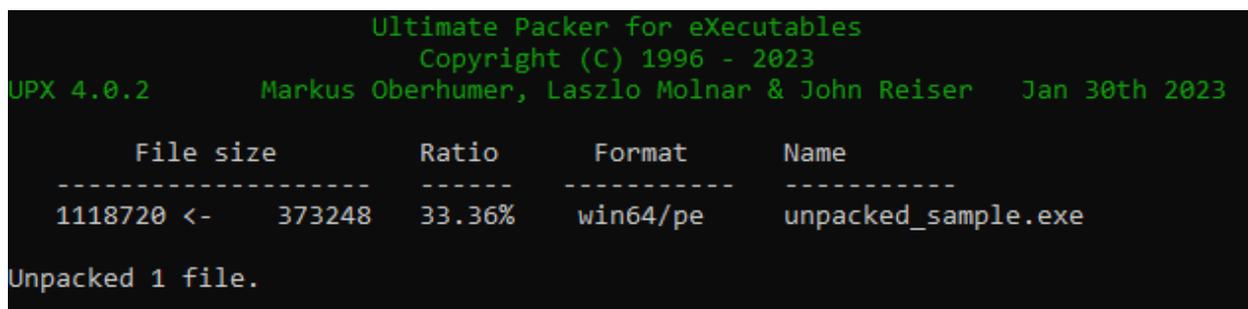
Рассмотрим на простом примере процесс распаковки исполняемого файлом упакованного с помощью UPX (с именем файла `packed_sample.exe`). Чтобы распаковать этот исполняемый файл, мы будем использовать инструмент командной строки UPX, который можно загрузить с официального сайта UPX (<https://upx.github.io/>).

Для распаковки исполняемого файла нужно выполнить следующую команду:

```
upx -d packed_sample.exe -o unpacked_sample.exe
```

Эта команда указывает UPX распаковать (-d) файл «`packed_sample.exe`» и вывести распакованный файл как «`unpacked_sample.exe`».

UPX распакует исполняемый файл и отобразит ход выполнения вместе со степенью сжатия:



```
Ultimate Packer for eXecutables
Copyright (C) 1996 - 2023
UPX 4.0.2      Markus Oberhumer, Laszlo Molnar & John Reiser   Jan 30th 2023

  File size      Ratio      Format      Name
  -----      -
1118720 <-  373248  33.36%  win64/pe  unpacked_sample.exe

Unpacked 1 file.
```

Рисунок 2. Скриншот процесса распаковки UPX

После завершения процесса распаковки вы можете использовать дизассемблеры, отладчики или другие инструменты реинжиниринга для анализа распакованного исполняемого файла «`unpacked_sample.exe`».

Заключение

В этой статье исследовано значение упаковки как метода защиты от вредоносного ПО. Упаковщики создают серьезные проблемы для антивирусного программного обеспечения и исследователей, эффективно скрывая истинную природу вредоносного кода, что требует разработки более сложных методов обнаружения и анализа. Были изучены различные типы упаковщиков, а также популярные инструменты упаковки.

Процесс распаковки вредоносного ПО является важнейшим компонентом обратной разработки, позволяющим специалистам по безопасности анализировать угрозы и противодействовать им. Несмотря на проблемы и ограничения, возникающие при работе с упакованными вредоносными программами, непрерывная разработка новых методов и инструментов необходима для того, чтобы идти в ногу с постоянно меняющимися видами угро.

Для эффективной борьбы с упакованными вредоносными программами крайне важно инвестировать в непрерывные исследования и разработки стратегий защиты от вредоносных программ. По мере того как авторы вредоносных программ совершенствуют свои методы

упаковки и разрабатывают индивидуальные решения, программное обеспечение для обеспечения безопасности и исследователи также должны расширять свои возможности для обнаружения, анализа и нейтрализации этих угроз.

Список использованных источников

1. Galloro N. et al. A Systematical and longitudinal study of evasive behaviors in windows malware //Computers & Security. – 2022. – Т. 113. – С. 102550.
2. Obaidat I. et al. Jadeite: A novel image-behavior-based approach for java malware detection using deep learning //Computers & Security. – 2022. – Т. 113. – С. 102547.
3. Afianian A. et al. Malware dynamic analysis evasion techniques: A survey //ACM Computing Surveys (CSUR). – 2019. – Т. 52. – №. 6. – С. 1-28.
4. Sgandurra D. et al. Automated dynamic analysis of ransomware: Benefits, limitations and use for detection //arXiv preprint arXiv:1609.03020. – 2016.
5. Raff E. et al. An investigation of byte n-gram features for malware classification //Journal of Computer Virology and Hacking Techniques. – 2018. – Т. 14. – С. 1-20.
6. Lee J., Lee B., Cho S. A Study on the Analysis Method to API Wrapping that Difficult to Normalize in the Latest Version of Themida //Journal of the Korea Institute of Information Security & Cryptology. – 2019. – Т. 29. – №. 6. – С. 1375-1382.
7. Oberhumer M. F. X. J. UPX the Ultimate Packer for eXecutables //http://upx.sourceforge.net/. – 2004.
8. Chevet S. Inside VMProtect.(2015). – 2015.

УДК 681.3

ИССЛЕДОВАНИЯ ЦИФРОВОЙ ФОРЕНЗИКИ В ИНЦИДЕНТАХ ИНФОРМАЦИОННОЙ БЕЗОПАСНОСТИ

Жаксығали Фараби Болатұлы

farabi.jax@gmail.com

Студент-магистрант ЕНУ им. Л.Н.Гумилева, Нур-Султан, Казахстан

Научный руководитель – А. М. Нурушева

Компьютерная форензика играет важную роль в расследовании и решении инцидентов информационной безопасности (*далее* – ИБ), таких как утечки конфиденциальных данных и атаки вредоносного ПО. Использование инструментов и техник компьютерной форензики значительно увеличилось в последние годы, поскольку организации стремятся защитить свои чувствительные данные и предотвратить угрозы ИБ. Эта исследовательская работа представляет собой комплексный анализ компьютерной форензики в инцидентах ИБ, включая реальные примеры, инструменты и их использование, гипотетический пример с детальным анализом, проведенным экспертами по форензике, и рекомендации по предотвращению.

Определения и терминология компьютерной форензики:

Компьютерная форензика - это процесс сбора, анализа и интерпретации электронных данных в целях выявления, документирования и сбора доказательств, связанных с преступлениями, нарушениями безопасности или другими инцидентами, связанными с использованием компьютеров, сетей и электронных устройств.

Ниже представлены некоторые из основных терминов и определений, используемых в компьютерной форензике:

1. Диск-образ: точная копия содержимого жесткого диска или другого носителя информации, которая может быть использована для анализа и восстановления данных.
2. Хеш-сумма: уникальный числовой идентификатор, созданный из содержимого файла или другого набора данных. Хеш-сумма используется для проверки целостности данных и определения, были ли они изменены.