0.1. *Лебедев Р.К., Ситнов В.Е.* Метод шифрования исполняемого кода с использованием перемещений ELF

Классическим методом противодействия исследованию программного обеспечения с закрытым исходным кодом является шифрование или упаковка машинного кода. Данный подход позволяет нейтрализовать инструменты статического анализа, так как для получения оригинального машинного кода из защищенного таким образом файла потребуются дополнительные операции или даже использование динамического анализа.

Традиционно при реализации таких преобразований код трансформируется в комбинацию из распаковщика и упакованного кода, причем распаковщик, как правило, не зависит от скрытого кода. Этот код распаковки неизбежно привлекает внимание исследователей и антивирусного программного обеспечения, зачастую считающего программу вредоносной только на основании используемого упаковщика [1].

В данной работе представлен новый метод упаковки исполняемого кода в операционной системе Linux, позволяющий исключить из зашифрованной программы код распаковщика. Такой результат достигается при помощи использования перемещений формата ELF на процессорной архитектуре x86-64, а в роли распаковщика выступает сама операционная система в процессе загрузки исполняемого файла.

Перемещения (relocations) — дополнительная информация, присутствующая во многих форматах исполняемых файлов, описывающая необходмые исправления для загрузки по адресу, отличному от того, для которого программа изначально была скомпилирована [2]. Обычно перемещения применяются к данным, например, указателям, но при помощи механизма Text Relocations при помощи них можно модифицировать и исполняемый код.

Идея предлагаемого метода состоит в выражении всего исполняемого кода программы при помощи перемещений, применяемых к секции машинного кода .text. Каждое из перемещений при обработке по одному машинному слову формирует машинный код программы, а до их обработки секция программы с кодом может содержать нули или произвольный код, предназначенный для запутывания исследователей. Были рассмотрены различные типы перемещений, а также механизм GNU_IFUNC, который получилось применить для исполнения произвольного кода в процессе обработки перемещений, что позволило реализовать различные алгоритмы шифрования.

Метод был реализован при помощи программной библиотеки LIEF языка программирования Python, позволяющей редактировать перемещения в файлах формата ELF [3]. Он был протестирован против декомпиляторов Ghidra и IDA, инструмента символьного исполнения angr, а также антивирусного ПО, доступного на сайте VirusTotal. В результате тестирования была подтверждена высокая эффективность метода против всех рассмотренных инструментов, более того, для декомпиляторов получилось добиться и ложной декомпиляции исполняемого кода.

Научный руководитель — д.т.н. Павский К. В.

Список литературы

[1] Ugarte-Pedrero X. et al. SoK: Deep packer inspection: A longitudinal study of the complexity of run-time packers // Proc. 2015 IEEE Symposium on Security and Privacy. IEEE, 2015. P. 659–673.

[2] Executable and Linkable Format 101 Part 3: Relocations. [Электронный ресурс]. URL: https://intezer.com/blog/malware-analysis/executable-and-linkable-format-101-part-3-relocations/ (дата обращения 14.09.2024).

[3] Thomas R. LIEF - Library to Instrument Executable Formats. [Электронный ресурс]. URL: https://lief.re/ (дата обращения 14.09.2024).