СЕКЦИЯ 6

УПРАВЛЕНИЕ РАЗВИТИЕМ БАНКОВСКИХ И ФИНАНСОВЫХ СИСТЕМ

ЗАЩИТА ПРОГРАММНОГО КОДА ФИНАНСОВЫХ СЕРВИСОВ МЕТОДОМ ОБФУСКАЦИИ

Беспалова Н.В.,
Финансовый университет при Правительстве Российской Федерации, Москва, Россия
NVBespalova@fa.ru
Былевский П.Г.
Московский государственный лингвистический университет, Москва, Россия
pr-911@yandex.ru

Аннотация. В ходе исследования возможностей применения методов обфускации для защиты программного кода финансовых сервисов проанализированы современные наиболее популярные решения, сформулированы критерии, выработаны способы оценки качества. Предлагается методика сравнения и выбора обфускаторов применительно к особенностям финансовой организации, финансовых сервисов и задачамобеспечения информационной безопасности.

Ключевые слова: информационная безопасность, дистанционные финансовые услуги, риски, обфускация, деобфускация, защита программного обеспечения, критерии сравнения, методы оценки качества.

Введение

Технологии искусственного интеллекта и математические методы широко используются для решения прикладных задач для приоритетных отраслей цифровой экономики. Так, в следующих работах предлагаются передовые нейросетевые и математические методы обеспечения кибербезопасности [1-8], оптимизации производственных процессов при добыче углеводородов [9], повышения эффективности сельскохозяйственного бизнеса [10]. Математические модели инновационного производства продуктов питания [11], интеллектуальной промышленной робототехники [12] и математического аппарата для задач криптографии [13-15], проблем транспорта [16,17], материаловедения [18] и финансовых рынков [19,20], находятся в центре внимания исследователей. Быстрый рост и развитие современных дистанционных финансовых сервисов сопровождается противодействием усложнению инструментария атак злоумышленников, включая использование обеими сторонами средств обфускации (маскировки кодов программного обеспечения). Обфускация приводит исполняемый программный код к виду, сохраняющему его функциональность, но затрудняющему понимание, анализ алгоритмов работы и их несанкционированную модификацию. Также обфускация с высокой эффективностью применяется, чтобы защитить код от рисков внесения злоумышленниками искажений и модификаций, но также иногда и для оптимизации программ, повышения компактности и скорости исполнения. Деобфускацию финансовые организации применяют для проверки защищённости своего программного обеспечения, особенно используемого в дистанционных сервисах в не доверенной интернет-среде, а также для анализа защищённых вредоносных программ. Существует несколько современных наиболее распространённых решений для обфускации, способные применяться для защиты и оптимизации программного обеспечения, используемого в дистанционных финансовых услугах. Критерии оценки качества важны для определения их сравнительных преимуществ и выбора наиболее оптимальных, в том числе применительно к особенностям разных финансовых организаций и их дистанционных сервисов.

1. Обзор наиболее широко применяемых обфускаторов

1. NET Reactor применим для всех языков программирования сборок .NET, используя собственную криптографическую технологию NecroBit, что является несомненным преимуществом. Присутствуют функции: маскировки названий методов, полей и программ (сборки уплотняются,

становясь более компактными, управляемые ресурсы предстают посторонним непонятными); кодируются строк (защищая паролиЮ SQL-запросы и др).

2. SmartAssembly присуще улучшение, позиционируемое разработчиками как уникальное, не встречающиеся у других известных подобных решений: непонятное постороннему видоизменение имен методов, базовых алгоритмов, строк, паролей и SQL-запросов, повышается компактность программ и сборок. Также применятся метод «родительской» обфускации, перемешивание согласно неявным правилам классификаций и описаний программ, структур классов.
3. FreeObfuscator открыт для некоммерческого использования; переназначает названия метаданных, методов, свойств, событий, полей, типов и пространств. Определить их подлинные значения можно, но исключительно при помощи специальных сигнатур.
4. CodeVeil маскирует сборки (EXE) и библиотект (DLL), а также, «насквозь», MSIL (по строкам, открытым классам, методам, свойствам и полям).
5. Goliath.NET преобразует имена методов и полей, шифрует «насквозь» все сборки, придаёт компактность и неузнаваемый вид программам, кодирует строки, маскирует пароли и SQL-запросы. За пользователем остаётся выбор статического и полиморфного вариантов обфускации.

Функции и характеристики рассматриваемых решений, применимых для обфускации программных продуктов, используемых в компьютерных системах и дистанционных сервисах финансовых организаций, сведены в Таблице 1.

Таблица 1. Сравнение характеристик и функций анализируемых обфускаторов

Название	**Стоимость**	**Control Flow**	**Шифрование MSIL**	**Преимущества**
.NET Reactor	от $180	Есть	Отсутствует	Собственная криптографиечская технология
SmartAssembly	от $795	Есть	Отсутствует	Защита сборки от изменений, соединение зависимостей, применения метода родительской обфускации
Free Obfuscator	Free	Отсутствует	Отсутствует	Бесплатный
Code Veil	от 900$	Есть	Отсутствует	Обфускатор шифрует по фрагментам сборку, записывая в себя
Goliath.NET	от 115$	Есть	Частично присутствует	Шифруется вся сборка, открыть с помощью рефлектора нельзя

Проведенное сравнение характеристики и функций проанализированных решений для обфускации программных кодов показало: в настоящее время финансовые организации испытывают высокую отребность в подобных продуктах для C# кода. Изучение статистики последних лет, отражающей инциденты информационной безопасности, связанные с несанкционированными изменениями злоумышленниками банковского софта, внедрением вредоносных кодов показывают: главными критериями выбора обфускаторов является оптимальное соотношением цены и качества. К главным параметрам качества относятся эффективность алгоритмов шифрования (стойкость к деобфускации и минимальное потребление ресурсов системы), дружественный пользовательский интерфейс и приемлемая цена.

2. Сравнение наиболее часто используемых актуальных средств обфускации

Обфускацию следует признать одним из самых популярных, распространённых и наиболее часто используемых средств защиты программного обеспечения. Использование обфускации преобразует исполняемые коды программы, сохраняя функциональность и существенно не повышая нагрузку на вычислительные мощности. Однако для постороннего, злоумышленника и потенциального нарушителя крайне затрудняется и требует намного большего времени и других ресурсов понимание кодов, алгоритмов, структуры и последовательности выполнения операций.Существенно затрудняется «разгадка», «расшифровка» рабочих кодов и порядок выполнения, поиск подходящих инструментов и специалистов, способных разобраться в проблеме.

Деобфускация представляет собой аналитическую деконструкцию «запутанных» кодов и алгоритомов для последующей реставрации исходных. Вероятность успеха, риски деобфускации определяются состязанием «щита» и «меча». В роли «щита» выступает уровень качества и надежности обфускатора, количество комплексно применяемых методов, но также компетентность

сотрудника финансовой организации, применяющего эти технические средства защиты информации. «Меч» в данном контексте – средства деобфускации и квалификация деобфускатора, тестировщика или же злоумышленника.

Разработчик и легальные пользователи программных продуктов при выборе обфускаторов руководствуются, как правило, уровнем надежности, наличием дружественного пользовательского интерфейса и риемлемой ценой, соотносимой с качеством. Основные способы обфускации классифицируемые по обрабатываемым объектам.

1. Лексическая обфускация переименовывает функции и переменные. Простата применения неожиданно оборачивается не слишком высокой эффективностью, поскольку не охватывает код и исполнение алгоритма целостно. Этот метод включает операции полного удаления или же придания неявных смыслов описаниям и комментариям в коде программы; устранения элементов, облегчающих и улучшающих визуальное восприятие программного кода. Также идентификаторов переименуются на трудночитаемые длинные символьные наборы с минимизацией смысловых ассоциаций; добавляется «белый шум», то есть избыточные операций, не нужные для выполнения, но не снижающие продуктивность программы. Лексическая обфускация меняет вид кода, имена переменных, но не порядок выполнения алгоритмов, что облегчает деобфускацию, реставрацию первоначального вида программы.
2. Обфускация данных сложнее, но результативнее лексического метода. Преобразуются, комбинировании и многократно по-новому комбинируются имеющиеся типы данных, дополнительно создаются новые. «Перепутывающая» маскировка хранилищ и типов данных, соединений усложняет представление структур используемых данных. Операция переупорядочивания видоизменяет последовательности представления и части переменных и позиционирование хранилищ данных, возрастает разнообразие способов обращения к элементам массива и разброс значений других переменных.
3. Обфускация управления маскирует самые характерные особенности защищаемой программы, запутывает структуру и последовательности управления и выполнения кода. Есть следующие виды обфускации управления, классифицируемые объекту и способу: вычислительная (меняет базовую структуру управления); соединения (объединяет-разъединяет фрагменты кода, усложняет и маскирует логику соотношений и взаимосвязей); последовательности (переупорядочивает комплексы циклов, выражений, существующие и вероятные последовательности). Итоги такой обфускации отличается от исходной программы дополнением новыми переменными и вычислительными операциями, обозначая цикл условным оператором.

3. Анализ различных способов оценки качества результатов обфускации

Экспертные методы анализа чаще всего включают три основных критерия эффективности обфускации: устойчивость, эластичность и ресурсную стоимость. Устойчивость выражает степень трудности реверсивной инженерии, деконструкцииобфусцированного кода, определяясь оценками качества кода, восстановленного «вручную» и сложность алгоритмов автоматизированной деобфускации. Эластичность характеризует соотношение применяемых методов обфускации с количеством и качеством программных средств, требующихся для встерчнойдеобфускации. Здесь нужно учитывать методы, программные продукты и другие средства деобфускации. Ресурсная стоимость определяется количеством и характеристиками системных ресурсов, которые дополнительно требуются для выполнения обфусцированных программ.

Оценка качества обфускации сильнее всего определяется параметром устойчивости. Суть его в том, что посторонний должен реставрировать исходную версию программы намного хуже, используя существенно больше ресурсов, тратя больше времени и усилий, чем если бы анализировал программу иными средствами, без доступа к коду. Для определения уровня устойчивости важна и степень доступности программы потенциальному нарушителю. Если доступ к исполняемому двоичному коду облегчён, деобфускация совершается преобразованием в описание на языке программирования более высокого уровня. Чтобы затруднить такую деобфускацию, используются структурные особенности кодов: команды перемешиваются с неисполняемыми данными и текстами описаний («белым шумом»), значительно повышая устойчивость к статическим методам дизассемблирования. Следует учитывать, что встречно заметно снизить стойкость обфускации может динамическое дизассемблирование, реставрирующая исходные коды в ходе их выполнения.

Для оценки эффективности методов обфускации с учётом циклической сложности и ветвления потока управления применяются следующие метрики сложности (числовых отображений параметров программ).

- Метрика LC - размеров процедуры, это простая метрика сложности кода. Чем сложнее и объёмнее процедура, тем выше сложность.
- Метрика YC - сложности циклической структуры, мощности транзитивного замыкания отношения достижимости посредствомграфа потока управления процедурой.
- Метрика DC - сложности потока данных выполняемой процедуры и количества уровней взаимосвязей используемых данных; означает количество замыкающих дуг графа.
- Метрика MC – циклической сложности графа и разветвления потока управления программ. Формула вычисления этой метрики усложнения программы в результате обфускации такова:

Результат, вычисляемый по формуле, показывает, насколько сложна и непонятна постороннему, как много ресурсов, усилий и времени может потребовать от тестирующего уязвимости или от злоумышленника «разгадка» обфусцированной программы в результате увеличения количества элементов кода и усложнения структуры взаимосвязей. Комплексное применение нескольких методов обфускации, маскирующих преобразований существенно затрудняет деконструкцию, реставрацию исходного кода. Также в качестве технического средства защиты программных продуктов, используемых для работы финансовых организаций и дистанционных клиентских сервисов, применяет маскировка уязвимостей преобразующим связываниеем вводимых и исходных потоков данных. Метрика циклической сложности и ветвления потока управления наиболее эффективна как средство оценки методов обфускации. Эта методика оценки включает полный анализ всех основных характеристик графа управления и необходимо требует специальных программных средств, а также участия квалифицированного эксперта в области технической защиты информации.

4. Заключение

Анализ наиболее распространённых решений, используемых для защиты программного кода методом обфускации (NET Reactor, SmartAssembly, FreeObfuscator, CodeVeil и Goliath.NET) позволил классифицировать применяемые способы: лексической обфускации, обфускации данных иобфускации управления. Сформулированы предпочтительные критерии выбора пользователем решений, организационно-экономические и качественные. К организационно-экономическим критериям относятся надежность, стоимость и «дружелюбный пользователю» интерфейс, к качественным устойчивость, эластичность и низкая вычислительные «стоимость» преобразований. Разработанная методика оценки средств обфускации рекомендуется специалистам по информационной безопасности финансовых организаций. Она может успешно применяться при выборе решений с учётом актуального перечня угроз, инструментария злоумышленников, специфики деловых процессов своей организации, перечня используемых и предоставляемых дистанционных сервисов.

Литература

1. *Osipov, A., Pleshakova, E., Liu, Y.* etal. Machine learning methods for speech emotion recognition on telecommunication systems. J ComputVirol Hack Tech (2023). https://doi.org/10.1007/s11416-023-00500-2.
2. *Tsapin, D., Pitelinskiy, K., Suvorov, S*. et al. Machine learning methods for the industrial robotic systems security. J ComputVirol Hack Tech (2023). https://doi.org/10.1007/s11416-023-00499-6.
3. *Lejun, Zhang,* et al. "Redundant data detection and deletion to meet privacy protection requirements in blockchain-based edge computing environment." China Communications 21.3 (2024): 149-159.
4. *Zhang, Lejun*, et al. "Research on Covert Communication Technology Based on Matrix Decomposition of Digital Currency Transaction Amount." KSII Transactions on Internet and Information Systems (TIIS) 18.4 (2024): 1020-1041.
5. *Ivanyuk, V*. Forecasting of digital financial crimes in Russia based on machine learning methods. J ComputVirol Hack Tech (2023). https://doi.org/10.1007/s11416-023-00480-3.
6. *Boltachev, E*. Potential cyber threats of adversarial attacks on autonomous driving models. J ComputVirol Hack Tech (2023). https://doi.org/10.1007/s11416-023-00486-x.
7. *Efanov, D., Aleksandrov, P*. &Mironov, I. Comparison of the effectiveness of cepstral coefficients for Russian speech synthesis detection. J ComputVirol Hack Tech (2023). https://doi.org/10.1007/s11416-023-00491-0.
8. *Mizinov, P.V., Konnova, N.S., Basarab, M.A*. et al. Parametric study of hand dorsal vein biometric recognition vulnerability to spoofing attacks. J ComputVirol Hack Tech (2023). https://doi.org/10.1007/s11416-023-00492-z.
9. *Osipov A.* et al., "Machine Learning Methods Based on Geophysical Monitoring Data in Low Time Delay Mode for Drilling Optimization," in IEEE Access, vol. 11, pp. 60349-60364, 2023, doi:10.1109/ACCESS.2023.3284030.
10. *Andriyanov, N.; Khasanshin, I.; Utkin, D.; Gataullin, T.; Ignar, S.; Shumaev, V.; Soloviev, V.* Intelligent System for Estimation of the Spatial Position of Apples Based on YOLOv3 and Real Sense Depth Camera D415. Symmetry 2022, 14, 148. https://doi.org/10.3390/sym14010148.

11. *Timofeev, I.; Pleshakova, E.; Dogadina, E.; Osipov, A.; Kochkarov, A.; Ignar, S.; Suvorov, S.; Gataullin, S.; Korchagin, S.* Mathematical Models and Methods for Research and Optimization of Protein Extraction Processes from Chickpea and Curd Whey Solutions by Electroflotation Coagulation Method. Mathematics 2022, 10, 1284. https://doi.org/10.3390/math10081284.
12. *Krakhmalev, O.; Korchagin, S.; Pleshakova, E.; Nikitin, P.; Tsibizova, O.; Sycheva, I.; Liang, K.; Serdechnyy, D.; Gataullin, S.; Krakhmalev, N*. Parallel Computational Algorithm for Object-Oriented Modeling of Manipulation Robots. Mathematics 2021, 9, 2886. https://doi.org/10.3390/math9222886.
13. *Barotov, D.; Osipov, A.; Korchagin, S.; Pleshakova, E.; Muzafarov, D.; Barotov, R.; Serdechnyy, D.* Transformation Method for Solving System of Boolean Algebraic Equations. Mathematics 2021, 9, 3299. https://doi.org/10.3390/math9243299.
14. *Zhang, J.; Kang, X.; Liu, Y.; Ma, H.; Li, T.; Ma, Z.*; *Gataullin, S*. A Secure and Lightweight Multi-Party Private Intersection-Sum Scheme over a Symmetric Cryptosystem. Symmetry 2023, 15, 319. https://doi.org/10.3390/sym15020319.
15. *Gataullin, S. T., &Gataullin, T. M.* (2023). To the Problem of a Point Source in an Inhomogeneous Medium. Mathematical Notes, 114(5), 1212-1216.
16. *Kositzyn, A.; Serdechnyy, D.; Korchagin, S.; Pleshakova, E.; Nikitin, P.; Kurileva, N*. Mathematical Modeling, Analysis and Evaluation of the Complexity of Flight Paths of Groups of Unmanned Aerial Vehicles in Aviation and Transport Systems. Mathematics 2021, 9, 2171. https://doi.org/10.3390/math9172171.
17. *Gataullin, T. M., &Gataullin, S. T.* (2022). Endpoint functions: mathematical apparatus and economic applications. Mathematical Notes, 112(5), 656-663.
18. *Korchagin, S.; Pleshakova, E.; Alexandrova, I.; Dolgov, V.; Dogadina, E.; Serdechnyy, D.; Bublikov, K.* Mathematical Modeling of Electrical Conductivity of Anisotropic Nanocomposite with Periodic Structure. Mathematics 2021, 9, 2948. https://doi.org/10.3390/math9222948.
19. *Gataullin, T.M., Gataullin, S.T., Ivanova, K.V.* (2021). Modeling an Electronic Auction. In: Popkova, E.G., Sergi, B.S. (eds) "Smart Technologies" for Society, State and Economy. ISC 2020. Lecture Notes in Networks and Systems, vol 155. Springer, Cham. https://doi.org/10.1007/978-3-030-59126-7_122.
20. *GataullinT. M., GataullinS. T. and IvanovaK. V.*, "Synergetic Effects in Game Theory," 2020 13th International Conference "Management of large-scale system development" (MLSD), Moscow, Russia, 2020, pp. 1-5, doi: 10.1109/MLSD49919.2020.9247673.