ИСТОЧНИКИ И УЯЗВИМОСТИ НЕСАНКЦИОНИРОВАННОГО ДОСТУПА К ПРОГРАММНОМУ ОБЕСПЕЧЕНИЮ

Кистерная А.В., студентка гр.241301

Белорусский государственный университет информатики и радиоэлектроники[1]
г. Минск, Республика Беларусь

Дворникова Т.Н. – магистр тех. наук

Аннотация. В работе проведен комплексный анализ источников и классификация угроз несанкционированного доступа к программному обеспечению. Рассмотрены уязвимости системного и прикладного ПО, механизмы их эксплуатации через сетевые протоколы и локальные интерфейсы. Выделены ключевые векторы атак, связанные с программно-математическим воздействием, включая классификацию вредоносных программ по среде обитания и способам внедрения в код. Особое внимание уделено угрозам непосредственного и удаленного доступа к исполняемым файлам, библиотекам и конфигурационным данным. Результаты исследования могут быть использованы при аудите защищенности ПО, разработке безопасного кода и выборе мер защиты в соответствии с требованиями регуляторов.

Ключевые слова. программное обеспечение, несанкционированный доступ, уязвимости кода, вредоносное программное обеспечение, сетевые протоколы, классификация угроз, безопасная разработка, защита данных.

В условиях повсеместной цифровизации и роста сложности программных продуктов обеспечение безопасности программного обеспечения (ПО) приобретает критическую значимость. Угрозы несанкционированного доступа (НСД) к ПО представляют собой совокупность действий, направленных на нарушение конфиденциальности, целостности или доступности программного кода, конфигурационных файлов и обрабатываемых данных. Реализация данных угроз возможна как в результате преднамеренных атак злоумышленников, так и вследствие ошибок разработки, некорректной настройки или случайных действий персонала [1].

Классификация угроз НСД к программному обеспечению включает три базовые категории. Первая — угрозы проникновения в исполняемую среду с использованием штатных функций ПО, включая эксплуатацию легитимных интерфейсов прикладных программ и системных библиотек. Вторая — угрозы создания нештатных режимов работы программного кода (DoS/DDoS), реализуемые путем переполнения буферов, передачи некорректных входных параметров или исчерпания вычислительных ресурсов. Третья — угрозы программно-математического воздействия, связанные с внедрением и активацией вредоносного кода в теле легитимного ПО [2]. Комбинированные атаки, сочетающие несколько векторов эксплуатации уязвимостей, представляют наибольшую опасность, так как позволяют обходить многоуровневые механизмы защиты программного кода.

Источниками угроз к ПО выступают нарушители, носители вредоносных программ и программные закладки. Нарушители подразделяются на внешних (атакующих через сетевые интерфейсы ПО) и внутренних (имеющих доступ к исходному коду, репозиториям или среде выполнения). Внутренние нарушители классифицируются по уровню привилегий в рамках работы с ПО: от рядовых пользователей приложений до разработчиков, тестировщиков и администраторов развертывания. Каждый тип нарушителя обладает специфическим набором возможностей воздействия на программный код, что необходимо учитывать при моделировании угроз. Классификация источников угроз с учетом категорий доступа к ПО и потенциальных вектов атак представлена в таблице 1.

Таблица 1 – Классификация источников угроз НСД к программному обеспечению

Категория источника	Тип/группа	Характеристика возможностей
Нарушитель (внешний)	Сетевые интерфейсы ПО, API	Сканирование портов сервисов, эксплуатация уязвимостей веб-приложений, инъекции кода (SQL, XSS), fuzzing-атаки
Нарушитель (внутренний)	Пользователи прикладного ПО	Копирование исполняемых файлов, модификация конфигураций, установка плагинов с вредоносным кодом
	Разработчики и тестировщики	Внедрение уязвимого кода, оставление отладочных функций, компрометация библиотек зависимостей
	Администраторы развертывания	Модификация политик выполнения кода, отключение средств контроля целостности ПО, анализ логов в обход аудита

Носители ВП	Отчуждаемые/встроенные носители	Инсталляция вредоносного кода через USB, сетевые пакеты, файлы данных

В рамках классификации уязвимостей программного обеспечения целесообразно выделять ошибки управления памятью, логические ошибки и ошибки конфигурации. К наиболее критическим относятся уязвимости переполнения буфера, позволяющие выполнить произвольный код в контексте процесса. Такие ошибки характерны для языков программирования без автоматического управления памятью (C, C++), где отсутствует проверка границ массивов. Также значительную долю составляют уязвимости веб-приложений, такие как инъекции (SQL, Command Injection) и межсайтовый скриптинг (XSS), связанные с недостаточной фильтрацией входных данных и доверием к пользовательскому вводу [3].

Согласно данным Common Weakness Enumeration (CWE), большинство уязвимостей обусловлено человеческим фактором на этапе написания кода. Стремление сократить время выхода продукта на рынок (time-to-market) часто приводит к пренебрежению процедурами безопасного кодирования и тестирования на проникновение. Дополнительно следует отметить уязвимости, связанные с зависимостями программного обеспечения. Современные приложения активно используют сторонние библиотеки и модули, что расширяет поверхность атаки. Компрометация одной библиотеки может привести к заражению множества программных продуктов (атаки на цепочку поставок). Поэтому анализ уязвимостей должен включать не только собственный код, но и весь стек зависимостей.

Последствия эксплуатации уязвимостей ПО варьируются от утечки конфиденциальной информации до полного захвата контроля над системой. В критических инфраструктурах это может привести к физическим повреждениям или остановке технологических процессов. Поэтому выявление уязвимостей на ранних этапах жизненного цикла разработки ПО является приоритетной задачей. Особую группу составляют уязвимости, связанные с реализацией сетевых протоколов в программном коде приложений. Активное использование стека TCP/IP в программных интерфейсах создает риски перехвата данных, подмены вызовов функций или организации отказов в обслуживании на уровне приложения. Детальная характеристика уязвимостей отдельных протоколов с точки зрения программной реализации приведена в таблице 2.

Таблица 2 – Уязвимости программной реализации протоколов стека TCP/IP

Протокол	Уровень OSI	Тип уязвимости	Последствия реализации
FTP, Telnet	Прикладной	Передача учетных данных в открытом виде, отсутствие шифрования сессии	Сниффинг трафика приложения, перехват сессионных токенов, получение НСД к функциям ПО
UDP	Транспортный	Отсутствие контроля целостности пакетов в обработчике	Подделка исходного адреса (spoofing), переполнение буфера приемника, сбой в работе сервиса
ARP	Канальный	Отсутствие аутентификации ответов в программном обработчике	ARP-spoofing, перехват межпроцессного взаимодействия, модификация передаваемых данных
TCP	Транспортный	Предсказуемость SEQ-номеров, уязвимость обработчика к SYN-флуд	Подмена сессий приложения (hijacking), исчерпание потоков обработки, отказ сервиса
DNS	Прикладной	Отсутствие проверки подлинности ответов в клиентском коде	DNS cache poisoning, перенаправление вызовов ПО на вредоносные ресурсы, загрузка вредоносных модулей

Угрозы непосредственного доступа к ПО реализуются при физическом или логическом контакте с исполняемой средой и направлены на модификацию исполняемых файлов, внедрение кода в память процесса, перехват системных вызовов или подключение программных кейлоггеров. Сетевые угрозы, реализуемые дистанционно, включают пассивный анализ сетевого трафика приложения, активное сканирование открытых портов сервисов, подмену доверенных компонентов (DLL hijacking) и внедрение вредоносных модулей через уязвимые сетевые библиотеки [4].

Программно-математические воздействия осуществляются посредством вредоносных программ (ВП), обладающих способностью к саморепликации в файловых структурах, сокрытию присутствия в процессах ОС и выполнению деструктивных функций в контексте легитимного ПО.

Современные вредоносные программы часто используют техники обфускации кода, полиморфизм и динамическую загрузку модулей, что затрудняет их статический анализ и обнаружение традиционными антивирусными средствами. Особую опасность представляют целевые атаки на цепочку поставки ПО (software supply chain), в которых компрометация инструментов сборки, репозиториев зависимостей или

систем обновления позволяет злоумышленникам массово распространять вредоносный код под видом легитимных обновлений.

Для нейтрализации рассмотренных угроз на уровне программного обеспечения рекомендуется применять комплексный подход, включающий: – меры безопасной разработки (Secure SDLC): статический и динамический анализ кода (SAST/DAST), ревью кода, использование безопасных библиотек; – технические средства защиты ПО: контроль целостности исполняемых файлов, песочницы (sandboxing) для изоляции подозрительного кода, механизмы контроля выполнения (DEP, ASLR); – организационные практики: регламентация доступа к репозиториям, верификация цифровых подписей обновлений, обучение разработчиков принципам безопасного кодирования; – мониторинг и реагирование: логирование критических событий выполнения ПО, автоматизированное обнаружение аномального поведения процессов, оперативное применение патчей [6].

Таким образом, детальный анализ источников, уязвимостей и методов реализации угроз НСД к программному обеспечению позволяет сформировать обоснованную модель угроз для конкретного программного продукта. Интеграция результатов классификации в процесс проектирования и разработки ПО обеспечивает выбор адекватных контрмер, направленных на минимизацию рисков нарушения конфиденциальности, целостности и доступности программного кода и обрабатываемых данных в соответствии с требованиями национального и международного законодательства в области информационной безопасности.

Список использованных источников:

1. Вострецова, Е. В. Основы информационной безопасности : учеб. пособие для вузов / Е. В. Вострецова. – Екатеринбург : Изд-во Урал. ун-та, 2019. – 204 с.

2. Базовая модель угроз безопасности персональных данных при их обработке в информационных системах персональных данных : утв. ФСТЭК России 15 февр. 2008 г.

3. Климентьев, К. Е. Компьютерные вирусы и антивирусы. Взгляд программиста / К. Е. Климентьев. – М. : ДМК Пресс, 2013. – 656 с.

4.Лукацкий, А. В. атак / А. В. Лукацкий. – СПб. : БХВ-Петербург, 2001. – 624 с.

5. Пулко, Т. А. Введение в информационную безопасность : учеб. пособие / Т. А. Пулко. – Минск : БГУИР, 2018. – 156 с.

6.Михеева, Е. В. Информационные технологии в профессиональной деятельности / Е. В. Михеева. – М. : Академия, 2008. – 336 с.