Данная методика была применена для создания системы учета и контроля ядерных материалов ACCORD-2005, которая сертифицирована ФСТЭК по требованиям безопасности и допускает обработку информации, содержащей государственную тайну.

СПИСОК ЛИТЕРАТУРЫ:

1. Требования по защите от несанкционированного доступа к информации в автоматизированных системах учета и контроля ядерных материалов. Гостехкомиссия России, Министерство Российской Федерации по атомной энергии. М., 1997.
2. *Анищенко А. А., Иванов К. В.* Разграничение доступа в автоматизированных системах учета и контроля ядерных материалов // Вопросы атомной науки и техники. 2008. № 2. С. 66–70.
3. *Румянцев А. Н.* От учета и контроля к управлению // Новости ФИС. 2004. № 5. С. 39–48.
4. *Федосеев В. Н., Мизин П. П., Шанин О. И.* Подход к программному обеспечению для российских СУиК следующего поколения // Новости ФИС. 2003. № 3. С. 21–30.
5. *Щеглов А. Ю.* Защита компьютерной информации от несанкционированного доступа. СПб.: Наука и Техника, 2004. – 384 с.

С. Е. Кириллов, Н. П. Лаврентьев

ПОИСК УЯЗВИМОСТЕЙ СЕТЕВЫХ ПРИЛОЖЕНИЙ МЕТОДОМ ПСЕВДОСЛУЧАЙНОГО НАГРУЗОЧНОГО ТЕСТИРОВАНИЯ

Осуществление нормального обмена информацией посредством сети является критическим для существования современного общества, что делает актуальной задачу тестирования безопасности сетевых приложений. Большая часть угроз информационной безопасности сетевых приложений связана с наличием уязвимостей, вызванных ошибками при разработке и кодировании. Для поиска ошибок широко применяется методика псевдослучайного нагрузочного тестирования. При псевдослучайном нагрузочном тестировании на вход программы подаются специальным образом сформированные входные данные и отслеживается корректность выполнения программы на этих входных данных. Известны две разновидности такого тестирования [1, 2]: мутационное тестирование и генерационное тестирование. При мутационном тестировании в валидные входные данные внедряются искажения, такие как вставки, замены и удаления блоков информации. Генерационное тестирование подразумевает использование модели данных для генерации множества тестовых данных. В общем случае оба этих подхода имеют большое количество тестовых входных данных. Для сокращения пространства возможных входных данных при тестировании безопасности сетевых приложений необходимо знание достаточно точной модели форматов данных исследуемого приложения. Множество форматов сетевых сообщений задается в формализованном описании сетевого протокола. Для закрытых протоколов создание модели входных данных (определение структуры сообщений/сессий) требует обратной разработки, которая, если проводить анализ вручную, в ряде случаев может иметь высокую сложность.

В данной работе проведена доработка ранее разработанного алгоритма восстановления сессий, а также приведен алгоритм восстановления структуры сессий исследуемого сетевого протокола, идейно продолжающий предложенный в [3] подход. На вход алгоритма поступает собранная трасса валидных взаимодействий в рамках исследуемого протокола. Первоначально из трассы выделяются отдельные сообщения, в которых ищутся текстовые, и выполняется первичная

кластеризация сообщений. После этого выполняется фаза рекурсивной кластеризации, в которой происходит уточнение форматов и разбиение кластеров согласно обновленным форматам. После этого выполняется фаза слияния кластеров, в которой выделяются истинные форматы сообщений путем слияния в один кластеров, близких по форматам сообщений. Полученные результаты планируется использовать в системе псевдослучайного нагрузочного тестирования сетевых приложений. Данный алгоритм изображен на рис. 1.

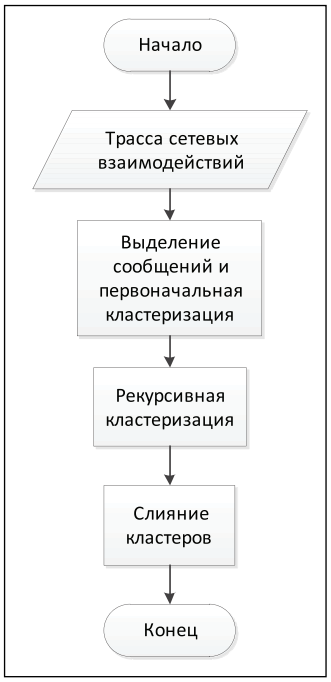

Рис. 1. Алгоритм восстановления формата сетевых сообщений

Разработанный алгоритм восстановления формата сетевых сообщений на выходе дает модель сетевого сообщения, позволяющую генерировать валидные сетевые сообщения. Модель данных содержит в себе множество форматов данных, каждый формат имеет определенное количество полей, каждое поле имеет свое множество значений, тип и семантику. Полученная модель данных будет использоваться в разрабатываемой автором системе поиска уязвимостей сетевых приложений.

СПИСОК ЛИТЕРАТУРЫ:

1. *Макаров А. Н.* Метод автоматизированного поиска программных ошибок в алгоритмах обработки сложноструктурированных данных // Прикладная дискретная математика. 2009. № 3. С. 117–227.
2. Automated whitebox fuzz testing. URL: http://research.microsoft.com/en-us/projects/atg/ndss2008.pdf (дата обращения: 31.03.2012).
3. Tupni: Automatic Reverse Engineering of Input Formats. URL: http://research.microsoft.com/pubs/101326/tupni-ccs08.pdf (дата обращения: 15.02.2012).