Основными преимуществами технологии являются её дешевизна и возможность автоматизации при высокой вероятности выявления уязвимостей. По данным [2] в цикле разработки компании Microsoft до 25% ошибок и уязвимостей обнаруживаются при помощи фаззинга. Другие технологии, такие как статический анализ, являются более дорогими и трудоёмкими.

Фаззинг применим для тестирования широкого круга приложений: обрабатывающих файлы определённых форматов, работающих через интерфейс командной строки, библиотек, клиент-серверных приложений. Как правило, для проведения тестирования требуется заранее знать, структуру данных, которые необходимо смоделировать. Получение информации о структуре данных во время выполнения является нетривиальной задачей.

В докладе рассматриваются два инструмента, наиболее эффективно реализующие все этапы фаззинга: Peach [3] и Sulley [4].

В докладе подробно рассматриваются следующие вопросы:

- Проводится аналитический обзор технологии фаззинга, классификация типов фаззинга, обзор основных задач, решаемых с его помощью, основных приёмов применяемых при этом.
- Ставятся эксперименты при помощи фреймворков Peach и Sulley.
- Предлагаются дополнительные алгоритмы, применимые для генерации входных данных.
- Предлагаются алгоритмы для разбора форматов данных во время выполнения.

Литература

1. P. Sutton, A. Greene, P. Amini. Fuzzing. Brute force vulnerability discovery. Pearson Education, 2007.
2. M. Howard, S. Lipner. The Security Development Lifecycle: SDL: A Process for Developing Demonstrably More Secure Software. Microsoft Press, 2006.
3. http://peachfuzzer.com/
4. http://code.google.com/p/sulley/

МЕТОД СИНГУЛЯРНОГО СПЕКТРАЛЬНОГО АНАЛИЗА ДЛЯ ВЫХОДНЫХ ПОСЛЕДОВАТЕЛЬНОСТЕЙ ШИФРОВ ГАММИРОВАНИЯ НА ОСНОВЕ ДИНАМИЧЕСКОГО ХАОСА

Сидоренко А. В., Шакинко И. В.

БГУ, Минск, Беларусь, e-mail: SidorenkoA@yandex.ru

Современные информационные технологии находят широкое применение в телекоммуникационных системах. Существенную роль при этом играют вопросы защиты информации. Созданию хаотических систем для защиты информации способствовали успехи, достигнутые при разработке систем динамического хаоса и появление хаотических отображений. Использование динамического хаоса для хаотических систем защиты информации обусловлено способностью хаотических отображений обеспечивать скрытость передачи зашифрованной информации. Детерминизм хаоса способствует шифрованию информации, а его случайность делает систему стойкой к вскрытию.

В данной работе проводится определение показателей выходных последовательностей шифров гаммирования на основе динамического хаоса с использованием метода сингулярного спектрального анализа. Определяется уровень главных компонент, меры отклонения от равномерного распределения. Визуализация информации производится в виде фазовых диаграмм, в которых по осям координат откладываются различные пары собственных векторов или главных компонент.

Вычислительный эксперимент проводился при использовании шифра гаммирования на основе динамического хаоса. Схема алгоритма – сеть Фейстеля, отображение пилообразное, число итераций z изменялось от одной до 1024. Длина анализируемых последовательностей $N = 10000$, длина гусеницы $M = 1000$. Результаты расчета уровня главных компонент I для открытого текста (1), шифра гаммирования на основе динамического хаоса при числах итераций $z = 8$ (2), $z = 64$ (3), а также шифров des в режиме cbc (4) и aes в режиме cbc (5) приведены в таблице

Табл. 1. Уровень главных компонент I, открытого текста (откр. текст), выходных последовательностей шифра гаммирования на основе динамического хаоса при числе итераций $z = 8$, $z = 64$, *шифра des в режиме cbc(des_cbc)*

Ном.гл.комп.	1000	999	998	997	996	995	994	993
1	0,3530	0,3530	0,3036	0,3035	0,2989	0,2979	0,2934	0,2934
2	0,2203	0,2202	0,2122	0,2118	0,2081	0,2080	0,2028	0,2027
3	0,226	0,2258	0,2250	0,2250	0,2109	0,2108	0,2015	0,2014
4	0,1899	0,1893	0,1880	0,1879	0,1841	0,1840	0,1821	0,1820
5	0,2287	0,2287	0,2179	0,2177	0,2135	0,2134	0,2083	0,2081

Как видно из таблицы 1, для открытого текста уровень главных компонент I (1) превышает значения для показателей исследуемых шифров (2-5), в среднем, на 50-60 процентов. Сравнительный анализ показателей, полученных для выходных последовательностей шифра гаммирования на основе динамического хаоса, практически совпадает с показателями, реализуемых шифрами des в режиме cbc и aes в режиме cbc.

ПЕРЕПОЛНЕНИЕ БУФЕРА: ПРОШЛОЕ, НАСТОЯЩЕЕ И БУДУЩЕЕ

Фомчин Д. О.

БГУ, НИИ ППМИ, Минск, Беларусь, e-mail: dimafomchin@gmail.com

Переполнение буфера по праву считается одной из самых опасных уязвимостей. При эксплуатировании появляется практически неограниченный доступ к атакуемой системе. Переполнение буфера известно ещё с самого появления компьютеров, и вот теперь, сквозь десятилетия, эта проблема не решена полностью.

Всего в базе бюллетеней по безопасности Microsoft около 1000 записей. Большинство из отмеченных уязвимостей относятся к переполнению буфера. Самые опасные – это те, которые позволяют удалённо выполнить произвольный код без участия пользователя, т.е. не требуется открывать файлы или заходить на какой-либо интернет-ресурс. Вот одни из этих уязвимостей: