УДК 004.9

СКАНИРОВАНИЕ ИСХОДНЫХ КОДОВ НА УЯЗВИМОСТИ

Студент гр. 113026 Г.А. Васильков,
преподаватель И.Л. Алифанова, ст. преподаватель О.В. Дубровина

Белорусский национальный технический университет

В современном мире развивающиеся технологии должны уделять большое внимание защите данных.Для обеспечения защищённости и целостности системы необходимо непрерывное сканирование ее состояния. Анализ исходного кода программ часто оказывается самым эффективным способом выявления новых уязвимостей.

Мы предлагаем сканер уязвимостей Bagur. Это приложение сканирует файлы исходного кода на языке программирования C\C++. На современном этапе этот язык служит для написания операционных систем, на нем написаны Unix\Linux, FreeBSD, OpenBSD, Windows и др.

Приложение обладает привычным, удобным и интуитивно понятным интерфейсом. По команде открытия файла из стандартного диалога загружается файл исходного кода программы на языке C\C++. Можно осуществлять поиск файлов из приложения.

В первую очередь Bagur производит поиск определённых функций, которые при вызове не безопасны или потенциально не безопасны. К таковым мы отнесли функции создания процесса, копирования, выделение памяти: CreateProcess, strcopy (StrCopy,wcscpy,_mbscpy), setbuf, gets, malloc, new, free, delete, sprintf.

Следующим шагом сканер пытается установить, есть ли возможные уязвимости, связанные с найденными функциями.

На втором этапе сканер пытается найти начало и конец параметров функции и сравнить их со значениями, хранящимся в каждой из потенциально небезопасных функций. При обнаружении совпадения значение параметра заносится в таблицу, в которой фиксируются найденные уязвимости. Структура таблицы, выводимой в качестве результата исследования кода, следующая (по колонкам слева направо):

- классификатор дефекта;
- коэффициент опасности, определяемый для данного вида дефекта (локальный, удалённый и т.д.);
- категория, к которой относится уязвимость;
- локация обнаруженной уязвимости - номер или номера строк в коде;
- полный путь к отсканированному файлу;
- время сканирования;
- дата сканирования.

На рисунке показана уже сформировавшаяся таблица.

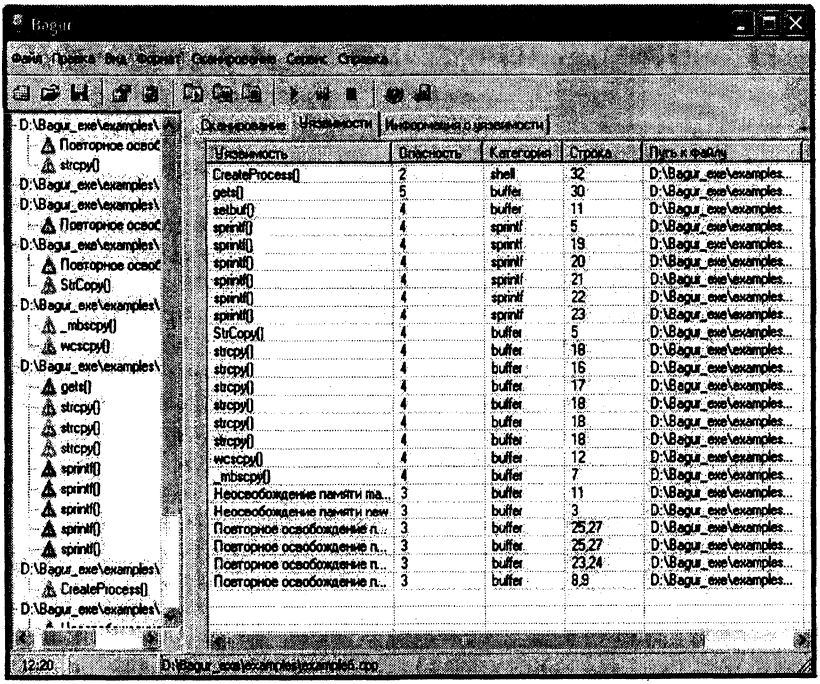

Рис. 1. Список найденных уязвимостей

Естественно, конечным итогом работы должно быть полное устранение слабых к атакам мест кода. Для нейтрализации опасности программист должен получить максимально полную информацию о «дыре» в приложении.

Вспомогательная информация, даваемая сканером по каждому из известных ему дефектов, поможет программисту локализовать проблему. Сканер предлагает сведения об уязвимости, полное её описание и методы борьбы с ней.

Уязвимости, найденные сканером, относятся к различным категориям, следовательно, имеют различные возможности для эксплуатации. Часть из них приводят к нарушению безопасности как при удалённом, так локальном доступе. Другие же, например CreateProcess(), могут привести к возникновению опасности только при локальной работе.

Анализ уязвимостей в большинстве случаев трудоемкий, рутинный и малопроизводительный процесс. К результату ведет только последовательный, скрупулезный анализ с многочасовыми экспериментами и чтением документации.