АУДИТ БЕЗОПАСНОСТИ В ИНФОРМАЦИОННЫХ СИСТЕМАХ

Т.Р. Колесова, Е.С. Белоусова

Для защиты информации в информационных системах необходимо проводить постоянный аудит безопасности. В настоящее время приказом Оперативно-аналитического центра при Президенте Республики Беларусь № 64 от 11 октября 2017 г. предъявляются следующие требования по обеспечению аудита безопасности в информационных системах: определение состава и содержания информации о событиях безопасности, подлежащих регистрации; сбор, запись и хранение информации о событиях безопасности в течение установленного срока хранения, но не менее шести месяцев; мониторинг (просмотр, анализ) информации о сбоях в механизмах сбора информации и о достижении предела объема (емкости) памяти устройств хранения уполномоченными пользователями; осуществление мониторинга (просмотра, анализа) событий безопасности уполномоченными субъектами информационной системы.

Для реализации данных требований можно использовать системы сбора и обработки данных событий информационной безопасности. В настоящее время на территории Республики Беларусь сертифицированы следующие системы: ArcSight Enterprise Security Manager, IBM Security Qradar SIEM, AccelOps, FortiAnalyzer, Efros Config Inspector. На мировом рынке наиболее популярны первые две системы.

ArcSight ESM поддерживает широкий перечень разнообразных источников событий, выполняет нормализацию на очень высоком уровне, имеет широкие возможности тонкой отладки, кастомизации и мощный корреляционный функционал. Однако присутствуют и минусы - сложность первичного изучения продукта и его высокая стоимость. IBM Security Qradar SIEM имеет более слабый корреляционный функционал, его возможности тонкой отладки и кастомизации ограничены, но зато он обладает более простым и понятным интерфейсом. ArcSight ESM необходимо использовать крупным организациям ввиду его гибкости и богатого функционала. IBM Security Qradar SIEM рационально использовать организациям поменьше, которые не готовы заниматься тонкой настройкой системы и которым не нужна специфичная гибкость платформы для реализации основных функций.

Литература

1. О внесении изменений в некоторые приказы Оперативно-аналитического центра при Президенте Республики Беларусь [Электронный ресурс]. – URL: http://pravo.by/upload/docs/op/T61703911_1507928400.pdf (дата обращения: 14.05.2018).

ОБЕСПЕЧЕНИЕ БЕЗОПАСНОСТИ ВЕБ-ПРИЛОЖЕНИЙ НА СТОРОНЕ БРАУЗЕРА ПОЛЬЗОВАТЕЛЯ

Е.А. Криштопова, О.С. Медведев, А.В. Кистюк

По многочисленным исследованиям более 80 % веб-сайтов содержат критические уязвимости, вероятность автоматизированного заражения страниц уязвимого веб-приложения вредоносным кодом составляет сегодня приблизительно 15–20 %.

Обеспечение безопасности веб-приложений – это объемный процесс, который включает выполнение действий как на уровне заинтересованных сторон, так и на уровне отдельных компонентов системы функционирования веб-приложения. Остановимся на аспектах обеспечения безопасности веб-приложения на уровне компьютера и браузера пользователя.

Так как на компьютер и браузер пользователя данные для веб-приложения поступают из недоверенного источника, то обеспечить их полноценную безопасности невозможно. В этом случае средствами обеспечения безопасности на стороне браузера являются применение правила ограничения домена, использование песочницы (Sandbox) для тестирования и работы с недоверенными или разрабатываемыми приложениями, включение блокировки вредоносных сайтов, применение механизмов аутентификации, включение изоляции вкладок, обеспечение безопасности и смягчения угроз использования компонентов веб-приложения (например, Flash, Java и др.) и дополнений.

Литература

1. The Ten Most Critical Web Application Security Risks [Электронный ресурс]. – URL: https://www.owasp.org/images/7/72/OWASP_Top_10-2017_en.pdf (дата обращения: 11.05.2018).

2. Евсеев Д. Введение в тему безопасности веб-приложений [Электронный ресурс]. – URL: https://www.ptsecurity.com/upload/corporate/ru-ru/webinars/ics/Д.Евтеев_Введение_в_тему_безоп_веб_прилож.pdf. (дата обращения: 11.05.2018).

КОНТРОЛЬ БЕЗОПАСНОСТИ В КЛИЕНТ-СЕРВЕРНЫХ ПРИЛОЖЕНИЯХ ПРИ ИСПОЛЬЗОВАНИИ ФРЕЙМВОРКА MICROSOFT SIGNALR НА ОСНОВЕ КЛИЕНТСКИХ ГРУПП

В.В. Кузнецов

При разработке клиент-серверных приложений в системах информационной безопасности острой ставится задача запросов клиента к серверу, особенно учитывая проблемы связанные с несанкционированным доступом в базу данных, отправку специфических аргументов серверу и других.

Целью настоящей работы явилась разработка подхода контроля безопасности запросов клиентов к серверу.

Фреймворк Microsoft SignalR [1], обеспечивающий двустороннее взаимодействие в клиент-серверных web-приложениях позволяет на стороне сервера (backend-side) применять атрибуты клиентских групп для классов, обеспечивающих API (Application programming interface) клиентов к серверу, в результате чего исключается необходимость проверять права клиента при обработке запросов. Клиентские группы, такие как например «Пользователи» и «Администраторы» хранятся на машине (ПК) выполняющего функции сервера в разделе «Управление компьютером/Локальные пользователи и группы/Группы». Добавив соответствующие группы, предоставляется возможность их использования в добавлении аттрибутов, например [Authorize(“Custom Administrators group”)], после чего доступ к соответствующему классу или методу получат только те пользователи, имена (домены) которых включены в соответствующие группы.

Таким образом, разработан подход по контролю безопасности клиент-серверных веб-приложений при использовании фреймворка Microsoft SignalR с технологией применений атрибутов для определенных клиентских групп, указанных на серверной машине с операционной системой Windows.

Литература

1. SignalR [Электронный ресурс]. – URL: https://docs.microsoft.com/en-us/aspnet/signalr (дата обращения: 16.05.2018).

УЯЗВИМОСТИ ПОВРЕЖДЕНИЯ ПАМЯТИ В УСТРОЙСТВАХ «INTERNET OF THINGS»

В.Ф. Кулиш

Устройства «интернета вещей» используют широкий диапазон архитекур центрального процессора, но наибольшее распространение получили архитектуры ARM и MIPS. Использование таких архитектур обусловлено их низким энергопотреблением и, соответсвенно, низким количеством выделяемого тепла при работе. Однако использование таких архитектур не защищает от уязвимостей повреждения памяти. Также как и устройства с архитектурой x86, такие устройства также подвержены уязвимостям переполнения буфера и уязвимостям форматных строк.

Уязвимости переполнения буфера обнаружились еще в начале компьютерной эпохи и продолжают существовать по сей день. Уязвимостям такого типа подвержены языки программирования, в которых управление процессом выделения памяти отдано на откуп программисту (пример: C, C++). При создании программ разработчику необходимо контролировать размер помещаемых в переменную данных, память под которую была