Пусть Alyz — граф потока управления анализируемой программы, состоит из n вершин, a_q — q-я вершина в Alyz, $q=\overline{1,n}$, Sig — граф потока управления поведенческой сигнатуры, состоит из m вершин, s_g — какая-либо конкретная вершина графа поведенческой сигнатуры, $g=\overline{1,m}$, $n<m$. Пусть a_q, s_g состоят из k и l — стандартных функций языка JavaScript, входящих в каждый конкретный список функций вершины; ρ, n, m, k, $l \in N$, и пусть гомоморфизм рассматривается как функция Ψ, которая отображает Sig в Alyz, такая, что если в графе Sig существует путь, соединяющий вершины s_i и s_j, то и в графе Alyz существует путь, соединяющий вершины $\Psi(s_i)$ и $\Psi(s_j)$.

Пусть слова wa_q и ws_g определены как объединения всех символов (которыми являются функции языка JavaScript), входящих в эти слова.

Пусть ρsydr — вероятность существования гомоморфизма Ψ графа Sig в граф Alyz.

cros(wa_q,ws_g) — функция «схожести» слов wa_q и ws_g, принимающая значения «0» или «1», DLD — функция, вычисляемая на основании алгоритма Дамерау—Левенштейна для всех вершин графа потока управления анализируемой программы и графа потока управления поведенческой сигнатуры [4].

В случае, если существует f(ρsydr,m) вершин $a_q \in$ Alyz, для которых существует $s_g \in$ Sig, таких, что cros(wa_q,ws_g) = 1, то принимается решение о проверке существования гомоморфизма Ψ графа Sig в граф Alyz. Все обнаруженные вершины a_q с cros(wa_q,ws_g) = 1 в Alyz помечаются. На основании алгоритма «поиска простого пути» [5] проверяется, есть ли путь в Alyz, позволяющий соединить вершины a_q графа Alyz с cros(wa_q,ws_g) = 1 в той же последовательности, что и вершины s_g в графе Sig. В случае существования подобного пути принимается решение, что код вредоносен.

СПИСОК ЛИТЕРАТУРЫ:

1. Blended Attacks and Web 2.0 Threats: Are You Ready for 2009? URL: http://securitylabs.websense.com/content/Alerts/3277.aspx.
2. Ask MAMA what the Web is. URL: http://www.opera.com/press/releases/2008/10/15/.
3. *Вояковская Н. Н., Москаль А. Е., Булычев Д. Ю., Терехов А. А.* Анализ потока управления. URL: http://www.intuit.ru/department/sa/compilersdev/12/.
4. *Левенштейн В. И.* Двоичные коды с исправлением выпадений, вставок и замещений символов // Докл. АН СССР. 1965. Т. 163. № 4. С. 845—848.
5. *Кнут Э. Д.* Искусство программирования. Том 1. Основные алгоритмы. М.: Издательский дом «Вильямс», 2000. — 832 с.

В. В. Филатов

СИСТЕМЫ УПРАВЛЕНИЯ СОБЫТИЯМИ ИНФОРМАЦИОННОЙ БЕЗОПАСНОСТИ

В условиях сложившейся ситуации в области защиты информации проблема сбора информации о событиях становится наиболее актуальной. В организациях участились инциденты информационной безопасности, связанные с несовершенством систем защиты информации, используемых для защиты автоматизированных систем.

В крупных организациях используются сложные информационные системы, состоящие из большого числа сетевых устройств, операционных систем, антивирусного программного

обеспечения, систем управления базами данных, систем обнаружения вторжений, межсетевых экранов и т. д. Перечисленные подсистемы имеют встроенные средства защиты информации и мониторинга событий информационной безопасности. Для получения полноценной картины об угрозах информационной безопасности в подобных системах необходимо анализировать файлы журналов, содержащих огромное число записей, что, как правило, просто не под силу администраторам ИБ. Помочь в решении данной проблемы могут автоматизированные системы сбора информации о событиях.

Одним из таких решений является IBM Tivoli Security Operations Manager (TSOM). TSOM — это набор программных модулей для построения системы сбора и корреляции сообщений от различных устройств и программ обеспечения информационной безопасности. Данная система сбора событий предоставляет возможность автоматизировать трудоемкие, часто повторяющиеся операции, выполняемые специалистами по информационной безопасности для выявления угроз и предотвращения атак.

Основными функциями TSOM являются:

· централизованный сбор, накопление и анализ событий информационной безопасности от различных источников;

· корреляция полученных данных в соответствии с моделями угроз информационной безопасности;

· выявление угроз информационной безопасности;

· выполнение автоматических действий, направленных на предотвращение зафиксированных угроз и атак;

· оперативное уведомление оператора системы об инцидентах информационной безопасности [1].

TSOM состоит из следующих модулей:

· Central Management System (CMS) — центральная система управления, собирающая в единую базу данных информацию о событиях от различных модулей, производит корреляцию и анализ собранных данных для выявления угроз безопасности;

· Event Aggregation Module (EAM) — накапливает данные, поступающие из различных систем безопасности, нормализует, фильтрует, группирует и затем передает в центральную систему управления;

· Universal Collection Module — используется для сбора информации о событиях от источников, которые невозможно подключить напрямую к модулю EAM. Принимает сообщения от систем на основе правил и передает в модуль EAM [2].

TSOM анализирует и располагает по приоритетам данные о событиях, используя множественный анализ и корреляции:

· Статистическая корреляция (Statistical Correlation) выявляет подозрительные события, используя усовершенствованный метод анализа событий и хостов.

· Корреляция на основе правил (Rule-Based Correlation) выявляет известные атаки и нарушения политики безопасности, принятой в организации [1].

Корреляция — это взаимосвязь двух и более параметров в событии, выявляющая модели угроз информационной безопасности. Примером таких корреляций могут быть:

· изменение политики аудита в системе;

· очистка журнала аудита в системе;

· пять неуспешных попыток входа в систему под учетной записью пользователя;

· три неуспешные попытки входа в систему под учетной записью администратора.

Поскольку TSOM поддерживает большое число различных систем, работа администратора ИБ существенно упрощается за счет сопоставления, анализа и корреляции данных, поступающих

из различных источников. Таким образом, появляется возможность оперативного выявления угроз безопасности, регистрации инцидента и проведения соответствующих действий по обнаружению нарушителя и предотвращению атак.

Нарушения политики безопасности могут иметь серьезные последствия для организации, такие как потеря репутации на рынке, огромные финансовые затраты. Поэтому организации нуждаются в быстром выявлении угроз безопасности и реагировании на возникшие нештатные ситуации. TSOM способен помочь предотвратить вторжения, повысить уровень безопасности в организации и существенно облегчить труд администраторов ИБ.

СПИСОК ЛИТЕРАТУРЫ:

1. http://www-01.ibm.com/software/tivoli/products/security-operations-mgr/.
2. Документация IBM Tivoli Security Operations Manager.

А. Д. Чорняк

СТАТИСТИЧЕСКИЙ МЕТОД ВЫДЕЛЕНИЯ СИГНАТУР РАЗРУШАЮЩИХ ПРОГРАММНЫХ ВОЗДЕЙСТВИЙ

Одним из основных методов борьбы с вирусами является сигнатурный анализ [1]. Однако отсутствие жестко заданного алгоритма выделения и методики оценки сигнатур, а также наличие случайных ошибок, возникающих при ручном анализе, снижают надежность данного подхода. Таким образом, возникает необходимость создания автоматизированной системы выделения сигнатур разрушающих программных воздействий (РПВ). В данной работе предлагается статистический подход к автоматизации процесса выделения сигнатур.

Для начала необходимо выделить множество файлов, подверженных РПВ (R). С этой целью производится запуск модели РПВ, после чего осуществляется сравнение контрольных сумм файловой системы до и после заражения. Элементы множества R можно получить не только исполнением самого РПВ, но и запуском его потомков. Чем больше множество R, тем меньше вероятность ошибок типа «false negative», а это является наиболее важным фактором при сигнатурном анализе [1].

Далее необходимо создать максимально полное множество всех программ, не подверженных РПВ. Это множество будет представлять собой «чистую систему» (С). Чем оно больше, тем меньше вероятность ошибок типа «false positive».

По сути, задача выявления сигнатур сводится к нахождению последовательности, присутствующей во всех файлах, зараженных данной моделью РПВ, и не встречающейся ни в одном файле «чистой системы». Т. е. необходимо найти новое множество потенциальных сигнатур (SP), представляющее собой вычитание множества С из множества R.

Любую из полученных последовательностей можно использовать в качестве сигнатуры. Однако то, что выбранные последовательности не встречаются в «чистой системе», еще не гарантирует отсутствие ложных срабатываний. Это связано с тем, что невозможно составить полное