УДК 004.7

НАБЛЮДЕНИЕ АНОМАЛИЙ В СЕТИ

А.А. Сапожников

NETWORK ANOMALIES OBSERVATION

A.A. Sapozhnikov

Основная цель данной работы - создание системы мониторинга ИТ-инфраструктуры предприятия на основе различных продуктов с открытыми исходными кодами. Элементы данной системы должны требовать минимум усилий по конфигурированию, и динамически адаптироваться к изменяющимся параметрам наблюдаемых систем.

Ключевые слова: обнаружение вторжений, аномалии, сети, мониторинг.

That paper discusses one of network monitoring problem decision based on various open source software. Monitoring system is easily configurable and dynamically adopt to variations of monitored network.

Keywords: intrusion detection, anomaly, networks, monitoring.

Введение

Локальные вычислительные сети современных предприятий наполнены различными приложениями и решают множество задач. Поведение сети формируется пользователями, сервисами программно-аппаратных средств, другими сетевыми устройствами. Под нормальным функционированием сети мы имеем в виду следующее: гарантированное предоставление сервисов и устойчивость к различным стрессовым воздействиям, которые в самом общем рассмотрении можно разделить на непроизвольные (отказ оборудования, ошибки в программах) и произвольные (целенаправленные атаки).

1. Наблюдаемые события

Существуют три уровня обороны ИТ-инфраструктуры. Уровень первый - межсетевой экран. На данном уровне происходит защита от ненужного или злонамеренного трафика. Второй уровень - сетевые системы обнаружения вторжений (NIDS). Данные системы детектируют нежелательный и злонамеренный трафик внутри ИТ-инфраструктуры. Третий уровень - верификация целостности данных - отслеживание изменений в системе, например, путем сравнения криптографических контрольных сумм файлов с данными в базе.

Для достижения максимального уровня защищенности объекта ИТ-инфраструктуры, необходимо реализовывать на нем все три уровня защиты, причем каждый уровень должен быть независимым, т.е. в том случае, если злоумышленник проникнет за один из уровней защиты, то все остальные продолжают полноценно функционировать. Таким образом, реализуется добавочная защита [1].

Количество возможных векторов удаленной атаки, весьма ограничено по сравнению с локальными атаками.

Как правило, предотвратить атаку невозможно. Если известно, что какая-то система уязвима, то значительно эффективнее будет ликвидировать данную уязвимость, либо удалить систему из ИТ-инфраструктуры до полного восстановления статуса защищенной системы.

Именно поэтому есть смысл сконцентрировать все усилия на мониторинге последствий атаки.

Атаки можно разделить на три уровня.

• Транспортный уровень.

Под атаками на транспортный уровень будем понимать, атаки на протоколы канального, сетевого и транспортного уровней стека TCP/IP. К этому уровню относятся, например, различные виды сканирования, arp-spoofing, ip-spoofing, ping of death [4].

• Прикладной уровень.

Под этим уровнем будем понимать атаки направленные на ошибки в реализации различных протоколов прикладного уровня стека TCP/IP.

Примерами подобных атак могут быть dns cache-poisoning, smb-die, apache http-chunked encoding.

• Уровень сервиса.

К атакам данного уровня отнесем, всевозможные атаки вызванные ошибками в некорректной обработке пользовательских данных. Примерами подобных атак будут XSS, SQL-injection, различные переполнения, двойное освобождение памяти и т.д. [5, 6]

Обнаружить атаку можно двумя методами.

• Сигнатурный. Данный метод сводится к поиску признаков уже известных атак.

Сапожников Антон Александрович – аспирант ЮУрГУ; anton.a.sa@gmail.com

Sapozhnikov Anton Aleksandrovich – postgraduate student of SUSU; anton.a.sa@gmail.com

• Аномальный. Заранее известно какими функциональными параметрами обладает то или иное приложение или сервис в нормальном состоянии, и любое отклонение от него считается атакой.

Преимущество сигнатурного метода в том, что он практически не подвержен ложным срабатываниям. Минусом данного метода, является невозможность обнаруживать не заложенные в систему атаки.

Метод поиска аномалий позволяет реагировать на ранее неизвестные атаки, но подвержен ложным срабатываниям и требует точной настройки для каждого наблюдаемого объекта.

Оба метода обнаружения атак могут работать на всех трех уровнях.

Результатом любой атаки, в случае успеха, является утечка информации, например, атака ether-leak, либо изменение каких-либо параметров атакованной системы, например открытие порта, прекращение работы некоторой программы, значительное изменение объемов использования некоторого ресурса системы программой (например, памяти, процессорного времени и т.п.), запуск новой программы на выполнение. Подобные события, происходящие в системе, можно отслеживать с помощью программных средств [2].

Так как значительное изменение объемов использования ресурсов может происходить и при нормальном режиме работы системы (например, увеличение обращений к почтовому серверу в начале рабочего дня), то возникает проблема определения поведения программы: нормального или аномального. Предлагается решение данной проблемы на основе статистических методов анализа.

2. Обнаружение аномалий

Идея заключается в следующем. В основном, все реальные автоматизированные информационные системы имеют циклический характер функционирования, который определяется рабочей неделей или производственным технологическим процессом. Предположим, что существует отдельный период на начальном этапе работы системы, в течение которого мы можем утверждать, что система работает в нормальном режиме. Если такой интервал времени существует, то назовем его периодом обучения [7].

В течение периода обучения будем отслеживать использование различных ресурсов программами и на базе накопленной информации сможем построить функцию прогнозирования дальнейшего поведения программ.

В период рабочей эксплуатации системы будем постоянно производить мониторинг использования различных ресурсов программами. Полученные данные будем сравнивать с прогнозируемыми. Если различие между прогнозируемым и фактическим использованием ресурсов превышает некоторое допустимое значение, то поведение программы считается аномальным. При этом принимается решение об изменении режима функционирования программы с целью предотвращения нарушения стабильности работы всей системы в целом.

В некоторых случаях возможно применять более простой способ обнаружения аномалий, для этого у наблюдаемого параметра системы определяются верхняя и нижняя границы его некоторой числовой характеристики в период нормальной работы. В случае же выхода за их пределы, детектируется аномальное поведение.

Вырожденным случаем определения границ, является полное отсутствие, или близкое к нулю, какого-либо параметра. Например, о наблюдаемой системе известно, что на ней функционирует только веб-сервер, т.е. в период обучения наблюдался только трафик на порт 80, но в результате атаки, обнаруживается трафик на порт 6667, которого раньше не было.

3. Структура системы

Разработанная система представляет собой совокупность следующих модулей:

• сбора статистики;

• обучения;

• прогнозирования дальнейшего поведения программ;

• реагирования на аномальное поведение программ.

Модульная структура позволяет создать приложение легко переносимое на различные платформы, причем в дальнейшем возможно будет реализовать клиент-серверную архитектуру без внесения больших изменений, поскольку интерфейс обмена данными между модулями строго унифицирован.

Одной из проблем мониторинга больших сетей является огромное количество информации, которое необходимо проанализировать, поэтому с целью снижения требований к аппаратной реализации модулей сбора статистики, необходимо увеличить их число с целью реализации распределенной системы мониторинга, что повысит отказоустойчивость системы в целом [9].

Модули сбора статистики получают информацию об использовании различных ресурсов автоматизированной системы программами и передают ее модулю анализа через стандартизированный интерфейс. Это позволяет сделать наиболее сложные модули обучения, анализа и прогнозирования поведения программ платформенно-независимыми.

Модуль обучения анализирует данные, собранные модулем статистики в период обучения, и строит функцию, прогнозирующую дальнейшее поведение программы. Функция прогнозирования строится на базе функциональной зависимости использования ресурса программой от времени в

течение периода обучения (далее исходная функция) следующим образом.

Вначале производится сглаживание исходной функции для отфильтровывания случайных шумов. Сглаживание выполняется вычислением скользящего взвешенного пятиточечного среднего с оптимально подобранными весовыми коэффициентами:

$$WMA_t = \frac{\sum_{i=t-5}^{t} h_i \cdot W_{t-i}}{\sum_{i=1}^{5} W_i}, \quad (1)$$

где t - время измерения параметра, W - весовой коэффициент, h - наблюдаемый параметр.

Далее производится разделение исходной функции на две составляющие. Первая - тренд - определяет тенденцию в использовании ресурса. Вторая - сезонная компонента - определяет периодическую составляющую исходной функции. Выделение тренда осуществляется вычислением математического ожидания значений исходной функции на достаточно большом временном отрезке. Сезонная компонента находится как разность сглаженной исходной функции и тренда. При дальнейшем анализе сезонной компоненты производится выделение основных гармонических составляющих. Для этого используется дискретное преобразование Хартли (аналог преобразования Фурье) с последующим отбрасыванием малозначимых коэффициентов.

Прямое дискретное преобразование Хартли

$$H_k = \sum_{i=0}^{N-1} h_i \cdot \mathrm{cas}\left(\frac{2\pi}{N} ki\right), \quad k = 0, \ldots, N-1. \quad (2)$$

Обратное дискретное преобразование Хартли:

$$h_i = \frac{1}{N} \sum_{k=0}^{N-1} H_k \cdot \mathrm{cas}\left(\frac{2\pi}{N} ik\right), \quad i = 0, \ldots, N-1, \quad (3)$$

где

$$\mathrm{cas}\alpha = \cos\alpha + \sin\alpha. \quad (4)$$

Функция прогнозирования строится как сумма тренда и основных гармонических составляющих сезонной компоненты с соответствующими коэффициентами. Затем производится расчет среднеквадратичного отклонения сглаженной исходной функции от функции прогнозирования, на основе которого определяется качество прогноза и вычисляется максимально допустимое отклонение фактического поведения от ожидаемого.

Модуль реагирования на аномальное поведение программ анализирует поступающие сообщения и может выполнять ряд функций, таких как оповещение администратора системы о произошедшем событии, отключение источника аномалии от остальной сети, с целью снизить возможное негативное влияние на другие компоненты наблюдаемой автоматизированной системы (АС), запись события в журнал регистрации для дальнейшего анализа. Между тем, сами факты обнаружения аномалий являются событиями в наблюдаемой АС, которые должны быть подвергнуты анализу, как сигнатурному, так и статистическому. Примером подобного события может быть массовое распространение вируса внутри сетевого сегмента АС, реакцией на подобное происшествие может быть полное отключение сегмента до выяснения и устранения причин, вызвавших распространение зловредного кода, с целью ограничить распространение зловредного кода на другие сегменты сети наблюдаемой автоматизированной системы.

4. Перспективы

В рамках данного проекта в перспективе планируется:

- создание адаптивной модели поведения процесса, позволяющей учитывать новые формы поведения, не являющиеся аномальными для данного процесса и типа ресурса; которые могут возникать в результате переконфигурирования процесса или какого-либо изменения в режиме функционирования;
- снижение необходимости человеческого вмешательства в работу системы, для принятия решений о предпринимаемых действиях при обнаружении аномального поведения; вплоть до полной автоматизации процесса реагирования на какие-либо аномальные отклонения показателей в работе системы [8].

Выводы

Данная разработка является инструментом, позволяющим повысить отказоустойчивость и облегчить администрирование автоматизированных систем. Следствием этого является снижение затрат на обслуживание АС в целом. Кроме того, кратковременный сбой в работе большинства существующих информационных систем может привести к потере важной информации, значительному экономическому ущербу, уменьшению количества клиентов и т.п. Поэтому, любое увеличение отказоустойчивости системы способно снизить потери, возникающие в результате сбоев АС [3].

Литература

1. Щеглов, А.Ю. Защита компьютерной информации от несанкционированного доступа / А.Ю. Щеглов. - СПб.: Наука и Техника, 2004. - 384 с.

2. Lucas, Michael W. Absolute OpenBSD. UNIX for the Practical Paranoid / Michael W. Lucas. - No Starch Press. ISBN I-886411-99-9, 2003. - 528p.

3. Garfinkel, Schwartz, Spafford. Practical Unix & Internet Security f Garfinkel, Schwartz, Spafford. - O'Reilly. ISBN 0-596-00323-4, 2003.-984p.

4. Russell. Stealing the Network: How to Own the Box / Russell, Dubrawsky, FX, Grand, Mullen. - Syngress Publishing. ISBN 1-931836-87-6, 2003. - 330p.

5. Foster, James C. Programmer's Ultimate Security DeskRef/ James C Foster, Steven C Foster. - Syngress. ISBN 1-932266-72-0, 2004. - 700p.

6. Hoglund, Greg. Exploiting Software. How to Break Code / Greg Hoglund, Gary McGraw. - Addison Wesley. ISBN 0-201-78695-8, 2004.-512p.

7. Мониторинг состояния автоматизированной системы и обеспечение стабильности / А.А. Сапожников, А.В. Жуков, М.Л. Карманов, П.В. Збицкий // Сборник материалов Всероссийского конкурса инновационных проектов аспирантов и студентов по приоритетному направлению развития науки и техники «Информационно-телекоммуникационные системы». - М.: ГНИЙ ИТТ«Информика», 2005.-С. 123-124.

8. Сапожников, А.А. Нейронные сети для анализа журналов регистрации I А.А. Сапожников II Безопасность информационного пространства: материалы всерос. науч.-практ. конф. - Екатеринбург: УГТУ-УПИ, 2005. - С. 21-22.

9. Сапожников, А.А. Практика централизованного мониторинга сетей / А.А. Сапожников /I Международная конференция «Проблемы функционирования информационных сетей»: материалы конф. - Новосибирск: ЗАО РИЦ Прайс Курьер, 2006. - С. 257-260.

Поступила в редакцию 10 апреля 2009 г.