

УДК 004.056+519.83

Алгоритм выбора контролируемых узлов в системе обнаружения вторжений на основе модели биматричной игры

Шматова Е.С., студент

*Россия, 105005, г. Москва, МГТУ им. Н.Э. Баумана,
кафедра «Информационная безопасность»*

*Научный руководитель: Быков А.Ю., к.т.н, доцент
Россия, 105005, г. Москва, МГТУ им. Н.Э. Баумана,
кафедра «Информационная безопасность»*

abykov@bmstu.ru

Введение

Система обнаружения вторжений – программное или программно-техническое средство, реализующее функции автоматизированного обнаружения действий в информационной системе (ИС), направленных на преднамеренный доступ к информации, специальные воздействия на информацию в целях ее добывания, уничтожения, искажения и блокирования доступа к ней.

При построении СОВ необходимо задать требования, предъявляемые к системе. Затем следует проанализировать текущую конфигурацию сети и определить оптимальные точки подключения сенсоров СОВ, таким образом задав объекты мониторинга сети. Сенсор (датчик) СОВ является программным или программно-техническим компонентом СОВ, предназначенным для сбора и первичного анализа информации о событиях в контролируемой ИС, а также – передачи этой информации анализатору СОВ.

В рамках крупного предприятия целесообразно рассматривать распределенные СОВ с множеством сенсоров, размещенных по всей компьютерной сети и имеющих централизованный анализатор событий [8]. Такие системы способны собирать и анализировать значительно больший объем информации, что позволяет обнаруживать широкий спектр атак. Однако внедрение и управление большим количеством сенсоров с учетом финансовых, кадровых и других ограничений может стать непростой задачей. Также необходимо учесть тот факт, что сбор данных с множества узлов способен создать дополнительную нагрузку на сеть, которая, в случае полномасштабной атаки, может превысить ее пропускную способность. Следовательно, задача поиска оптимальных точек

подключения сенсоров является наиболее критичным этапом конфигурирования распределенной СОВ.

Цель данной статьи состоит в разработке математической постановки задачи выбора контролируемых узлов в СОВ и рассмотрении алгоритмов, которые могут быть использованы для решения поставленной задачи.

Обзор публикаций по настоящей проблеме показывает, что на данный момент уже предложен ряд эффективных моделей и методов. Так для решения задачи были использованы многокритериальные генетические алгоритмы оптимизации [1], анализ графов атак [3], теоретико-игровые подходы, а также формальный подход, основанный на логике, для описания сетей, их анализа и определения способа размещения сенсоров [4] и др. Однако рассматриваемый вопрос все еще остается открытым и имеется возможность внести свой вклад решение задачи. В настоящей статье будет предложена постановка задачи выбора контролируемых узлов в СОВ на основе модели биматричной игры, в которой участвуют две противоборствующие стороны. Подобные модели принятия решений рассматриваются в теории игр, которая нашла широкое применение в области, связанной с обеспечением информационной безопасности [5,6,9].

1. Математическая постановка задачи

1.1 Исходные данные

Рассмотрим постановку задачи выбора контролируемых узлов в СОВ на основе модели биматричной игры, в которой участвуют две противоборствующие стороны: нарушитель и защитник. В рамках задачи в качестве стороны защиты выступает СОВ. Подобная постановка была представлена в [2].

Исходные данные задачи:

1) $S = \{s_1, s_2, \dots, s_n\}$ – множество узлов сети, на которых могут находиться защищаемые информационные ресурсы, где $N = \{1, 2, \dots, n\}$ – множество индексов узлов.

2) $w_i > 0, \forall i \in N$ – стоимость защищаемых ресурсов (активов) узла сети (при нарушении требований безопасности w_i составляет возможный ущерб).

3) $c_{zi} > 0, \forall i \in N$ – стоимость защиты узла.

4) $c_{ni} > 0, \forall i \in N$ – стоимость проведения атаки на узел со стороны нападения.

5) $p_{ni} > 0, \forall i \in N$ – вероятность необнаружения атаки на узле.

1.2 Показатели игроков

В отношении каждого узла для сторон защиты и нападения возможны две стратегии: сторона защиты проводит защиту узла или не проводит, сторона нападения проводит атаку на узел или нет.

Предложим детерминированную постановку задачи. Для стороны защиты введем булеву переменную $p_i \in \{0, 1\}, \forall i \in N$:

$$p_i = \begin{cases} 1, & \text{если } i - \text{ый узел ИС защищен,} \\ 0, & \text{в противном случае.} \end{cases}$$

Аналогично для стороны нападения введем булеву переменную $q_i \in \{0, 1\}, \forall i \in N$:

$$q_i = \begin{cases} 1, & \text{если } i - \text{ый узел будет атаковаться,} \\ 0, & \text{в противном случае.} \end{cases}$$

Переменные p_i и q_i образуют векторы $\vec{P} = (p_1, p_2, \dots, p_n)$ и $\vec{Q} = (q_1, q_2, \dots, q_n)$, соответственно, которые определяют стратегии игроков по отношению ко всем узлам сети.

Сперва рассмотрим платежные матрицы сторон защиты и нападения для отдельного i -го узла, которые представлены в таблицах 1 и 2, соответственно.

Таблица 1

Матрица выигрыша стороны защиты для i -го узла

Сторона нарушителя \ Сторона защиты	Защищает	Не защищает
	Нападает	Не нападает
Нападает	$-p_{hi}w_i - c_{zi}$	$-w_i$
Не нападает	$-c_{zi}$	0

Таблица 2

Матрица выигрыша стороны нападения для i -го узла

Сторона нарушителя \ Сторона защиты	Защищает	Не защищает
	Нападает	Не нападает
Нападает	$p_{hi}w_i - c_{hi}$	$w_i - c_{hi}$
Не нападает	0	0

Приведенные таблицы задают биматричную игру [10] для отдельного узла. Для подобных игр можно получить решение оптимальное по Нэшу в чистых или смешанных стратегиях [10].

Вернемся к построению игры, охватывающей все узлы ИС. Выигрыш стороны защиты в целом, который можно рассматривать как результат игры всех «защитников» узлов, определяется возможным ущербом и затратами с противоположным знаком:

$$\begin{aligned} F_3(\vec{P}, \vec{Q}) &= \sum_{i \in N} [p_i q_i (-p_{ni} w_i - c_{zi}) - (1 - p_i) q_i w_i - p_i (1 - q_i) c_{zi}] = \\ &= \sum_{i \in N} [p_i q_i w_i (1 - p_{ni}) - q_i w_i - p_i c_{zi}]. \end{aligned} \quad (1)$$

Выигрыш стороны нападения, рассматриваемый как результат игры «атакующих» отдельные узлы, определяется возможным ущербом для стороны защиты минус затраты на проведение атак:

$$\begin{aligned} F_n(\vec{P}, \vec{Q}) &= \sum_{i \in N} [p_i q_i (p_{ni} w_i - c_{ni}) + (1 - p_i) q_i (w_i - c_{ni})] = \\ &= \sum_{i \in N} [q_i w_i - p_i q_i w_i (1 - p_{ni}) - q_i c_{ni}]. \end{aligned} \quad (2)$$

Введем обозначения:

$$U_{np}(\vec{P}, \vec{Q}) = \sum_{i \in N} p_i q_i w_i (1 - p_{ni}) - \text{непредотвращенный ущерб стороной защиты};$$

$U_{max}(\vec{Q}) = \sum_{i \in N} q_i w_i$ – максимальный ущерб, который может быть нанесен стороной нападения при отсутствии защиты;

$$Z_3(\vec{P}) = \sum_{i \in N} p_i c_{zi} - \text{затраты стороны защиты};$$

$$Z_n(\vec{Q}) = \sum_{i \in N} q_i c_{ni} - \text{затраты стороны нападения}.$$

Запишем показатели стороны защиты и нападения ((1) и (2), соответственно) в другой форме:

$$F_3(\vec{P}, \vec{Q}) = U_{np}(\vec{P}, \vec{Q}) - U_{max}(\vec{Q}) - Z_3(\vec{P}), \quad (3)$$

$$F_n(\vec{P}, \vec{Q}) = U_{max}(\vec{Q}) - U_{np}(\vec{P}, \vec{Q}) - Z_n(\vec{Q}). \quad (4)$$

1.3 Ограничения

Компоненты показателей (3) и (4), определяющие затраты защитника и нападающего перенесем в ограничения.

Общая стоимость ресурсов СОВ, выделяемых на защиту сети, равна

$C_3(\vec{P}) = \sum_{i \in N} c_{zi} p_i$. На ресурсы для стороны защиты введем следующее ограничение:

$$\sum_{i \in N} c_{zi} p_i \leq C_3^{max}, \quad (5)$$

где C_3^{max} – некоторый максимальный размер ресурсов, выделенных на защиту.

По аналогии для стороны нападения введем ограничения на ресурсы, выделяемые на проведение атак:

$$\sum_{i \in N} c_{ni} q_i \leq C_n^{max}, \quad (6)$$

где C_n^{max} – максимальный размер ресурсов, выделенных на проведение атак.

Исключим слагаемые с затратами из показателей (3) и (4). Показатель стороны защиты примет вид:

$$F_3(\vec{P}, \vec{Q}) = U_{np}(\vec{P}, \vec{Q}) - U_{max}(\vec{Q}) \quad (7)$$

Показатель стороны нападения:

$$F_n(\vec{P}, \vec{Q}) = U_{max}(\vec{Q}) - U_{np}(\vec{P}, \vec{Q}). \quad (8)$$

Ограничение для стороны защиты (5), ограничение для стороны нападения (6).

В этом случае получаем игру с нулевой суммой, выигрыш защитника – его ущерб с противоположным знаком, выигрыш нападающего – ущерб защитника.

1.4 Модель игры

Каждая из сторон стремится максимизировать свои показатели. Следовательно, необходимо найти оптимальную стратегию стороны защиты, обеспечивающую максимальный выигрыш. Такая стратегия может быть найдена с помощью соответствующих критериев оптимальности.

Для стороны защиты с учетом ограничения (5) имеем следующее:

$$F_3(\vec{P}, \vec{Q}) \rightarrow \max_{\vec{P} \in M_p} F_3(\vec{P}, \vec{Q}),$$

$$M_p: \sum_{i \in N} c_{3i} p_i \leq C_3^{max},$$

где M_p – множество допустимых векторов $\vec{P}$ для стороны защиты.

Аналогично для стороны нападения с учетом ограничения (6):

$$F_n(\vec{P}, \vec{Q}) \rightarrow \max_{\vec{Q} \in M_q} F_n(\vec{P}, \vec{Q}),$$

$$M_q: \sum_{i \in N} c_{ni} q_i \leq C_n^{max},$$

где M_q – множество допустимых векторов $\vec{Q}$ для стороны нападения.

При фиксированных значениях компонент вектора $\vec{P}$ или вектора $\vec{Q}$ может быть получена задача булевого программирования.

2. Алгоритмы решения задачи

2.1 Критерий оптимизации

Модель игры с учетом целевых функций (ЦФ) (7) и (8) может быть сведена к игре, заданной платежной матрицей. Однако размерность такой матрицы может быть достаточно велика: число строк будет равно числу допустимых значений вектора $\vec{P}$, удовлетворяющих ограничению (5), а число столбцов будет равно числу допустимых

значений вектора $\vec{Q}$, удовлетворяющих ограничению (6). Следовательно, при большой размерности платежной матрицы поиск такого решения будет затруднителен.

Для поиска решений можно использовать различные критерии оптимальности, такие как критерии Лапласа, Вальда, Гурвица или Сэвиджа. Поскольку каждая из сторон стремится максимизировать свой выигрыш, следует рассмотреть максимаксный критерий. Однако игровая модель была сведена к игре с нулевой суммой, где выигрыш защитника – его ущерб с противоположным знаком, а выигрыш нападающего – ущерб защитника, поэтому целесообразно записать критерий максимакса в следующем виде:

$$\min_{\vec{P} \in M_p} \max_{\vec{Q} \in M_q} F_n(\vec{P}, \vec{Q}). \quad (9)$$

При решении задачи максимизации показателя $F_n(\vec{P}, \vec{Q})$ по вектору $\vec{Q} \in M_q$ при фиксированных значениях вектора $\vec{P}$ или задачи минимизации показателя $F_n(\vec{P}, \vec{Q})$ по вектору $\vec{P} \in M_p$ при фиксированных значениях вектора $\vec{Q}$ решаются задачи булевого программирования с линейным ограничением (5) или (6) и показателем качества (8). Такие задачи относятся к классу NP -полных задач оптимизации [7]. Для решения задачи можно предложить точные методы, подходящие для решения задачи по критерию (9), рассмотренные в [7].

2.2 Аддитивный алгоритм неявного перебора

Среди точных методов в булевом программировании всегда можно выделить метод полного перебора. Однако есть методы, позволяющие некоторым образом полный перебор.

Аддитивный алгоритм неявного перебора на решетке и на дереве применяется для решения задач линейного программирования (ЗЛП) с булевыми переменными. Пусть задача представлена в виде:

$$f(\vec{x}) = \sum_{j=1}^n c_j x_j \rightarrow \min, \quad \sum_{j=1}^n a_{ij} x_j \leq b_i (i = 1, 2, \dots, m), \quad x_j = \{0, 1\}, \quad (10)$$

где все коэффициенты c_j при переменных x_j положительны и упорядочены следующим образом: $c_1 \leq c_2 \leq \dots \leq c_n$. Любую ЗЛП можно свести к данному виду.

Каждому узлу решетки соответствует вектор $\vec{x} = \langle x_1, x_2, \dots, x_n \rangle$, в котором часть переменных принимает значение 1 (фиксированные переменные), а остальные переменные принимают значение 0 (свободные переменные). Узел доминирует все узлы нижестоящих уровней, которым соответствуют кортежи, полученные путем замены тех или иных нулей единицами. Если кортеж исходного узла располагается на k -ом уровне, то множество Ω , состоящее из всех номинируемых им кортежей, включая и сам кортеж, состоит из 2^{n-k}

альтернатив. Проверка этого множества на существование в нем допустимой альтернативы со значением ЦФ лучшим по сравнению с ранее достигнутым рекордным значением называется зондированием.

Приемы, позволяющие сократить полный перебор:

1. *Расширение числа ограничений.* В число ограничений вводится нулевое ограничение вида $\sum_{j=1}^n a_{0j} x_j \leq b_0^s$, где $a_{0j} = c_j$, а $b_0^s = z_p^s$ – рекордное значение ЦФ, полученное на предыдущем s-ом зондировании.

2. *Проверка наиболее предпочтительного кортежа.* В задаче минимизации это кортеж зондируемого множества, в котором все свободные переменные заменены нулями. Если данный кортеж допустим, то отпадает необходимость дальнейшего зондирования.

3. *Проверка необходимого условия для ограничений.* Чтобы в зондируемом множестве могла находиться альтернатива, улучшающая достигнутый рекорд, необходимо, чтобы выполнялось следующее соотношение:

$$\nexists (i \in \{0, 1, \dots, m\}) (\sum_{j \in \Omega} \min(a_{ij}, 0) > b_i - \sum_{j \in N/\Omega} a_{ij} x_j), \quad (11)$$

где $N \in \{1, \dots, n\}$ – множество индексов всех переменных; Ω – множество индексов свободных переменных; N/Ω – множество индексов фиксированных переменных. Если соотношение не выполняется, то этого достаточно, чтобы считать рассматриваемое множество прозондированным и не содержащим допустимых альтернатив, улучшающих рекорд.

4. *Определение значений свободных переменных.* Если наиболее предпочтительная альтернатива не допустима и соотношение (11) выполняется, то зондируемое множество считается подозрительным на существование других допустимых альтернатив. К свободным переменным могут быть представлены следующие требования:

$$\forall (k \in \Omega) ((\exists (i \in \{0, 1, \dots, m\}) (\sum_{j \in \Omega} \min(a_{ij}, 0) + |a_{ik}| > b_i - \sum_{j \in N/\Omega} a_{ij} x_j)) \Rightarrow \\ \Rightarrow x_k = 0 \text{ при } a_{ik} > 0 \text{ и } x_k = 1 \text{ при } a_{ik} < 0) \quad (12)$$

Если в результате процедуры, отвечающей требованию (12), определяются все свободные переменные, то полученное решение называется полным, последнее называется полным допустимым решением, если удовлетворяет также требованию допустимости.

Структурная схема алгоритма поиска на дереве приведена на рисунке 1.

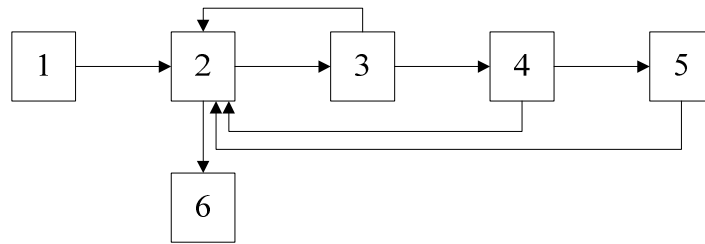


Рис. 1. Структурная схема алгоритма поиска не дереве

Описание алгоритма:

Шаг 1. Начать список задач зондирования введением в него двух задач, определяемых фиксированием значений некоторой переменной $x_i = 0$ и $x_i = 1$, задать начальный рекорд $z_p^0 = +\infty$ в задачах минимизации и $z_p^0 = -\infty$ в задачах максимизации.

Шаг 2. На каждой $(s+1)$ -ой итерации проверяется список задач. Если он пуст, то вычисления окончены, переходим к шагу 6. Если он не пуст, то выбираем из списка одну задачу (задача из списка вычеркивается) и переходим к шагу 3.

Шаг 3. Проверить для выбранной задачи выполнение соотношения (11). Если условие не выполняется, то положить $z_p^{s+1} = z_p^s$ и перейти к шагу 2. Если условие выполняется, то в случае полноты проверяемого решения $\vec{x}^{s+1}$ обновить рекорд $z_p^{s+1} = f(\vec{x}^{s+1})$ и перейти к шагу 2, а в случае его неполноты – к шагу 4.

Шаг 4. С помощью соотношения (12) произвести расширение частичного решения. Если расширение возможно и расширенное решение $\vec{x}_p^{s+1}$ полно и доступно, обновляем рекорд $z_p^{s+1} = f(\vec{x}_p^{s+1})$ и переходим к шагу 2. В противном случае переходим к шагу 5, сохранив прежнее значение рекорда ($z_p^{s+1} = z_p^s$).

Шаг 5. Выбрать некоторую оставшуюся свободной после расширения решения переменную x_k . Положить $x_k = 0$ и $x_k = 1$ и внести в список задач соответствующие две задачи. Перейти к шагу 2.

Шаг 6. Работа алгоритма закончена.

2.3 Описание алгоритма решения задачи

Ниже представлен алгоритм, предложенный в [7], с помощью которого можно найти решение задачи по критерию (9):

Описание алгоритма:

Шаг 1. Задаем начальное решение $\vec{P} = \|0, 0, \dots, 0\|^T$, полагаем $UMinMax = \infty$.

Шаг 2. Задаем начальное решение $\vec{Q} = \|0, 0, \dots, 0\|^T$, полагаем $UMax=0$.

Шаг 3. Запускаем аддитивный алгоритм неявного перебора, описанный в пункте 2.2, в котором ведется поиск максимального значения целевой функции (8), соответствующее решение $\vec{Q}$ записывается в массив $QRec$. Допустимость решения $\vec{Q}$ проверяется по ограничению (6). На выходе получаем значение $UMax$.

Шаг 4. Если $UMax < UMinMax$, то полагаем $UMinMax = UMax$ и сохраняем решение $QRec$ в массиве $QRec2$ (рекордное решение по критерию минимакс), сохраняем значение массива $\vec{P}$ в массиве $PRec$ (рекордное решение по критерию минимакс).

Шаг 5. С помощью аддитивного алгоритма неявного перебора (пункт 2.2) получаем новое решение $\vec{P}_{next}$, допустимое по ограничению (5), и переходим к Шагу 2. Если допустимых решений больше нет, то алгоритм прекращает свою работу, выполняется переход к шагу 6.

Шаг 6. По завершении работы алгоритма значение показателя для первого игрока, (в качестве него выступает СОВ) найденное по критерию минимакс (9), будет равно $UMinMax$, полученное решение будет записано в массиве $PRec$, решение второго игрока будет в массиве $QRec2$.

Заключение

В статье рассмотрена математическая постановка задачи на основе биматричной игры с булевыми переменными, в которой принимают участие две противоборствующие стороны: сторона защиты и сторона нападения. В качестве задачи выступает проблема выбора контролируемых узлов в СОВ. Благодаря простым преобразованиям задача была сведена к игре с нулевой суммой. Такую задачу можно было представить в виде платежной матрицы. Однако размерность такой матрицы может быть достаточно велика, что затруднит поиск решения. Поэтому было решено в качестве критерия оптимальности использовать критерий минимакса. Это позволило свести решение задачи к решению последовательности задач булева программирования с линейными ограничениями (5) и (6) и показателем качества (8). Был предложен алгоритм решения задачи, позволяющий сократить полный перебор на основе методов неявного перебора.

Представленные модели и алгоритм могут быть применены для решения задач выбора контролируемых узлов при проектировании СОВ.

Список литературы

1. Chen H., Clark J.A., Shaikh S.A., Chivers H., Nobles P. Optimising IDS sensor placement // Availability, Reliability, and Security, 2010. ARES '10 International Conference. Los Alamitos, CA: IEEE. 2010. P. 315-320.
2. Lin Chen, Leneutre J. A Game Theoretical Framework on Intrusion Detection in Heterogeneous Networks // IEEE Transactions on Information Forensics and Security. 2009. Vol. 4(2). P. 165-178. DOI: 10.1109/TIFS.2009.2019154.
3. Noel S., Jajodia S. Attack graphs for sensor placement, alert prioritization, and attack response // Cyberspace Research Workshop. 2007. P. 120-127.
4. Rolando M., Rossi M., Sanarico N., Mandrioli D. A formal approach to sensor placement and configuration in a network intrusion detection system // SESS '06: Proceedings of the 2006 international workshop on Software engineering for secure systems, ACM. 2006. P. 65-71. DOI: 10.1145/1137627.1137638.
5. Xiannuan Liang, Yang Xiao. Game Theory for Network Security // IEEE Communication surveys and tutorials. 2013. Vol. 15(1). P. 472-486. DOI: 10.1109/SURV.2012.062612.00056.
6. Белый А.Ф. Компьютерные игры для выбора методов и средств защиты информации в автоматизированных система // Известия ЮФУ. Технические науки. Тематический выпуск: «Информационная безопасность». Ростове-на-Дону. 2008. № 8. С. 172-176.
7. Быков А.Ю., Алтухов Н.О., Сосенко А.С. Задача выбора средств защиты информации в автоматизированных системах на основе модели антагонистической игры // Инженерный вестник МГТУ им. Н.Э. Баумана. Электрон. журн. МГТУ им. Н.Э. Баумана. 2014. № 4. С. 525-542.
8. Васильев В.И., Котов В.Д. Современное состояние проблемы обнаружения сетевых вторжений // Вестник УГАТУ. 2012. Т.16, № 3 (48). С. 198-204.
9. Гуц А.К., Вахний Т.В. Теория игр и защита компьютерных систем: учебное пособие. Омск: Из-во ОмГУ, 2013. 160 с.
10. Стрекаловский А.С., Орлов А.В. Биматричные игры и билинейное программирование. М.: Физматлит, 2007. 224 с.