ПРИМЕНЕНИЕ МЕТОДОВ АРГУМЕНТАЦИИ ДЛЯ ЗАДАЧ ИНФОРМАЦИОННОЙ БЕЗОПАСНОСТИ[1]

Суворов А.В.[2], Моросин О.Л.[3], Вагин В.Н.[4]

В данной работе рассматриваются идеи и подходы к применению аргументации в задачах информационной безопасности. Решение проблем сетевой безопасности является сложной задачей, включающей большое количество факторов и требующей нахождения разумных компромиссов между поддержанием безопасности, стабильной работой, увеличением затрат на эксплуатацию и ограничениям по функциональности сложных информационных систем. В этой статье мы предлагаем использовать аргументацию, для обеспечения автоматизированной поддержки решений безопасности. Классические методы логического вывода не обладают механизмом «пересмотра» сделанных ранее выводов и отсутствует возможность обнаружения и разрешения конфликтов в знаниях. Одним из способов обработки конфликтных ситуация и противоречивой информации является применение аппарата аргументации. Аргументация дает больше возможностей для моделирования правдоподобных рассуждений. В работе используется система аргументации, основанная на пересматриваемых рассуждениях. Предложенный подход позволяет давать числовые оценки качества вырабатываемых системой рекомендаций, тем самым позволяя решить важную задачу – задачу выбора способа реагирования на имеющуюся в системе подозрительную активность. Кроме того, в работе приводится пример обработки возникающих в системе опасных ситуаций.

***Ключевые слова**: информационная безопасность, аргументация, пересматриваемые рассуждения, степени обоснования, разрешение противоречий.*

DOI:10.21581/2311-3456-2016-5-21-27

Введение

Современные системы безопасности должны обрабатывать большие объемы информации, которая часто бывает зашумленной и противоречивой [1]. Часто системы безопасности используют механизм классической логики (см., например, [2,3]) для описания правил обнаружения вторжений и других нежелательных действий пользователей в информационные системы. При этом, ложное срабатывание механизмов защиты является крайне нежелательным, так как может привести к значительным финансовым потерям. Простейшим примером являются брандмауэры (firewalls), содержащие тысячи[2] правил обнаружения подозрительной активности, при срабатывании которых происходит блокировка обмена данными с пользователем. Снижение процента ложных срабатываний систем защиты является важной проблемой [4], решение которой существенно повысит качество систем компьютерной безопасности. Обычно вопрос о срабатывания защиты рассматривается только со стороны, обнаружения факта вторжения и вопрос о возможных последствиях срабатывания защиты обычно не рассматривается [5]. В данной работе предлагается рассмотреть механизм принятия решения о срабатывания защиты не только с точки зрения оценки вероятности злонамеренных действий пользователя, но и с точки зрения оценки потенциальных рисков от применения механизмов защиты. Предлагается использовать механизм теории аргументации для определения необходимости применения защитных действий с учетом возможных рисков от ложного срабатывания систем защиты. Аргументация является формальным подходом к принятию решений, которая доказала свою эффективность в ряде областей. Под аргументацией обычно понимают процесс построения предположений, относительно некоторой анализируемой проблемы. Как правило этот процесс включает в себя обнаружение конфликтов и поиск путей их решения. В отличии от классической логики, аргументация предполагает, что могут быть доводы как «за», так и «против» некого предположения. В разделе 1 будут приведены основные понятия теории аргументации, в

1 Работа выполнена при финансовой поддержке РФФИ, проекты № 14-07-00862 и №15-01-05567 и проектной части государственного задания № 2.737.2014/К.

2 Суворов Александр Викторович, д.т.н., профессор, Заслуженный деятель науки и техники РФ, Финансовый университет при правительстве Российской Федерации, г. Москва, avsuvorov@list.ru.

3 Моросин Олег Леонидович, к.т.н., Национальный Исследовательский Университет «МЭИ», г. Москва, oleg@morosin.ru

4 Вагин Вадим Николаевич, д.т.н., профессор, Национальный Исследовательский Университет «МЭИ», г. Москва, vagin@appmat.ru

разделе 2 будет приведен упрощенный пример применения системы аргументации в системах сетевой безопасности.

1. Теоретические основы аргументации

Необходимость теории аргументации возникает из-за неполноты и недостоверности данных и знаний. Обычно, говоря об аргументации, выделяют три типа информации[6].

1) Объективная информация – информация, полученная из надежных источников, или которая может быть напрямую измерена или подтверждена. Например, утверждение «В центральной части России весной продолжительность светового дня увеличивается» является объективной информацией, подтверждаемое наукой и нашими наблюдениями. Такая информация, как правило, используется как неоспоримые аргументы.

2) Субъективная информация – информация полученная из менее надежных источников. Это могут быть некоторые предположения, суждения. Часто формулируются с помощью фраз «как правило», «обычно», «скорее всего». Такая информация и служит «источником» противоречий и конфликтов.

3) Гипотетическая информация. Она необходима для построения гипотез. Очень часто она является ложной информацией, и более того, даже может быть заранее неверной. Однако, построенные аргументы для ее опровержения могут быть полезны для других рассуждений. Например, маловероятно, что уровень мирового океана поднимется на метр в течении следующих 10 лет. Однако, такое предположение может быть полезным при планировании застройки прибрежных территорий, с учетом возможности их затопления. Часто при недостатке информации строятся те или иные гипотезы, и производится попытка доказать или опровергнуть их.

Все приведенные типы информации, могут быть использованы для аргументации. Объективная информация служит фактами и исходными посылками, субъективная информация является источником пересматриваемых выводов, а гипотетическая информация – помогает строить предположения.

Существуют несколько формализаций теории аргументации. Например, системы абстрактной аргументации, предложенные Дангом (Dung P.M.) [7], аргументационная система Лина и Шоэма (Lin F., Shoham Y.)[8], система Вресвийка (G.A.W. Vreswijk)[9], система аргументации Поллока(John L. Pollock)[10] и нектоторые другие.

Все эти подходы можно условно разделить на три типа[6].

1) Абстрактные системы, предложенные Дангом и позднее развиваемые Праккеном и Сартором. В этих системах аргументы представляются как элементы множества, в котором задано бинарное отношение «атака». В этих системах авторы полностью абстрагируются от внутренней структуры аргументов и природы множества аргументов. В таких системах отсутствует механизм получения новых аргументов, и задача сводится к поиску неконфликтующих аргументов на заданном множестве аргументов.

2) Согласованные системы (*coherence systems*). В таких системах основной стратегией обработки противоречий в базе знаний является выделение *согласованных* подмножеств из всего объема информации, имеющейся в базе знаний. Обычно такие системы базируютс я на классической логике, хотя возможны применения и модальных, временных или дескриптивных логик.

3) Системы пересматриваемых рассуждений. В таких системах обычно происходит введение пересматриваемого следствия в качестве элемента языка. То есть кроме импликации в обычном смысле вводится ее пересматриваемый аналог. Аргументы в таких системах представляются как последовательность рассуждений, ведущих к заключению, каждый шаг которых может подвергнуться поражению.

В данной работе будет подробно рассмотрена системы аргументации, основанная на пересматриваемых рассуждениях, предложенных Джоном Поллоком[10].

Пересматриваемые рассуждения. Прежде чем переходить к изложению основного материала, дадим необходимые определения и введем обозначения.

Определение 1. **Аргумент** - пара, состоящая из множества посылок и заключения[6]. Записывать такие пары будем в следующем виде p/X, где p заключение, а X множество посылок.

Например, аргумент $(p \rightarrow q)/\{\sim A, B\}$ означает, что из посылок $\sim A$, B следует $p \rightarrow q$. На всех иллюстрациях будем обозначать аргументы овалами. Для аргументов с пустым множеством посылок (такие аргументы называют **фактами**), будем писать только заключение. Например, фактом является утверждение, что Земля вращается вокруг Солнца.

Определение 2. **Интерес** – аргумент, который мы хотим обосновать в ходе монотонного и/или пересматриваемого вывода.

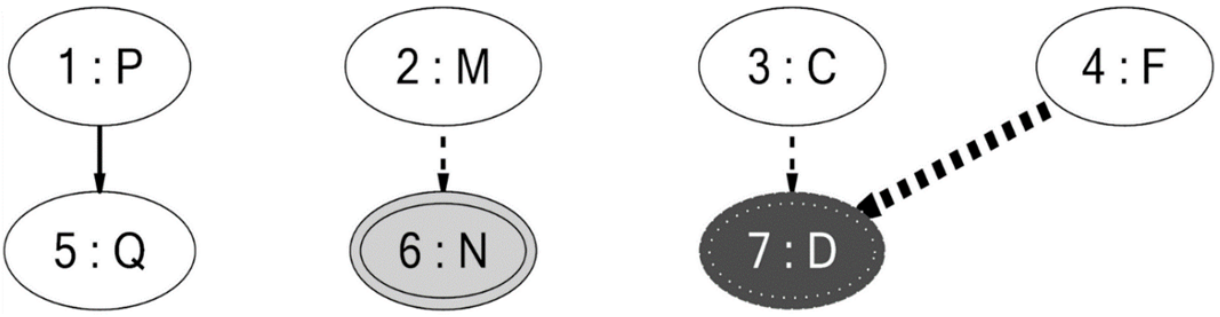

***Рис. 1**. Граф вывода*

Определение 3. **Граф вывода** – граф, показывающий взаимосвязи между аргументами и интересами. Он отображает, из каких аргументов порождается новый аргумент. Аналогично в нем показывается, из каких интересов получаются новые интересы. Кроме того, граф вывода отображает конфликты между аргументами при пересматриваемых рассуждениях.

Определение 4. **Дедуктивное следствие** – простые, дедуктивные правила вывода, означающие, что если истинно P, то истинно и Q. Такие правила не являются пересматриваемыми. Записывать такие правила будем так: $P \Rightarrow Q$. На графе вывода будем отображать их обыкновенными стрелками (см. аргументы P и Q на рис 1).

Определение 5. **Пересматриваемое следствие**. Это пересматриваемые правила вывода, которые могут быть получены, например, в результате индукции или абдукции. В данной работе нас не интересует конкретный механизм получения таких выводов, поэтому такие правила подаются декларативным образом на вход программы. Аргументы, полученные в результате таких выводов, будем называть пересматриваемыми. Записывать такие правила будем так: $M|\Rightarrow N$ На графе вывода такие связи будем отображать пунктирными стрелками, а пересматриваемые аргументы – двойным овалом (см. аргументы M и N на рис. 1).

Понятие конфликта – основа системы аргументации. Будем рассматривать два типа конфликтов – *опровержение* и *подрыв* [11] .

Определение 6. **Опровержение** (*rebutting*) – ситуация, когда некоторые аргументы опровергают заключения других аргументов. Иными словами аргумент $A_1=p_1/X_1$ опровергает аргумент $A_2=p_2/X_2$, когда заключение p_1 опровергает заключение p_2. Опровержение является симметричной формой атаки.

Определение 7. **Подрыв** (*undercutting*) – несимметричная форма атаки, когда один аргумент отрицает связь между посылками и заключением другого аргумента.

Определение 8. **Подрывающие доводы**. Это аргументы, поражающие связь между двумя другими аргументами, соединенными пересматриваемым следствием. Например, имеется аргумент F, подрывающий пересматриваемую связь $C|\Rightarrow D$ между аргументами C и D. Такие правила подрыва будем записывать в виде $F \Rightarrow (C\ @\ D)$. На графе вывода подрывающие аргументы и пораженные ими аргументы будем соединять жирной пунктирной стрелкой. Пораженные аргументы будем помечать темно-серым цветом (см. аргумент D на рис. 1).

На каждом шаге работы системы определение статусов каждого аргумента (поражается он или нет) играет ключевую роль. Введем необходимые для определения статусов поражения определения.

Аргумент называется *начальным*, если множество его предков пусто, то есть он задан изначально.

Базисом узла будем называть множество узлов, участвовавших в выводе этого узла.

Введём функцию, определяющую статус аргументов [10].

Функция σ назначает временный статус аргументам, дающая значения «пораженный» или «непораженный» подмножеству узлов графа вывода таким образом, что:

- σ назначает статус «непораженный» узлу n тогда и только тогда, когда σ присваивает статус «непораженный» всем узлам из базиса узла n и σ присваивает статус «пораженный» всем узлам, поражающим узел n.
- σ назначает статус «пораженный» узлу n тогда и только тогда, когда либо некоторым узлам из базиса n присвоен статус «пораженный», либо некоторым поражающим узел n узлам присвоен статус «непораженный».
- σ назначает окончательный статус аргумента n, если σ назначает временный статус и σ не уча-

ствует в назначении статусов другим аргументам, связанным с n.

Узел является непораженным, если все σ назначают ему статус «непораженный», иначе он пораженный.

Заключение обосновано в данный момент рассуждений тогда и только тогда, когда его поддерживают непораженные аргументы. Однако дальнейшие рассуждения могут выявить еще какие-либо значимые аргументы, которые меняют статус заключения на необоснованное или наоборот. При наличии ряда посылок и массива выводов и правил вывода можно сказать, что высказывание подтверждено тогда и только тогда, когда граф вывода, построенный на множестве всех возможных аргументов, содержит непораженный узел, соответствующий заключению. Подтвержденные высказывания являются «окончательно обоснованными» заключениями, которые система и стремится определить.

Степени обоснования в пересматриваемых рассуждениях. Прежде чем переходить к способам и алгоритмам вычисления степеней обоснования, рассмотрим как они могут задаваться и откуда получены. В данной статье для задания степеней обоснования используется числовая шкала [0,1], где 0 соответствует пораженному аргументу, 1 наиболее обоснованному аргументу. Степени обоснования могут быть двух типов:

1) степени обоснования исходных аргументов;

2) степени обоснования пересматриваемых правил.

Первый тип степеней обоснования присваивается каждому исходному аргументу, и представляет собой некою оценку достоверности источника, из которого получен данный аргумент. Например, по телевизору сказали, что вероятность осадков 70%. Соответственно мы можем построить аргумент *A1:Завтра(дождь)* со степенью обоснования 0.7. Степени обоснования будем записывать функцией *Jus(A)*. То есть для приведенного примера *Jus(Завтра(дождь))=0.7*. Конкретные механизмы получения степеней обоснования зависят прежде всего от предметной области. Например, это могут быть статистические данные (в 90% этот источник дает верные данные) или экспертные оценки (вероятность роста акций 60%).

Второй тип степеней обоснования связан с пересматриваемыми правилами. Как уже говорилось выше, часто пересматриваемые правила появляются в результате формализации знаний эксперта вида «Если А, то чаще всего В». Такие правила так же могут нести в себе некоторую числовую оценку.

Например, применение анальгина в 85% приводит к снижению температуры тела пациента (формально *R1*: (∀ *x*) *прием*(*анальгин, x*)|=> *понижение_температуры*(*x*)).

Одновременное использование обоих типов степеней обоснования является довольно сложной задачей и требует дополнительных исследований. В данной статье ограничимся рассмотрением степеней обоснования первого типа – для изначально заданных аргументов.

Итак, необходимо задать функцию *Jus(A)* для вычисления любого из аргументов в графе вывода. Будем считать, что для изначальных аргументов эта величина является определенной. На значение этой функции будут оказывать влияние два фактора – дерево вывода аргумента (т.е. степень обоснования аргументов, которые использовались в выводе данного аргумента) и конфликты с другими аргументами. Для удобства рассмотрим эти два фактора раздельно: $Jus_{anc}(A)$ – унаследованная степень обоснования и $Jus_{con}(A)$ – на сколько конфликт подрывает обоснование аргумента. Пусть $\mathrm{Anc} = \{\mathrm{Anc_i}\}$, $\mathrm{i} \in 1..\mathrm{n}$ – множество аргументов $\mathrm{Anc_i}$, учувствовавших в выводе аргумента A, n – количество таких аргументов.

$$Jus_{anc}(A)=min(\{Jus(A1), Jus(A2)\ldots Jus(An)\}), \quad (1.1)$$

где $A_1, A_2,\ldots,A_n$ – аргументы, использовавшиеся при выводе A.

Формулу (1.1) называют принципом слабейшей связи [12]. Надо отметить, что это не единственных подход к вычислению степени обоснования, в ряде работ применяется байесовский подход (см., например, [13]).

Отметим, что из формулы (1.1) следует, что, если производить вычисление степеней обоснования рекурсивно, начиная от исходно заданных аргументов, то можно искать минимум, не из всех аргументов в базисе, а только на предыдущем шаге. Таким образом, если у аргумента один предок, то его унаследованная степень обоснования будет равна степени обоснования его предка.

Если при вычислении Jus_{anc} ищутся наиболее слабые аргументы, то при определении того, насколько конфликт уменьшает обоснования, используется наиболее сильные аргументы. Пусть A_{confl} – множество аргументов, вступающих в конфликт с A, n=| A_{confl} |, тогда

$$\boldsymbol{Jus_{con}(A)} = \begin{cases} \mathbf{max}_{i\in 1..n}\{\boldsymbol{Jus_{anc}(Aconfl_i)}\}, \boldsymbol{n > 0}; \\ \mathbf{0}, \text{в противном случае.} \end{cases} \quad (1.2)$$

В формуле (1.2) используется Jusanc для того, чтобы верно обрабатывать случаи, когда между аргументами есть конфликт типа опровержение.

Итак, окончательно:

$$Jus_{con}(A) = \begin{cases} \max_{i \in 1..n}\{Jus_{anc}(Aconfl_i)\}, n > 0; \\ 0, \text{в противном случае.} \end{cases} \quad (1.3)$$

Данная модель степеней обоснования была успешно реализована и протестирована на многих задачах аргументации. Приведем пример того, как разрабатываемая система аргументации может быть применена для улучшения систем сетевой безопасности.

2. Применение аргументации в системах сетевой безопасности

Рассмотрим пример задачи сетевой безопасности, приведенной в [5]. Для наглядности постановка задачи будет несколько упрощена.

Пусть имеется сложная информационная система, защищенная некоторой системой безопасности. Система безопасности в случае выявления подозрительной активности сигнализирует о появившейся угрозе, её типе и оценке вероятности реальности угрозы. Предположим, что в системе имеются несколько открытых портов и на 80 порте запущен веб сервис для обработки запросов клиентов. По умолчанию при обнаружении угрозы на одном из портов, данный порт блокируется. В результате блокировки порта все службы, использующие данный порт должны быть остановлены. Если будет остановлен веб сервис, то возникнет критическая ошибка и компания понесет серьезные издержки. Если угроза не велика – нельзя допустить возникновения критических ошибок в результате применения защитных мер. Сетевые черви относятся к классу не очень опасных сетевых воздействий. Предположим, что система безопасности обнаружила на 80 порту подозрительную активность, похожую на атаку сетевого червя.

На формальном языке данная задача примет следующий вид, где A1-A5 – исходные аргументы, R1 – подрывающее правило, R2 – пересматриваемое правило вывода:

A1: $attack(port\_80,\ warm)$ – обнаружена подозрение на атаку сетевого червя на 80 порту;

A2: $use(web\_service,\ port\_80)$ – 80 порт используется веб-сервисом;

A3: $\forall x\ \forall y\ (block(x) \& use(y,\ x) \rightarrow stop(y))$ – при блокировки порта, все службы, использующие данный порт должны быть остановлены;

A4: $stop(web\_service) \rightarrow critical\_error$ – остановка веб-сервиса является критической ошибкой.

A5: $\sim serious\_attack(warm)$ – сетевые черви относятся к классу не очень опасных сетевых воздействий;

R1:$\forall y \sim serious\_attack(y) \& critical\_error \Longrightarrow \forall x\ attack(x,y)\ @\ block(x)$ – подрывающее правило, утверждающее, что если угроза не велика – нельзя допустить возникновения критических ошибок в результате применения защитных мер;

R2: $\forall x \forall y\ attack(x,y)) \mid\Rightarrow block(x)$ – пересматриваемое правило, утверждающее, что по умолчанию при обнаружении угрозы на одном из портов, данный порт блокируется.

На рис. 2 представлен граф вывода для данной задачи. Разберём ход решения данной задачи по шагам. Аргументы A1-A5 – заданы изначально. Пересматриваемый аргумент A6:$block(port\_80)$ получается с помощью аргумента A1: $attack(port\_80, warm)$ и пересматриваемого правила R2:$\forall x \forall y\ attack(x,y)) \mid\Rightarrow block(x)$. Аргументы A7 и A8 получаются с помощью сколемизации из аргумента A5 (переменные, связанные квантором всеобщности заменяются на свободные переменные, обозначаемые $\_x$ и $\_y$ соответственно). Аргумент A9: $block(\_x) \rightarrow (use(\_y,\ \_x) \rightarrow stop(\_y))$ получается из аргумента

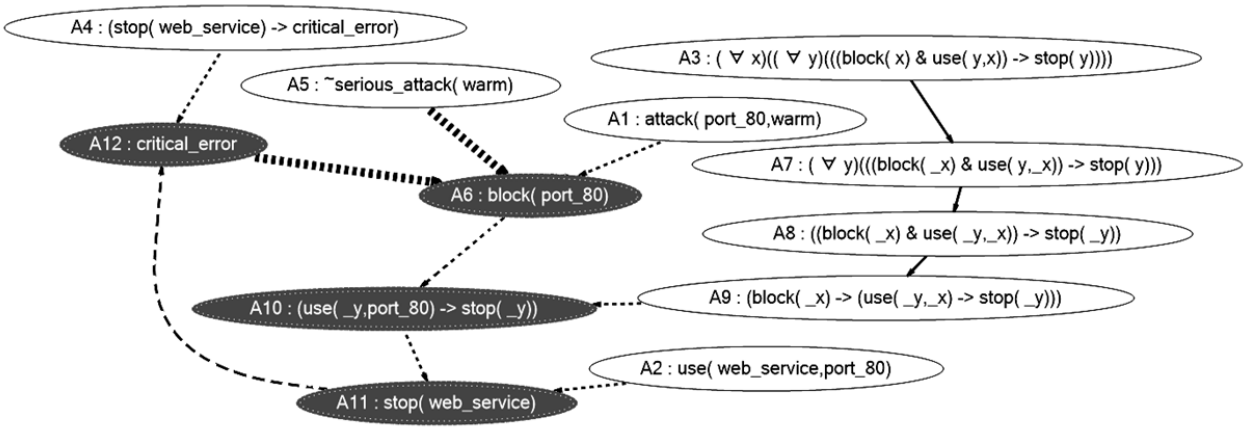

***Рис. 2**. Пример применения аргументации для принятия решения о блокировке*

A8: $block(\_x)$ & $use(\_y, \_x) \rightarrow stop(\_y)$ с помощью правила разъединения посылок. Пересматриваемый аргумент A10: ($use(\_y, port\_80)$ -> $stop(\_y)$) получается из пересматриваемого аргумента A6: $block(port\_80)$ и аргумента A9: $block(\_x) \rightarrow (use(\_y, \_x) \rightarrow stop(\_y))$ по правилу Modus Ponens. Пересматриваемый аргумент A11: $stop(web\_service)$ получается из пересматриваемого аргумента A2: $use(web\_service, port\_80)$ и аргумента A10: ($use(\_y,port\_80)$ -> $stop(\_y)$) по правилу Modus Ponens.

Пересматриваемый аргумент A12: $critical\_error$ получается из пересматриваемого аргумента A4: $stop(web\_service) \rightarrow critical\_error$ и аргумента A11: $stop(web\_service)$ по правилу Modus Ponens. Аргументы A12: $critical\_error$ и A5: $\sim serious\_attack(warm)$ по подрывающему правилу R1:$\forall y \sim serious\_attack(y)$& $critical\_error$ ==>$\forall x\ attack(x,y)$ @ $block(x)$ подрывают пересматриваемую связь между аргументами A1: $attack(port\_80, warm)$ и A6: $block(port\_80)$, делая аргумент A6 пораженным.

В результате решения данной задачи методами аргументации на основе пересматриваемых рассуждений, аргумент A6: $block(port\_80)$ стал пораженным, то есть система приняла решение, что возможная атака сетевого червя не 80 порт не настолько велика, чтобы принять решение о блокировка всего трафика на данном порту, так как это приведет к значительным издержкам.

Заключение

Предложенная идея применения аргументации в системах сетевой безопасности, по мнению авторов, позволит сделать данные системы более гибкими и даст возможность оценивать целесообразность применения тех или иных защитных механизмов. Для упрощения примера, в нем не использовались числовые оценки вероятностей обнаружения уязвимости, критичности тех или иных подсистем, и опасности применения средств защиты, но в разделе 1 описан способ обработки такой информации путём введения степеней обоснования в систему аргументации. Применение аргументации со степенями обоснования позволит давать числовые оценки вырабатываемых системой рекомендаций, тем самым позволит решить еще одну важную задачу – задачу выбора способа реагирования на имеющуюся в системе подозрительную активность.

Рецензент: *Плесневич Геральд Станиславович, кандидат физико-математических наук, профессор кафедры ПМ ФБГОУ ВО НИУ «МЭИ», PlesnevichGS@mpei.ru.*

Литература

1. Защита информации в информационных системах: учебное пособие / Евсеев В.Л., Суворов А.В. – Москва: 2016. – 275 с.
2. Ou X., Govindavajhala S., Appel A. W. MulVAL: A Logic-based Network Security Analyzer //USENIX security. – 2005.
3. Eronen P., Zitting J. An expert system for analyzing firewall rules //Proceedings of the 6th Nordic Workshop on Secure IT Systems (NordSec 2001). – 2001. – С. 100-107.
4. Khosravifar B., Bentahar J. An experience improving intrusion detection systems false alarm ratio by using honeypot //Advanced Information Networking and Applications, 2008. AINA 2008. 22nd International Conference on. – IEEE, 2008. – С. 997-1004.
5. Bandara, A. K., Kakas, A., Lupu, E. C., & Russo, A. (2006). Using argumentation logic for firewall policy specification and analysis. In Large Scale Management of Distributed Systems (pp . 185-196). Springer Berlin Heidelberg
6. Philippe Besnard and Anthony Hunter, "Elements of argumentation", MIT press, 2008, 298 p.
7. Bondarenko A., Dung P.M., Kowalski R.A., Toni F. "An abstract argumentation-theoretic framework for defeasible reasoning", Ibid. 1997, V. 93(1-2) pp. 63-101.
8. Lin F., Shoham Y. Argument systems. A uniform basis for nonmonotonic reasoning// Proc. Of the First Int. Conf. on Principles of Knoledge Representation and Reasoning. San Mateo, CA: Morgan Kaufmann Publishers Inc, 1989, pp. 245-355.
9. Vreeswijk G.A.W. "Abstract argumentation systems". Artificial Intelligence 1997. V. 90, pp 225-279.
10. John L. Pollock, "How to Reason Defeasibly," Artificial Intelligence 57, 1992, pp. 1-42.
11. В.Н. Вагин, Е.Ю. Головина, А.А. Загорянская, М.В. Фомина "Достоверный и правдоподобный вывод в интеллектуальных системах" / Под ред. В.Н. Вагина, Д.А. Поспелова. 2-е издание дополненное и исправленное. ФИЗМАТЛИТ, 2008. 712 с.
12. John L. Pollock "Defeasible reasoning with variable degrees of justification", Artificial Intelligence 2001. V. 133, pp 233-282.
13. R. Haenni, J. Kohlas, N. Lehmann. "Probabilistic Argumentation Systems", 1999, Handbook of Defeasible Reasoning and Uncertainty Management Systems, Dordrecht: Volume 5: Algorithms for Uncertainty and Defeasible Reasoning, Kluwer, pp. 221–287

APPLICATION OF THE CASE TO THE PROBLEMS OF INFORMATION SECURITY

Suvorov A.V.[5], Morosin O.L.[6], Vagin V.N.[7]

This paper discusses the ideas and approaches to the use of reasoning in the information security problems. The solution of network security issues is a complex task, involving a large number of factors and requires a finding of a reasonable compromise between maintaining security, stability, increase operating costs and limitations in terms of functionality of complex information systems. In this article, we propose to use arguments to provide automated support for security solutions. Classical methods of inference do not have the mechanism of «revision» of earlier findings and there is no ability to detect and resolve conflicts in knowledge. One way of handling conflict situations and conflicting information is the use of reasoning machine. The argument gives more opportunities for simulation of plausible reasoning. In this paper we used reasoning system based on revised by reasoning. Proposed work approach allows to give the numerical evaluation of the quality of recommendations produced by the system, thus allowing to solve the important task – the task of choosing a way to respond to the existing suspicious activity in the system. Furthermore, in an example of the processing of dangerous situations arising in the system.

***Keywords**: information security, reasoning, arguments are reviewed, the degree of justification, the resolution of contradictions.*

References

1. Zashchita informatsii v informatsionnykh sistemakh: uchebnoye posobiye/ Yevseyev V.L., Suvorov A.V. – Moskva: 2016. – 275 s.
2. Ou X., Govindavajhala S., Appel A. W. MulVAL: A Logic-based Network Security Analyzer //USENIX security. – 2005.
3. Eronen P., Zitting J. An expert system for analyzing firewall rules //Proceedings of the 6th Nordic Workshop on Secure IT Systems (NordSec 2001). – 2001. – S. 100-107.
4. Khosravifar B., Bentahar J. An experience improving intrusion detection systems false alarm ratio by using honeypot //Advanced Information Networking and Applications, 2008. AINA 2008. 22nd International Conference on. – IEEE, 2008. – S. 997-1004.
5. Bandara, A. K., Kakas, A., Lupu, E. C., & Russo, A. (2006). Using argumentation logic for firewall policy specification and analysis. In Large Scale Management of Distributed Systems (pp. 185-196). Springer Berlin Heidelberg
6. Philippe Besnard and Anthony Hunter, «Elements of argumentation», MIT press, 2008, 298 p.
7. Bondarenko A., Dung P.M., Kowalski R.A., Toni F. «An abstract argumentation-theoretic framework for defeasible reasoning», Ibid. 1997, V. 93(1-2) pp. 63-101.
8. Lin F., Shoham Y. Argument systems. A uniform basis for nonmonotonic reasoning// Proc. Of the First Int. Conf. on Principles of Knoledge Representation and Reasoning. San Mateo, CA: Morgan Kaufmann Publishers Inc, 1989, pp. 245-355.
9. Vreeswijk G.A.W. «Abstract argumentation systems». Artificial Intelligence 1997. V. 90, pp 225-279.
10. John L. Pollock, «How to Reason Defeasibly,» Artificial Intelligence 57, 1992, pp. 1-42.
11. V.N. Vagin, Ye.YU. Golovina, A.A. Zagoryanskaya, M.V. Fomina «Dostovernyy i pravdopodobnyy vyvod v intellektual'nykh sistemakh» / Pod red. V.N. Vagina, D.A. Pospelova. 2-ye izdaniye dopolnennoye i ispravlennoye. FIZMATLIT, 2008. 712 s.
12. John L. Pollock «Defeasible reasoning with variable degrees of justification», Artificial Intelligence 2001. V. 133, pp 233-282.
13. R. Haenni, J. Kohlas, N. Lehmann. «Probabilistic Argumentation Systems», 1999, Handbook of Defeasible Reasoning and Uncertainty Management Systems, Dordrecht: Volume 5: Algorithms for Uncertainty and Defeasible Reasoning, Kluwer, pp. 221–287

5 Alexander Suvorov, Dr.Sc., Prof., Financial University under the Government of the Russian Federation, Moscow, avsuvorov@list.ru.
6 Oleg Morosin, PhD, National Research University «MEI», Moscow, oleg@morosin.ru.
7 Vadim Vagin, Dr.Sc., Prof., National Research University «MEI», Moscow, vagin@appmat.ru.