УДК 004.056.53

Д.И. Котенко, И.В. Котенко, И.Б. Саенко

МЕТОДИКА ИТЕРАЦИОННОГО МОДЕЛИРОВАНИЯ АТАК В БОЛЬШИХ КОМПЬЮТЕРНЫХ СЕТЯХ

Котенко Д.И., Котенко И.В., Саенко И.Б. **Методика итерационного моделирования атак в больших компьютерных сетях.**

Аннотация. В статье рассматриваются основные компоненты методики итерационного моделирования атак в больших компьютерных сетях, которыми являются формальная модель, алгоритмы анализа вероятностных графов атак и программные средства их реализации. Формальная модель итерационного моделирования атак включает модели процессов определения задач моделирования, построения моделей атак, запуска моделей и анализа результатов моделирования атак. Алгоритмы анализа вероятностных графов атак обеспечивают расчет метрик защищенности и нахождение подграфов атак, ассоциированных со сценариями действий нарушителей. Программные средства анализа моделей атак для больших компьютерных сетей обеспечивают их статический анализ и анализ динамических характеристик.

Ключевые слова: моделирование атак, большая компьютерная сеть, граф атак, анализ защищенности.

Kotenko D.I., Kotenko I.V., Saenko I.B. **Methodology of iterative attack modelling in large computer networks.**

Abstract. The paper presents the basic components of the methodology of iterative attack modelling in large computer networks, which constitues formal model, analysis algorithms for probabilistic attack graphs and software. Formal model of iterative attack modelling process involves the process models of task definition modelling, attack model building, model execution and model result analysis. Probabilistic attack graph analysis algorithms provide calculation of security metrics and finding attack sub-graphs associated with intruder action scripts. Software tools for attack model analysis in large computer networks provide their static analysis and analysis of dynamic characteristics.

Keywords: attack modeling, large computer networks, attack graph, security analysis.

1. Введение. Моделирование атак в компьютерных сетях в настоящее является достаточно перспективным и эффективным направлением в области анализа защищенности информации и оценки рисков нарушения ее безопасности применительно к современным инфокоммуникационным системам. Несомненным достоинством моделирования атак как метода научного исследования является его инвариантность по отношению к структуре и составу анализируемой компьютерной сети, а также к характеру поведения нарушителей безопасности информации и сетевых информационных процессов [1–3].

Большие компьютерные сети как объекты моделирования атак выделяются в отдельный класс. Задачи обеспечения безопасности информации, которые достаточно успешно решаются с использованием разнообразных инструментальных средств в малых и средних компью-

терных сетях, не удается также эффективно решать для больших сетей. Сложности моделирования атак в больших компьютерных сетях связаны с неполнотой и неопределенностью информации, доступной для использования средствами моделирования, а также большой вычислительной сложностью алгоритмов построения и анализа моделей атак.

Помимо этого, большим компьютерным сетям присущ ряд характерных особенностей, обусловливающих высокую сложность моделирования атак. К их числу, например, можно отнести следующие факторы: высокую сложность и неопределенность структуры сети; сильную неоднородность сетевых устройств, используемых для построения сети; слабую регламентированность действий пользователей сети; высокую сложность администрирования безопасности и другие [4]. В результате моделирование атак в больших компьютерных сетях, например, в Интернет, приходится многократно повторять для получения все более точных моделей, постепенно меняя различные их параметры.

Результатом этого является тот факт, что моделирование атак в больших компьютерных сетях является итерационным процессом, который сам по себе имеет значительный научный интерес и по этой причине требует разработки формальной модели и методики своего проведения.

Настоящая работа посвящена рассмотрению авторской методики моделирования атак и анализа защищенности больших компьютерных сетей, предусматривающей итерационный характер моделирования при активном участии специалистов в подготовке и переопределении требований для каждой итерации. Тем самым работа развивает ряд результатов, полученных авторами в области анализа защищенности и моделирования атак в компьютерных системах и сетях [4–14], а также в области управления информацией и событиями безопасности [15, 16]. Рассматриваются ключевые составляющие разработанной методики, каковыми являются формальная модель итерационного процесса моделирования атак, а также алгоритмы и программные средства анализа моделей атак, формируемых на каждой итерации моделирования.

2. Формальная модель итерационного процесса моделирования атак. В силу указанной выше специфики больших компьютерных сетей процесс моделирования атак, протекающих в этих сетях, состоит из итераций, включающих следующие группы взаимосвязанных процессов: определение (переопределение) задачи, построение модели, запуск модели, анализ результатов [4].

По этой причине формальную модель итерационного процесса моделирования атак для больших компьютерных сетей целесообразно представить четверкой, имеющей следующий вид:

$$AMD = \langle TD, AMB, ME, MRA \rangle,$$

где TD — множество процессов определения задач моделирования, AMB — множество процессов построения моделей атак, ME — множество процессов запуска моделей, MRA — множество процессов анализа результатов моделирования атак.

Рассмотрим подробнее состав множеств TD, AMB, ME и MRA.

Процессы определения задач моделирования (TD). Множество процессов определения задач представляется в виде тройки $TD = \langle RP, SDR, NA \rangle$, где RP — множество процессов подготовки требований, необходимых для остальных процессов моделирования атак; SDR — множество процессов получения и структурирования исходных данных об анализируемой сети; NA — множество процессов приобретения знаний, позволяющих управлять процессами построения и анализа моделей атак. Рассмотрим более детально каждое из этих трех множеств.

Множество процессов подготовки требований RP определяется, прежде всего, существующими стандартами, связанными с обеспечением информационной безопасности, в частности, требованиями руководящих документов ФСТЭК России [17–19]. Также требования могут быть основаны на принципе «разумной достаточности» [20], включающем следующий набор утверждений:

абсолютно непреодолимой защиты создать невозможно;

необходимо соблюдать баланс между затратами на защиту и получаемым эффектом, в том числе и экономическим, заключающимся в снижении потерь от нарушений безопасности;

стоимость средств защиты не должна превышать стоимости защищаемой информации (или других ресурсов — аппаратных, программных);

затраты нарушителя на несанкционированный доступ к информации должны превышать тот эффект, который он получит, осуществив подобный доступ.

В связи с итерационным характером процессов моделирования множество процессов подготовки требований RP разделяется на подмножества и представляется в следующем виде:

$$RP = \langle RP_1, RP_2, ..., RP_n, ..., RP_m \rangle,$$

где n — номер итерации, m — количество итераций. Структура RP приведена на рис. 1. На каждой итерации RP_n объединяет три множества процессов следующим образом: $RP_n = \langle RP_{n-1}, ORC_n, NRP_n \rangle$, где RP_{n-1} — процессы подготовки требований для предыдущей итерации; ORC_n — множество процессов корректировки требований, полученных в начале предыдущей $(n-1)$-й итерации; NRP_n — множество процессов подготовки новых требований для текущей n-й итерации. В первой итерации $RP_1 = \langle NRP_1 \rangle$.

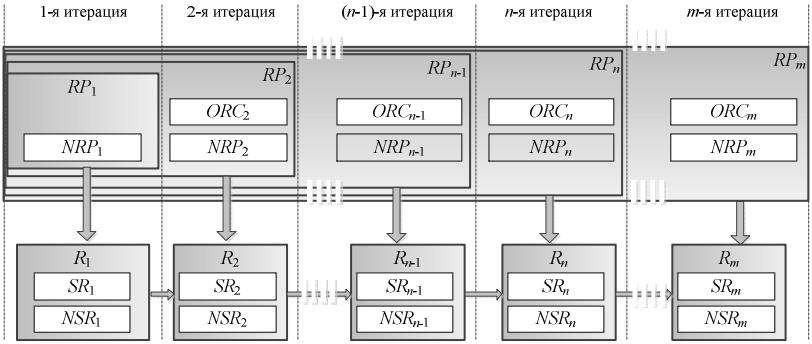

Рис. 1. Структура множества процессов подготовки требований.

В результате процессы RP формируют множество $R = \langle R_1, R_2, ..., R_n, ..., R_m \rangle$, состоящее из m версий требований к процессам моделирования, где n — номер версии, совпадающий с номером итерации. Причем при $n > 1$ версия R_n основывается на требованиях, полученных на предшествующей итерации, и $R_n = \langle SR_n, NSR_n \rangle$, где SR_n — стандартизованные требования, NSR_n — требования, основанные на принципе «разумной достаточности».

Множество процессов получения и структурирования исходных данных SDR, с одной стороны, тесно связано с множеством RP, а с другой — определяется доступностью источников информации об анализируемой сети. Аналогично множеству RP, множество SDR представляется в виде

$$SDR = \langle SDR_1, SDR_2, ..., SDR_n, ..., SDR_m \rangle,$$

где n — номер итерации, m — количество итераций. На каждой итерации получение исходных данных объединяет три множества процессов, что может быть представлено следующим образом: $SDR_n = \langle SDR_{n-1}, SDC_n, NDR_n \rangle$, где SDR_{n-1} — процессы получения и структурирования исходных данных для предыдущей итерации; SDC_n — множество процессов корректировки исходных данных, полученных в начале предыдущей $(n-1)$-й итерации; NDR_n — множество процессов получения новых данных для текущей n-й итерации. В первой итерации $SDR_1 = \langle NDR_1 \rangle$.

Для хранения и обработки данных обычно используют базы данных, в которых наиболее распространенной в настоящее время является реляционная модель данных. Однако, в общем случае, на каждой итерации может использоваться своя модель данных, отличная от модели, используемой на предыдущей итерации. Даже в рамках одной модели данных структуры, используемые для хранения и обработки данных, могут различаться. Поэтому SDR_n, в общем случае, включает в себя процессы конвертации данных.

Множество процессов приобретения знаний NA, прежде всего, зависит от доступности первичных источников знаний, то есть экспертов в области моделирования атак в компьютерных сетях. При этом множества RP и SDR должны обеспечивать множество NA вторичными источниками знаний, в качестве которых могут выступать базы данных и текстовые документы, содержащие информацию об анализируемой сети. Множество NA имеет структуру, аналогичную множествам RP и SDR, и представляется в виде

$$NA = \langle NA_1, NA_2, ..., NA_n, ..., NA_m \rangle,$$

где n — номер итерации, m — количество итераций. На каждой итерации приобретение знаний объединяет три множества процессов и представляется в виде $NA = \langle NA_{n-1}, NC_n, NNA_n \rangle$, где NA_{n-1} — множество процессов приобретения знаний для предыдущей итерации; NC_n — множество процессов корректировки знаний, приобретенных в начале предыдущей $(n-1)$-й итерации; NNA_n — множество процессов приобретения новых знаний для текущей n-й итерации. В первой итерации $NA_1 = \langle NNA_1 \rangle$. Важно отметить, что, в общем случае, на каждой итерации возможна своя модель представления знаний, отличная от

модели, используемой на предыдущей итерации. Поэтому NC_n может включать процессы извлечения знаний из баз знаний.

Процессы построения моделей атак (AMB) могут быть декомпозированы различным образом в зависимости от предпочтительных классификаций атак, классификаций методов и средств моделирования атак. Так как в предлагаемой методике основной акцент делается на сокращение вычислительной сложности алгоритмов, описывающих процессы моделирования атак, множество процессов построения моделей атак AMB декомпозируется следующим образом:

$$AMB = \langle AMB_1, AMB_2, ..., AMB_n, ..., AMB_m \rangle,$$

где n — номер итерации, m — количество итераций.

Структура множества AMB представлена на рис. 2.

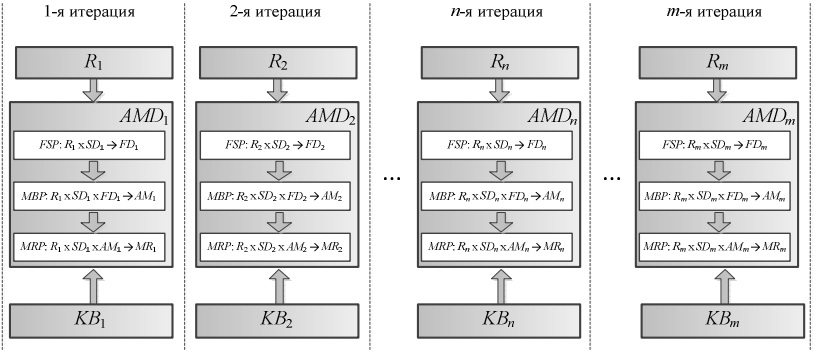

Рис. 2. Структура множества процессов построения моделей атак.

На каждой итерации соответствующее множество процессов рассматривается в виде следующего набора функций:

$$AMB_n = \langle FSP(R_n, SD_n), MBP(R_n, SD_n, fd_n), MRP_n(R_n, SD_n, am_n) \rangle,$$

где $FSP : R \to FD$ — функция, обеспечивающая выбор формализма для внутреннего представления модели атак, используемого средством моделирования; $MBP : R \times SD \times FD \to AM$ — функция, применяемая для построения моделей атак в виде внутренней структуры с использованием различных способов сокращения вычислительных затрат; $MRP_n : R \times SD \times AM \to MR_n$ — функция, предназначенная для формирования внешних представлений модели атак при выполнении итерации с номером n, используемых для различных видов анализа модели

атак, визуализации модели или для обмена данными модели между различными инструментальными средствами.

Функции в наборе AMB_n определяют отношения между следующим множествами:

множеством R, включающим все возможные требования к построению моделей атак для анализируемой сети;

множеством SD, которое включает все данные об анализируемой сети, в том числе, данные о логической и физической организации, оборудовании сети, программном обеспечении, методах кодирования, принципах функционирования, особенностях пользовательского интерфейса и другие;

множеством FD, которое содержит все известные в настоящее время формализмы, которые могут быть использованы для формального представления моделей атак;

множеством AM, которое содержит модели атак анализируемой сети, построенные в течение всех m итераций;

множеством MR_n, которое включает все возможные представления модели am_n атак, включая как формальные представления, так и неформальные.

Аргументами набора функций AMB_n являются следующие множества:

R_n — множество требований, полученных в результате выполнения процессов RP_n, причем $R_n \subset R$;

SD_n — множество исходных данных, полученных в результате выполнения процессов SDR_n, причем $SD_n \subset SD$;

fd_n — выбранный формализм для внутреннего представления модели атак, причем $fd_n \in FD$;

am_n — модель атак, причем $am_n \in AM$.

Результатами процессов построения моделей атак являются следующие элементы:

fd_n — выбранный формализм для внутреннего представления модели атак;

am_n — модель атак;

$\{mr_{ni} | i \in [1..k]\}$ — множество из k различных представлений модели атак am_n, причем $mr_{ni} \in MR_n$.

Для управления множеством процессов AMB может использоваться база знаний KB_n, построенная в результате выполнения процессов NA_n. Наличие базы знаний позволяет применять различные эвристики для сокращения перебора при применении функций MBP и MBP_n. Результатом использования эвристик является выделение среди всех связей между элементами множеств R, SD, FD, AM и MR только наиболее существенных связей с точки зрения экспертов, знания о которых заложены в базу знаний.

В качестве способа внутреннего представления моделей полагается целесообразным использовать вероятностные графы атак, поскольку они позволяют учитывать неопределенность с помощью вероятностных оценок, сопоставленных со связями в графах, а неполноту — с помощью ограничений, накладываемых на эти оценки, а также на любые значения параметров, сопоставленных с вершинами графа. Не исключается расширение методики таким образом, что для представления моделей атак будут использованы другие формализмы, например, сети Петри.

Под *вероятностным графом атак* будем понимать ациклический ориентированный граф $G = \langle S, E, S_s, s_g, P, L \rangle$, где $S = \{s_i | i \in [1..k]\}$ — множество из k вершин, в котором каждая вершина рассматривается как *элементарное условие возникновения инцидента*; $E : S \to S$ — множество дуг, которое можно рассматривать как *отношение перехода* между вершинами, моделирующее конъюнкцию или дизъюнкцию элементарных условий возникновения инцидента; $S_s \subset S$ — множество *начальных состояний*; $s_g \in S$ — *целевая вершина*, которую можно ассоциировать с конкретной атакой; $P = \{p_i | i \in [1..k-1]\}$ — множество *вероятностных оценок* возникновения инцидентов; $L : E \to P$ — функция *разметки дуг*, которая для каждой дуги графа задает оценки вероятностей возникновения инцидентов.

Каждая вершина s_i ассоциируется с логическим утверждением, которое может быть представлено тройкой $\langle x, V, H \rangle$, где x — наименование переменной, V — область определения переменной, H — множество, описывающее ограничения на значения переменной x, причем $H \subset V$. Таким образом, если переменная x принимает значе-

ние, принадлежащее множеству H, то соответствующее утверждение s_i принимает значение «истина», иначе — значение «ложь».

Вероятностная оценка p_i может быть определена либо с помощью одиночного коэффициента неопределенности, либо интервалом неопределенности $p_i = [Low, High]$.

В некоторых случаях для моделирования атак в компьютерных сетях может оказаться достаточным использовать байесовские графы атак, в которых вероятностные оценки задаются одиночными коэффициентами неопределенности.

Процессы запуска моделей (ME). Множество процессов запуска моделей ME декомпозировано следующим образом:

$$ME = \langle ME_1, ME_2, ..., ME_n, ..., ME_m \rangle,$$

где n — номер итерации, m — количество итераций. Структура множества ME аналогична структуре AMB и представлена на рис. 3.

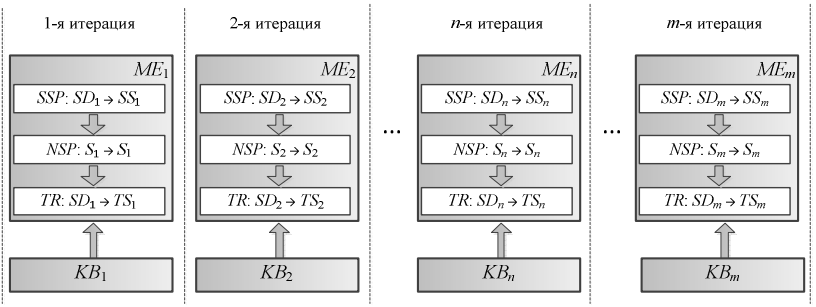

Рис. 3. Структура множества процессов запуска моделей атак.

На каждой итерации соответствующее множество процессов рассматривается в виде набора элементов $ME = \langle SSP_n, NSP_n, TR_n \rangle$, где $SSP_n : SD_n \to SS_n$ — функция формирования начального состояния модели am_n; $NSP_n : S_n \to S_n$ — функция, обеспечивающая переход от одного к другому состоянию модели am_n; $TR_n : SD_n \to TS_n$ — функция, обеспечивающая переход модели am_n в завершающее состояние; S_n — множество всех возможных состояний модели am_n; SS_n — множество возможных начальных состояний модели am_n, причем

$SS_n \subset S_n$; TS_n — множество возможных конечных состояний модели am_n, причем $TS_n \subset S_n$.

Использование базы знаний KB_n позволяет применять различные эвристики для сокращения перебора при переходе модели от одного состояния к другому. Результатом использования эвристик является выделение среди всех связей между элементами множества S_n только наиболее существенных связей с точки зрения экспертов, знания которые заложены в базу знаний.

Процессы анализа результатов моделирования атак (MRA). Множество процессов анализа результатов моделирования атак MRA представлено следующим образом:

$$MRA = \left\langle MRA_1, MRA_2, ..., MRA_n, ..., MRA_m \right\rangle,$$

где n — номер итерации, m — количество итераций. Структура множества MRA представлена на рис. 4.

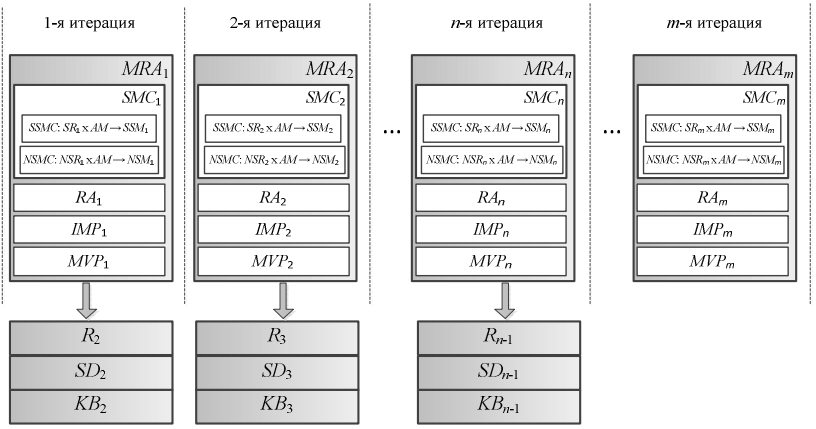

Рис. 4. Структура множества процессов анализа результатов моделирования атак.

На каждой итерации соответствующее множество процессов рассматривается в виде следующего набора составляющих элементов: $MRA_n = \left\langle SMC_n, RA_n, IMP_n, MVP_n \right\rangle$, где SMC_n — множество процессов расчета метрик защищенности; RA_n — множество процессов анализа рисков на основе принципа «разумной достаточности»; IMP_n — мно-

жество процессов построения моделей нарушителей; MVP_n — процессы визуального представления результатов моделирования (отображения данных на экране компьютера в различной форме или печати отчетов), позволяющие пользователю принять решения по обеспечению безопасности сети, сформировать новые требования R_{n+1}, подготовить обновленные данные SD_{n+1} и создать обновленную базу знаний KB_{n+1} для следующей итерации моделирования.

Множество SMC_n предлагается рассматривать как набор из элементов: $SMC_n = \langle SSMC_n, NSMC_n \rangle$, где $SSMC_n : SR_n \times AM \to SSM_n$ — функция расчета метрик защищенности, связанных со стандартными требованиями; $NSMC_n : NSR_n \times AM \to NSM_n$ — функция расчета метрик защищенности, связанных с требованиями, основанными на принципе «разумной достаточности»; SSM_n — множество значений стандартных метрик защищенности; NSM_n — множество значений нестандартных метрик защищенности.

Рассмотрим теперь алгоритмы и программные средства, предназначенные для реализации разработанной методики.

3. Алгоритм статического анализа вероятностных графов атак. Среди алгоритмов разработанной методики наибольшую значимость имеют алгоритмы анализа вероятностных графов атак. Поэтому ограничимся их рассмотрением. При этом следует отметить, что имеются алгоритмы статического и динамического анализа этих графов.

Алгоритм статического анализа вероятностных графов атак *PAGAA* (листинг 1) предусматривает, что в зависимости от текущих требований могут выбираться различные метрики защищенности, расчет которых связан с различной степенью охвата графов атак, и различные критерии их разбиения и агрегации.

```
PAGAA (AG, G, SecurityMetricRuleList, PartitioningRuleList, AggregationRuleList,
ProbabilityRange, D, IL)
{
  ASGL.Add (AG);
  Continue = True;
  While (Continue) {
    Continue = False;
    For (i = 0; i < ASGL.Count; i++) {
      curAG = ASGL[i];
      T = GetOperationType (curAG, PartitioningRuleList);
```

```
      If (T == OperationType.Cut
        || T == OperationType.Divide) {
        CutAndDivide (curAG, G, T, PartitioningRuleList, tmpASGL);
        newASGL.Add (tmpASGL);
        Continue = True;
      } else
        newASGL.Add (curAG);
    }
    ASGL = newASGL;
  }
  For (i = 0; i < ASGL.Count; i++) {
    curAG = ASGL[i];
    IMC (curAG, curAG.Goal, IM);
    FloydWarshall (IM, W);
    Aggregation (curAG, IM, W, G, D, AggregationRuleList, AAG);
    metricType = GetMetricType (SecurityMetricRuleList);
    Switch (MetricType) {
      Case MetricType.Type1:
        CalcMetric (curAG, SecurityMetricRuleList, SM);
        Break;
      Case MetricType.Type2:
        CalcMetric (curAG, IM, SecurityMetricRuleList, ProbabilityRange, SM);
        Break;
      Case MetricType.Type3:
        CalcMetric (curAG, W, SecurityMetricRuleList, ProbabilityRange, SM);
        Break;
      Default:
        SM = NULL;
        Break;
    }
    IL.Add (curAG, SM, IM);
  }}
```

Листинг 1. Алгоритм статического анализа вероятностных графов атак.

Исходными данными алгоритма *PAGAA* являются: *AG* — вероятностный граф атак; *G* — целевая вершина; *SecurityMetricRuleList* — список правил расчета метрики защищенности; *PartitioningRuleList* — список правил разбиения на части; *AggregationRuleList* — список правил агрегации; *ProbabilityRange* — диапазон вероятностей; *D* — глубина анализа. Результатом работы алгоритма является список элементов *IL*, каждый из которых пред-

ставляет собой тройку $\langle ASG, SM, IM \rangle$, где ASG — подграф атак; SM — значение метрики защищенности; IM — матрица смежности графа ASG.

Поскольку моделирование атак ориентировано на большие компьютерные сети, расчет метрики защищенности может потребовать проверки значительного количества условий и выполнения значительного количества действий. Множество сочетаний таких условий и действий предлагается представлять в виде правил, структура которых представлена в табл. 1.

Таблица 1. **Типы правил**

Тип правила	Логическая структура
Правило расчета метрики защищенности	Идентификатор правила; Описание; Предусловие применения правила; Действия правила, связанные с расчетом метрики защищенности; Постусловие применения правила, содержащее ограничения для следующего применения данного или других правил расчета метрики защищенности
Правило разбиения графа атак на части	Идентификатор правила; Описание; Тип операции (*Cut* или *Divide*); Предусловие применения правила; Действия правила
Правило агрегации	Идентификатор правила; Описание; Предусловие применения правила; Действия правила

Список правил расчета метрики защищенности передается в алгоритм через список *SecurityMetricRuleList*. Расчет метрики защищенности выполняется с помощью функции *CalcMetric*, которая регулирует сложность вычислений в зависимости от текущих требований.

В алгоритме *PAGAA* предусмотрен расчет трех типов метрик защищенности, различающихся сложностью производимых вычислений.

Для расчета метрики первого типа достаточно выполнить обход всех вершин графа. Примером метрики первого типа является метрика "годность к эксплуатации" E, вычисляемая по следующей формуле:

$$E = \sum_{i=1}^{n} E(h_i),$$

где h_i — хост в сети, n — количество хостов, $E(h_i)$ — функция годности хоста к эксплуатации.

Для расчета метрики второго типа требуется поиск пути графа, соответствующего минимальному или максимальному значению, возвращаемому некоторой заданной функцией оценки. Примером метрики второго типа является метрика, показывающая максимальную вероятность достижения цели атаки из вершин, выбираемых по некоторому заданному условию.

Для расчета метрики третьего типа требуется вычисление и использование матрицы достижимости. Примером метрики третьего типа может служить метрика распространения атаки, представленная в работе [21], дающая оценку ущерба, который может возникнуть в результате того, что нарушитель достигнет своей цели при совершении атаки, начиная с некоторого хоста.

Следует отметить, что для сокращения вычислений путем фокусировки внимания на наиболее существенных фрагментах графа атак и обеспечения удобной визуализации графа в алгоритме *PAGAA* предусмотрены следующие вспомогательные процедуры: разбиения графа на части *CutAndDivide* и агрегации фрагментов графа *Aggregation*.

Разбиение графов на части в процедуре *CutAndDivide* предлагается проводить с помощью двух операций над графами *Cut* и *Divide* [22]. Операция *Cut* предназначена для удаления дуг и вершин в соответствии с *Cut* –правилами. Операция *Divide* предназначена для разделения графов на подграфы в соответствии с *Divide* –правилами, в результате чего одной вершине исходного графа может соответствовать несколько вершин, каждая из которых относится к образованному подграфу и получает разметку, совпадающую с разметкой исходной вершины, а множества связанных с ними дуг не пересекаются. Структура *Cut* –правил и *Divide* –правил приведены выше в табл. 1. На рис. 5 представлены примеры разбиения вероятностных графов атак на части с помощью операций *Cut* и *Divide* для случая дизъюнкции элементарных условий.

В процедуре *Aggregation*, осуществляющей агрегацию фрагментов вероятностных графов атак, также используется набор правил для учета текущих требований (см. табл. 1). Пример операции агрегации фрагментов вероятностных графов атак представлен на рис. 6.

В процедуре *Aggregation* используется функция получения списка узлов, из которых достижим текущий узел. Предполагается, что данная функция либо сама использует алгоритм Флойда–Уоршалла для определения путей в графе, либо использует готовый список путей, полученный в качестве побочного результата при построении

матрицы достижимости с помощью алгоритма Флойда–Уоршалла на тех шагах алгоритма, которые предшествуют процедуре *Aggregation*.

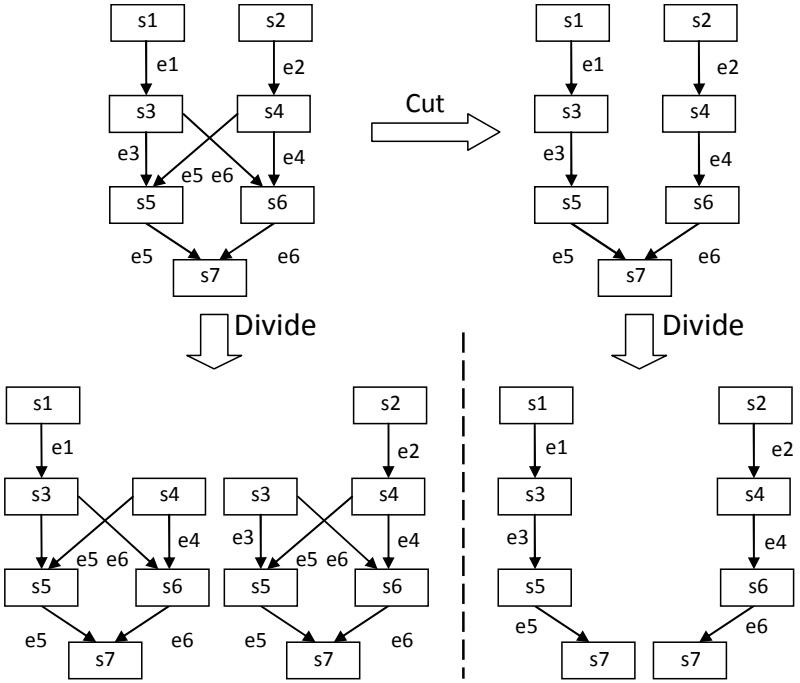

Рис. 5. Примеры разбиения вероятностных графов атак на части с помощью операций *Cut* и *Divide*.

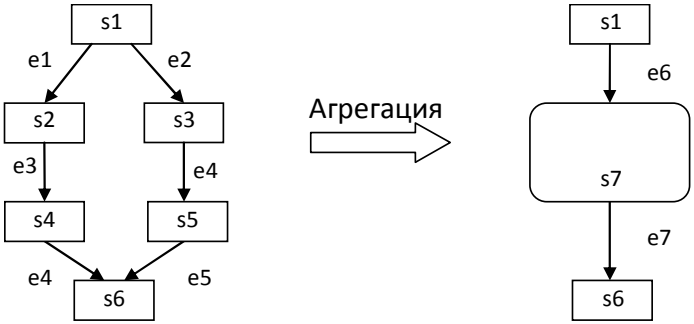

Рис. 6. Пример агрегации фрагментов вероятностных графов атак.

Результат агрегации зависит от списка правил агрегации, структура которых была представлена в табл. 1. Различные варианты агрегации, для которых используются правила по умолчанию, представлены в табл. 2.

Таблица 2. **Варианты агрегации фрагментов графа атак**

Название	Описание правил
Агрегация по хостам	Агрегируются пути, в которых все связи, кроме последней, начинаются и заканчиваются узлами, относящимися к одному и тому же хосту.
Агрегация по маршрутизаторам	Агрегируются пути, в которых все связи, кроме последней, начинаются и заканчиваются узлами, относящимися к одному и тому же маршрутизатору.
Агрегация параллельных путей	Агрегируются пути длины не больше n, начинающихся хостом h_1 и заканчивающихся хостом h_2, $h_1 \neq h_2$, причем внутри агрегированных путей нет узлов, относящихся к h_1 и h_2.
Агрегация разведывательных действий	Агрегируются пути, начинающиеся и заканчивающиеся узлами, относящимися к одному и тому же хосту и связанные только с разведывательными действиями нарушителя (использование утилит *ping*, *nmap* т.п.)
Агрегация действий, нарушающих политику безопасности	Агрегируются пути, начинающиеся и заканчивающиеся узлами, относящимися к одному и тому же хосту и связанные только с действиями нарушителя по изменению состава или свойств объектов политики безопасности (ролей, привилегий, уровней доступа и др.)

Оценки вычислительной сложности алгоритмов, использованных в составе алгоритма статического анализа вероятностных графов атак, представлены в табл. 3.

4. Алгоритм динамического анализа вероятностного графа защищенности. Для динамического анализа защищенности сетей на основе предварительно построенных вероятностных графов атак разработан алгоритм *SelectScenarios* (листинг 2), в котором формируются сценарии действий нарушителей на основе выбора текущего расчета метрики защищенности среди трех типов, описанных выше.

Правила расчета метрики защищенности и правила разбиения графа на части, передаваемые в алгоритм *SelectScenarios*, по своей

структуре аналогичны правилам, используемым в алгоритме *PAGAA* (см. табл. 1).

Таблица 3. **Оценки временной вычислительной сложности алгоритмов**

Алгоритм	**Оценка временной вычислительной сложности**
CutAndDivide	$O(nlr)$, где n — количество узлов, l — количество дуг, r — количество правил разбиения графа на части
Aggregation	$O(n!r)$, где n — размерность матрицы достижимости, r — количество правил агрегации фрагментов графа
*Floyd*Warshall (алгоритма Флойда–Уоршалла)	$O(n^3)$, где n — размерность матрицы смежности
IMC (алгоритм построения матриц смежности)	$O(nm)$, где n — количество узлов, m — количество дуг
CalcMetric для метрики типа 1	$O(nr)$, где n — количество узлов, r — количество правил расчета метрики защищенности
CalcMetric для метрики типа 2	$O(nm)$, где n — количество узлов, m — количество дуг
CalcMetric для метрики типа 3	$O(wr)$, где w — множество путей, связывающих достижимые вершины, r — количество правил для расчета метрики защищенности; или в случае если необходимо сначала определить пути между достижимыми вершинами, то оценка следующая: $O(n^3r)$, где n — размерность матрицы смежности

```
SelectScenarios (AG, G, SecurityMetricRuleList, PartitioningRuleList, ProbabilityRange,
IntruderResourceRanges, SecurityMerticRange, Scenarios)
{
  ASGL.Add (AG);
  Continue = True;
  While (Continue) {
    Continue = False;
    For (i = 0; i < ASGL.Count; i++) {
      curAG = ASGL[i];
      T = GetOperationType (curAG, PartitioningRuleList);
      If (T == OperationType.Cut
       || T == OperationType.Divide) {
        CutAndDivide (curAG, G, T, PartitioningRuleList, tmpASGL);
```

```
        newASGL.Add (tmpASGL);
        Continue = True;
      } else
        newASGL.Add (curAG);
    }
   ASGL= newASGL;
}
For (i = 0; i < ASGL.Count; i++) {
    curAG = ASGL[i];
   IMC (curAG, curAG.Goal, IM);
   metricType = GetMetricType (SecurityMetricRuleList);
   Switch (MetricType) {
       Case MetricType.Type1:
         CalcMetric (curAG, SecurityMetricRuleList, SM);
         Break;
       Case MetricType.Type2:
         CalcMetric (curAG, IM, SecurityMetricRuleList, ProbabilityRange, SM);
         Break;
       Case MetricType.Type3:
         CalcMetric (curAG, W, SecurityMetricRuleList, ProbabilityRange, SM);
         Break;
       Default:
         SM = NULL;
         Break;
   }
   P = CalcProbability (curAG, IM);
   R = CheckResources (curAG, IntruderResourceRanges);
   If (
    (ProbabilityRange == NULL
    || P.Low >= ProbabilityRange.Min
    && P.High <= ProbabilityRange.Max)
    && (IntruderResourceRanges == NULL || R)
    && (SecurityMerticRange == NULL
    || SM >= SecurityMerticRange.Min
    && SM <= SecurityMerticRange.Max)
   )
   Scenarios.Add (curAG); }}
```

Листинг 2. Алгоритм определения релевантных вероятностных графов атак.

Исходными данными алгоритма *SelectScenarios* являются вероятностный граф атак AG, целевая вершина G, правила расчета метрики защищенности *SecurityMetricRuleList*, правила разбиения на части

PartitioningRuleList, диапазон вероятностей *ProbabilityRange*, ограничения на значения переменных *IntruderResourceRanges* и диапазон допустимых значений метрики защищенности *SecurityMetricRange*. Результатом является список подграфов атак *Scenarios* , ассоциированных со сценариями действий нарушителей.

Вершины графа атак в алгоритме *SelectScenarios* ассоциируются с логическими утверждениями, учитывающими ограничения на значения переменных, характеризующих состояния компьютерной сети. Используя такие ассоциации, заранее подготовленную базу данных угроз, а также значения метрики защищенности и вероятности, связанные с дугами графа атак, осуществляется отбор предварительно построенных вероятностных графов атак в соответствии с требованиями текущей итерации процессов моделирования атак.

Алгоритм *SelectScenarios* реализован с помощью функций, обеспечивающих выбор среди нескольких предварительно созданных графов или подграфов атак таких графов (подграфов), которые в наибольшей степени удовлетворяют требованиям пользователя, представленным в виде диапазона вероятностей *ProbabilityRange*, ограничений на ресурсы нарушителей *IntruderResourceRanges* и диапазона допустимых значений метрики защищенности *SecurityMetricRange*. Список выбираемых графов атак или выбираемых сценариев действий нарушителей вместе с привязанными к ним предварительно вычисленными значениями некоторой метрики защищенности передается в качестве значения параметра *IM* , который формируется в процедуре *IMC* .

В процессе работы алгоритма производится вычисление вероятности достижения цели из узлов графа на основе вероятностных оценок, которые определены для каждой связи в графе атак, с помощью функции *CalcProbability*. При этом предполагается, что в случае конъюнктивных связей события переходов между связанными узлами являются независимыми, а в случае дизъюнктивных связей события переходов между связанными узлами являются несовместными.

Полученные в результате выполнения алгоритма *SelectScenarios* наборы сценариев действий нарушителей могут рассматриваться не только по отдельности, но и как несколько параллельно выполняемых сценариев. Это соответствует случаю распределенных атак, например, атак типа DDoS. При этом следует отметить, что качество моделирования распределенных атак существенно зависит от того, каким образом будет осуществляться разбиение графа на части с помощью про-

цедуры *CutAndDivide*, что, в свою очередь, определяется набором правил разбиения, сформированных на основе текущих требований.

5. Программные средства анализа моделей атак для больших компьютерных сетей. Для статического анализа моделей атак в больших компьютерных сетях разработана программа на языке Java, основными функциями которой являются вычисление метрик защищенности, основанных на предварительно построенных вероятностных графах атак, а также их частях или агрегированных фрагментах, а также автоматизированное формирование отчетов, содержащих значения метрик защищенности, информацию по уязвимостям сети и рекомендации по их устранению.

Программа статического анализа моделей атак включает компонент визуализации вероятностных графов атак, редактор правил расчета метрик защищенности, компонент анализа вероятностных графов атак, компонент разбиения вероятностных графов атак на части, компонент агрегации фрагментов вероятностных графов атак, компонент доступа к базе данных и компонент поддержки контекстного справочника.

Компонент визуализации графов атак поддерживает различные представления графа атак, тем самым фокусируя внимание пользователя на наиболее интересных фрагментах графа. В зависимости от текущих настроек, представления графа атак отличаются наличием/отсутствием отображения вероятностных оценок связей и различными вариантами агрегации путей графа, осуществляемой с использованием процедуры *Aggregation*.

Редактор правил расчета метрик защищенности предназначен для редактирования правил, структура которых представлена в табл. 1. Предусмотрено два вида расчета: на основе требований (рекомендуемые значения метрик защищенности) и на основе вероятностных графов атак (фактические значения метрик защищенности). Алгоритмы расчета метрик защищенности описываются в действиях правил на встроенном языке.

Компонент анализа вероятностных графов атак обеспечивает программную реализацию алгоритмов *PAGAA* и *SelectScenarios*. Пользователю предоставляется возможность редактирования правил *SecurityMetricRuleList*, *PartitioningRuleList* и *AggregationRuleList*.

Для представления результатов анализа вероятностных графов атак предназначен *компонент генератора отчетов*, который обеспечивает формирование полного отчета или отдельных разделов отчета в

зависимости от текущих информационных потребностей пользователей. Сформированные отчеты могут быть представлены в текстовом или графическом виде. В процессе генерации отчетов в текстовой форме используются шаблоны. В базовую конфигурацию генератора отчетов включены шаблоны, позволяющие различным образом представить отчеты в виде простого текста или в виде HTML–документов. Шаблоны представляют собой различные комбинации элементов отчета с разделителями, заголовками и HTML–тегами.

Особенностью компонента генератора отчетов является использование предварительно подготовленного набора рекомендаций по повышению уровня защищенности сети, которые сопоставлены с набором известных уязвимостей и сохранены в базе данных комплекса. В процессе формирования отчета для каждой уязвимости осуществляется поиск соответствующих рекомендаций.

Для данных отчета, которые содержат числовые значения, предусмотрено графическое представление в виде графиков, диаграмм–областей, столбчатых и круговых диаграмм. За счет одновременного графического отображения метрик защищенности для нескольких хостов или других элементов сети пользователи получают наглядное представление информации и возможность оперативно сравнивать защищенность различных фрагментов большой компьютерной сети.

Для фокусировки внимания на конкретных фрагментах и особенностях сети предусмотрена фильтрация данных отчетов по типам элементов сети, по свойствам элементов и по значениям свойств, связанных с уязвимостями.

В листинге 3 представлен пример текстового представления отчета о результатах анализа вероятностного графа атак, в котором метрика опасности возникновения атакующего действия обозначена как *Danger* со значениями *High*, *Medium* и *Low*. В соответствии с данным примером "узким местом" является хост "Server", через который проходит 43 трассы. Поэтому пользователю рекомендуется изменить конфигурацию этого хоста, в частности, сменить операционную системы или ее версию.

Уязвимые хосты:

Хост: Firewall
Уязвимости: Banners (Danger:High), OS = WIN_2003 (Danger:High), Services (Danger:High)

Хост: TDM

Уязвимости: Banners (Danger:Low), OS = LINUX (Danger:Low), Services (Danger:Low)

Хост: PDA
Уязвимости: Banners (Danger:Low), OS= LINUX (Danger:Low), Services (Danger:Low)

Хост: PositifConsole
Уязвимости: OS = WIN_XP (Danger:High), Banners (Danger:High), Services (Danger:High), SYNflood (Danger:High)

Хост: AP
Уязвимости: OS = LINUX (Danger:Medium), Banners (Danger:Medium), Services (Danger:Medium), SYNflood (Danger:Medium)

Хост: Laptop
Уязвимости: Banners (Danger:Low), OS = WIN_CE (Danger:Low), Services (Danger:Low)

Хост: Server
Уязвимости: ftp-dictionary (Danger:Medium), OS = WIN_2000 (Danger:Medium), ServU-list-l (Danger:Medium), Services (Danger:Medium), Portfuck (Danger:Medium), ServU-MDTM (Danger:Medium), Banners (Danger:Medium), SYNflood (Danger:Medium), ServU-MKD (Danger:Medium)
Рекомендации:

1. ServU-list-l: Необходимо обновить Serv-U до последней версии
2. Portfuck: Необходимо сменить операционную систему или обновить MS Windows 2000 до последней версии
3. ServU-MDTM: Необходимо обновить Serv-U до последней версии
4. ServU-MKD: Необходимо обновить Serv-U-server до последней версии

Метрики безопасности, определяемые конфигурацией сети:

Количество хостов в анализируемой сети: 7
Количество установленных экземпляров операционных систем на хостах:

1. OS = WIN_CE : 1
2. OS = LINUX : 3
3. OS = WIN_XP : 1
4. OS = WIN_2003 : 1
5. OS = WIN_2000 : 1

Метрики безопасности хостов:

Уровень критичности хоста:

1. Firewall : High
2. TDM : Low
3. PDA : Low
4. PositifConsole : High

5. AP : Medium
6. Laptop : Low
7. Server : Medium

Метрики безопасности действий нарушителя:

Уровень критичности действий и трасс:

1. ping : Low
2. OS : Low
3. Services : Low
4. SYNflood : Low
5. ftp-dictionary : High
6. Services->Banners : Low
7. ServU-list-l : Low
8. ServU-MKD : Low
9. ServU-MDTM : Low
10. Services->Banners->OS : Low
11. Portfuck : Low

Самый критичный хост:

Хост: Server

1. количество путей, проходящих через хост: 43

Итоговые метрики защищенности хостов:

Количество различных хостов в сети: 7
Список хостов сети: PDA | Firewall | AP | PositifConsole | TDM | Server | Laptop

---

Последовательности действий нарушителей:

Общее количество различных типов последовательностей действий: 11
Список различных типов последовательностей действий:
Services->Banners->OS | ServU-list-l | ServU-MKD | ftp-dictionary | OS | Portfuck | SYN-flood | ServU-MDTM | Services->Banners | Services | ping

---

Пути на графе атак:

Количество путей на графе атак: 43
Количество опасных путей: 8
Количество путей, проходящих через хосты

1. PDA: 5
2. Firewall: 5
3. AP: 6
4. PositifConsole: 6

5. TDM ->> 5
6. Server: 43
7. Laptop: 5

Итоговая опасность возникновения атакующих действий:

Danger: High

Листинг 3. Отчет о результатах анализа вероятностных графов атак.

Для анализа динамических характеристик разработаны следующие Java–программы: формирования сценариев действий нарушителей, исполнения сценариев действий нарушителей и анализа результатов моделирования атак. В результате использования этих программ анализ динамических характеристик может быть выполнен на основе истории разведывательных и атакующих действий, выполненных в процессе запуска сценариев действий нарушителей.

В качестве формализмов представления требований в программах анализа моделей атак использованы правила и ограничения в виде диапазонов значений. Использованы несколько типов правил, отличающиеся по своей структуре и выполняемым действиям, что связано с особенностями алгоритмов, которые их используют (см. табл. 1). Также предусмотрены следующие ограничения: диапазоны вероятностей достижения целей нарушителем, глубина анализа графа атак, ограничения на ресурсы нарушителей и диапазоны допустимых значений метрики защищенности.

Программа формирования сценариев действий нарушителей содержит компонент определения релевантных действий нарушителей по вероятностным графам атак, редактор правил расчета метрик защищенности, компонент экспорта сценариев действий нарушителей в MS Visio, компонент визуализации сценариев действий нарушителей, компонент доступа к базе данных и компонент поддержки контекстного справочника.

Компонент определения релевантных сценариев действий нарушителей по вероятностным графам атак обеспечивает программную реализацию алгоритма *SelectScenarios* . Результатом работы компонента является список подграфов атак, удовлетворяющих требованиям пользователя. К связям подграфов привязаны действия нарушителя. Правила разбиения графа атак на части определены таким образом, что не допускают ветвлений, полученных в результате работы компонента. Сценарии действий нарушителей представляют собой *трассы*, т.е.

пути в графе атак. Построение трасс необходимо для определения "узких мест" в графе атак и формирования рекомендаций по их устранению.

Программа исполнения сценариев действий нарушителей включает интерпретатор сценариев действий нарушителей, компоненты интерфейса с внешними программными и аппаратными средствами, компонент доступа к базе данных и компонент поддержки контекстного справочника.

Компонент исполнения сценариев действий нарушителя позволяет последовательно выполнить действия сценариев, сформированных компонентом определения релевантных действий нарушителей по вероятностным графам атак. Предусмотрено как пошаговое выполнение сценариев, требующее подтверждения от пользователя перед каждым шагом, так и выполнение сценариев без подтверждений. Если сценарии представляют собой трассы, то они выполняются без остановки. В случае если сценарии содержат ветвления, пользователю предлагается выбрать одну из альтернатив, содержащуюся в сценарии. Компонент может либо только имитировать исполнение шагов сценариев, либо выполнять реальные действия посредством компонентов интерфейса с внешними программными и аппаратными средствами. Предусмотрена возможность выделения текущих исполняемых шагов сценариев на фоне общей структуры сценариев с помощью компонента визуализации сценариев действий нарушителей.

6. Заключение. Таким образом, в основе разработанной методики итерационного моделирования атак в больших компьютерных сетях лежит представленная в настоящей работе формальная модель, в которой процесс моделирования реализуется в виде следующих друг за другом итераций, на которых последовательно уточняются параметры анализируемой сети и предъявляемые к ней требования.

Разработанные алгоритмы и программные средства, разработанные в рамках рассматриваемой методики, ориентированы на снижение вычислительной сложности процессов моделирования атак, нахождение «узких мест» в сети и выработку рекомендаций по их устранению.

Дальнейшие исследования связываются с расширением типов рассматриваемых метрик защищенности и правил и использованием возможности проведения на их основе логического вывода.

Литература.

1. *Федотов А. М.* Информационная безопасность в корпоративной сети // Проблемы безопасности и чрезвычайных ситуаций. М.: ВИНИТИ, 2008. № 2. С. 88–101.

2. *Сердюк В. А.* Организация и технологии защиты информации: обнаружение и предотвращение информационных атак в автоматизированных системах предприятий. М.: НИУ ВШЭ, 2011.
3. *Zhang S., Song S.* A. Novel Attack Graph Posterior Inference Model Based on Bayesian Network // Journal of Information Security, 2011. № 2. Pp. 8–27.
4. *Котенко Д.И., Котенко И.В., Саенко И.Б.* Методы и средства моделирования атак в больших компьютерных сетях: состояние проблемы // Труды СПИИРАН. Вып. 3(22). СПб.: Наука, 2012.
5. *Котенко И.В., Степашкин М.В., Богданов В.С.* Архитектуры и модели компонентов активного анализа защищенности на основе имитации действий злоумышленников // Проблемы информационной безопасности. Компьютерные системы. 2006, №. 2, С. 7–24.
6. *Котенко И.В., Степашкин М.В., Богданов В.С.* Анализ защищенности компьютерных сетей на различных этапах их жизненного цикла // Изв. вузов. Приборостроение. Т. 49, № 5, 2006, С. 3–8.
7. *Котенко И.В, Степашкин М.В., Дойникова Е.В.* Анализ защищенности автоматизированных систем с учетом социо–инженерных атак // Проблемы информационной безопасности. Компьютерные системы. 2011, № 3, С. 40–57.
8. *Котенко И.В, Степашкин М.В., Котенко Д.И., Дойникова Е.В.* Оценка защищенности информационных систем на основе построения деревьев социо–инженерных атак // Изв. вузов. Приборостроение, Т. 54, № 12, 2011. С. 5–9.
9. *Котенко И.В., Воронцов В.В., Чечулин А.А., Уланов А.В.* Проактивные механизмы защиты от сетевых червей: подход, реализация и результаты экспериментов // Информационные технологии, № 1, 2009. С. 37–42.
10. *Котенко И.В., Коновалов А.М., Шоров А.В.* Моделирование бот–сетей и механизмов защиты от них // Системы высокой доступности, № 2, 2011. С. 107–111.
11. *Котенко И.В., Коновалов А.М., Шоров А.В.* Агентно–ориентированное моделирование бот–сетей и механизмов защиты от них // Вопросы защиты информации, № 3, 2011. С. 24–29.
12. *Gorodetski V., Kotenko I.* Attacks against Computer Network: Formal Grammar–based Framework and Simulation Tool // Lecture Notes in Computer Science, Vol. 2516, 2002.
13. *Kotenko I.* Teamwork of Hackers–Agents: Modeling and Simulation of Coordinated Distributed Attacks on Computer Networks // Lecture Notes in Artificial Intelligence, Springer Verlag. Vol. 2691, 2003. Pp. 464–474.
14. *Козленко А.В., Авраменко В.С., Саенко И.Б., Кий А.В.* Метод оценки уровня защиты информации от НСД в компьютерных сетях на основе графа защищенности // Труды СПИИРАН. Вып. 3(22). СПб.: Наука, 2012.
15. *Котенко И.В., Саенко И.Б., Полубелова О.В., Чечулин А.А.* Применение технологии управления информацией и событиями безопасности для защиты информации в критически важных инфраструктурах // Труды СПИИРАН. Вып. 1(20). СПб.: Наука, 2012.
16. *Котенко И.В., Саенко И.Б., Полубелова О.В., Чечулин А.А.* Технологии управления информацией и событиями безопасности для защиты компьютерных сетей // Проблемы информационной безопасности. Компьютерные системы. № 2, 2012. С. 57–68.
17. ГОСТ Р ИСО/МЭК 15408–1–2002. Информационная технология. Методы и средства обеспечения безопасности. Критерии оценки безопасности информационных технологий. Часть 1. Введение и общая модель. М.: Госстандарт России, 2002. 35 с.
18. ГОСТ Р ИСО/МЭК 17799–2005. Информационная технология. Практические правила управления информационной безопасностью. М.: Стандартинформ, 2006. 55 с.

19. ГОСТ Р ИСО/МЭК 13335–1–2006. Информационная технология. Методы и средства обеспечения безопасности. Часть 1. Концепция и модели менеджмента безопасности информационных и телекоммуникационных технологий. М.: Стандартинформ, 2007. 18 с.
20. *Нестеров С.А.* Анализ и управление рисками в информационных системах на базе операционных систем Microsoft. INTUIT, 2009.
21. *Ahmed M., Al-Shaer E., Khan L.* A novel quantitative approach for measuring network security. Proceedings of IEEE INFOCOM 2008, April 2008. Pp. 1957–1965.
22. *Lee J., Lee H., In H.P.* Scalable attack graph for risk assessment. Proceedings of the 23rd International Conference on Information Networking (ICOIN'09), 2009. Pp. 78–82.

Котенко Дмитрий Игоревич — аспирант кафедры МО ЭВМ Санкт-Петербургского государственного электротехнического университета «ЛЭТИ». Область научных интересов: информационная безопасность в компьютерных сетях. Число научных публикаций — 6. dmitrykotenko1986@gmail.com; СПбГЭТУ«ЛЭТИ», ул. проф. Попова, д. 5, Санкт-Петербург, 197376, РФ; р.т. +7(812)234–9668. Научный руководитель — Молдовян А.А.

Kotenko Dmitry Igorevich — postgraduate student of Computer Software Department of Saint Petersburg State Electrotechnical University «LETI» (ETU). Research interests: network information security. The number of publications — 6. dmitrykotenko1986@gmail.com;Saint Petersburg State Electrotechnical University «LETI» (ETU), 5, Professor Popov st., Saint-Petersburg, 197376, Russia; office phone +7(812)234–9668. Scientific adviser — Moldovjan A.A.

Котенко Игорь Витальевич — д-р техн. наук, профессор; заведующий лабораторией проблем компьютерной безопасности СПИИРАН. Область научных интересов: безопасность компьютерных сетей, в том числе управление политиками безопасности, разграничение доступа, аутентификация, анализ защищенности, обнаружение компьютерных атак, межсетевые экраны, защита от вирусов и сетевых червей, анализ и верификация протоколов безопасности и систем защиты информации, защита программного обеспечения от взлома и управление цифровыми правами, технологии моделирования и визуализации для противодействия кибер–терроризму. Число научных публикаций — более 450. ivkote@comsec.spb.ru, www.comsec.spb.ru; СПИИРАН, 14-я линия В.О., д.39, Санкт-Петербург, 199178, РФ; р.т. +7(812)328–2642, факс +7(812)328–4450.

Kotenko Igor Vitalievich — Dr. Sc. in Technical Sciences, professor; head of Laboratory of Computer Security Problems, SPIIRAS. Research interests: computer network security, including security policy management, access control, authentication, network security analysis, intrusion detection, firewalls, deception systems, malware protection, verification of security systems, digital right management, modeling, simulation and visualization technologies for counteraction to cyber terrorism; The number of publications — more 450. ivkote@comsec.spb.ru, www.comsec.spb.ru; SPIIRAS, 39, 14-th Line V.O., St. Petersburg, 199178, Russia; office phone +7(812) 328–2642, fax +7(812)328–4450.

Саенко Игорь Борисович — д-р техн. наук, профессор; ведущий научный сотрудник лаборатории проблем компьютерной безопасности СПИИРАН. Область научных интересов: автоматизированные информационные системы, информационная безопасность. Число научных публикаций — более 250. ibsaen@mail.ru; СПИИРАН, 14-я линия В.О., 39, Санкт-Петербург, 199178, РФ; р.т. +7(812)328–2642, факс +7(812)328–4450.

Saenko Igor Borisovich — Dr. Sc. in Technical Sciences, professor; leading researcher, Laboratory of Computer Security Problems, SPIIRAS. Research interests: automated information systems, information security. The number of publications — more 250. ibsaen@mail.ru; SPIIRAS, 14-th line, 39, St. Petersburg, 199178, Russia; office phone +7(812)328–2642, fax +7(812)328–4450.

Поддержка исследований. В публикации представлены результаты исследований, поддержанные Министерством образования и науки Российской Федерации (государственный контракт 11.519.11.4008), грантами РФФИ (проекты 10–01–00826–а, 11–07–00435–а, 12-07-13119-офи_м_РЖД), программой фундаментальных исследований ОНИТ РАН (проект 2.2) и проектами Седьмой рамочной программы Европейского Союза SecFutur и MASSIF.

Рекомендовано лабораторией криптологии, заведующий лабораторией Молдовян Н.А., д–р техн. наук, профессор, заслуженный изобретатель РФ.
Статья поступила в редакцию 06.08.2012.

РЕФЕРАТ

Котенко Д.И., Котенко И.В., Саенко И.Б. **Методика итерационного моделирования атак в больших компьютерных сетях.**

В статье рассматриваются основные компоненты методики итерационного моделирования атак в больших компьютерных сетях, которыми являются формальная модель, алгоритмы анализа вероятностных графов атак и программные средства их реализации. Формальная модель итерационного моделирования атак включает модели процессов определения задач моделирования, построения моделей атак, запуска моделей и анализа результатов моделирования атак. Алгоритмы анализа вероятностных графов атак обеспечивают расчет метрик защищенности и нахождение подграфов атак, ассоциированных со сценариями действий нарушителей. Программные средства анализа моделей атак для больших компьютерных сетей обеспечивают их статический анализ и анализ динамических характеристик.

Формальная модель итерационного процесса моделирования атак для больших компьютерных сетей представлена четверкой, включающей множество процессов определения задач моделирования, множество процессов построения моделей атак, множество процессов запуска моделей и множество процессов анализа результатов моделирования атак. Все эти множества разделяются на подмножества, соответствующие итерациям моделирования.

Множество процессов определения задач на каждой итерации описывают подготовку требований, получение и структурирование исходных данных о сети, а также приобретение знаний. Множество процессов построения моделей атак описывает формализмы для внутреннего и внешнего представления модели и ее внутреннюю структуру. Множество процессов запуска моделей определяется с помощью функций, описывающих формирование начального состояния модели, а также переход модели в другие состояния, включая завершающее состояние. Множество процессов анализа результатов включает множества процессов расчета метрик защищенности, анализа рисков на основе принципа «разумной достаточности», построения моделей нарушителей и визуального представления результатов.

Алгоритмы реализации методики представлены алгоритмами статического и динамического анализа вероятностных графов атак, которые ориентированы на расчет трех типов метрик защищенности для больших компьютерных сетей. Для сокращения вычислений предусмотрены вспомогательные процедуры разбиения графа на части и агрегации фрагментов графа.

Рассмотренные программные средства ориентированы на нахождение «узких мест» в сети и выработку рекомендаций по их устранению. Приведен пример текстового представления отчета о результатах анализа вероятностного графа атак, в котором вычисляется метрика опасности возникновения атакующего действия, указаны трассы, проходящие через анализируемый хост, и выдаются рекомендации по изменению конфигурации хоста.

SUMMARY

Kotenko D.I., Kotenko I.V., Saenko I.B. **Methodology of iterative attack modelling in large computer networks.**

The paper describes the basic components of the methodology of iterative attack modelling in large computer networks, which constitutes formal model, analysis algorithms for probabilistic attack graphs and software. Formal model of iterative attack modelling process involves the process models of task definition modelling, attack model building, model execution and model result analysis. Probabilistic attack graph analysis algorithms provide calculation of security metrics and finding attack sub-graphs associated with intruder action scripts. Software tools for attack model analysis in large computer networks provide their static analysis and analysis of dynamic characteristics.

Formal model of iterative attack modeling process on large networks is presented by the quartet consisting of a set of processes of modeling task definitions, a set of processes of model development, a set of model run processes, and a set of processes of simulation results analysis. All these sets are divided into subsets, corresponding to iterations of modeling.

The set of processes of task definition at each iteration describes training requirements, getting and structuring input data about the network, as well as the acquisition of knowledge. The set of processes of model development describes the formalisms for internal and external representation of the model and its internal structure. The set of model run processes is defined with the use of the functions describing the formation of the initial state of the model, as well as transition model in other states, including the final status. The set of processes of simulation results analysis includes numerous processes for security metrics, risk analysis on the basis of the principle of reasonable sufficiency, offender model development and visual representation of the results.

Algorithms of methodology implementation are algorithms for static and dynamic analysis of probabilistic attack graphs that focused on three types of security metrics calculation for large computer networks. To reduce computation the support procedures of partitioning of the graph and aggregation graph fragments are provided.

Reviewed software is targeted at finding bottlenecks in the network, and making recommendations to address them. An example of a text report about the results of the analysis of probabilistic attack graph is presented in which the attacker's action risk metric is evaluated, the routes, passing through the target host, are specified and recommendations to modify the configuration of a host are issued.