УДК 004.056

А.А. Шкадов

МОДЕЛЬ УПРАВЛЕНИЯ ЗАЩИЩЕННОСТЬЮ КОМПЬЮТЕРНОЙ СЕТИ НА ОСНОВЕ ИЕРАРХИЧЕСКИХ РАСКРАШЕННЫХ СЕТЕЙ ПЕТРИ

Предложена модель управления защищенностью компьютерной сети на основе иерархических раскрашенных сетей Петри. Проведен анализ существующих подходов к управлению защищенностью компьютерной сети. Рассмотрены особенности применения модели на примере реализации угроз, направленных на Web – приложения.

Ключевые слова: раскрашенные сети Петри, политика безопасности, уязвимости, управление защищенностью

Анализ существующих подходов управления защищенностью компьютерной сети

Анализ существующих решений [1,2] показывает, что на сегодняшний день можно выделить два основных подхода к управлению защитой ресурсов компьютерной сети в условиях локальных или сетевых атак:

безопасность на основе распространяемых политик;

предотвращение вторжений путем нейтрализации уязвимостей.

В рамках первого подхода пользователь получает набор политик (ограничений), которые позволяют защититься от новой угрозы до появления сигнатур, однозначно идентифицирующих соответствующую атаку, или обновлений, исключающих последнюю.

Отличительной особенностью второго подхода является предупреждение локальных и удаленных атак путем анализа журналов регистрации событий на территориально-распределенных узлах компьютерной сети и реагирования на них в реальном времени. Управление защитой реализуется путем активизации соответствующего признакам атаки набора правил. В рамках защиты сетевого периметра от внешних злоумышленников данный подход предусматривает использование технологии предотвращения вторжений, которая позволяет избежать попадания на компьютер вирусов и червей, а также защититься от хакерских атак. Достигается это посредством своевременной блокировки отдельных портов (например, тех, через которые на компьютер попадает опасный на данный момент червь), запрета доступа к определенным файлам и папкам.

Вышеперечисленные подходы имеют ряд существенных недостатков. При первом защищенность сети ставится в зависимость от качества услуг внешнего сервиса. Также возникает проблема смены политик вследствие их большого количества. Недостатком второго подхода является отсутствие возможности обеспечения баланса между защищенностью и функциональностью компьютерной сети в условиях воздействий злоумышленника.

На основе вышеизложенного можно сделать вывод об актуальности создания модели, согласно которой в зависимости от возникающих уязвимостей и воздействующих на них угроз путем динамической смены политик безопасности достигался бы необходимый уровень защищенности.

В данном случае под угрозой будем понимать событие, реализация которого способна нанести ущерб защищаемому объекту путем воздействий на его компоненты. Под политикой безопасности – множество установленных на некотором интервале функционирования компьютерной сети параметров безопасности системы защиты.

В основу модели положен следующий подход: «защищаемый объект», с одной стороны, и считающаяся потенциально враждебной «среда» – с другой. Последняя представлена субъектом, который, используя уязвимости различных компонентов системы, реализует определенные угрозы безопасности информации и НСД.

В качестве критерия r_z будем рассматривать сформулированную в явном виде и в содержательных терминах цель, которая ставится для защищаемого объекта с точки зрения повышения его уровня безопасности относительно исходного. В самом общем виде

целью является перевод системы из исходного состояния меньшей защищенности, $z_0 \in Z$, в другое, желаемое, состояние большей защищенности z^*. Цель реализуется СЗИ с использованием политик безопасности [3]. Свою задачу СЗИ решают путем осуществления воздействий на элементы множества уязвимостей. Характер воздействий в содержательном смысле может быть самый разнообразный, но их результат для данной модели сводится к уменьшению количества успешно реализованных угроз. Рассмотрим особенности применения данного подхода для управления защищенностью ресурсов сети, используя метод имитационного моделирования.

Имитационная модель управления защищенностью ресурсов компьютерной сети на основе иерархических раскрашенных сетей Петри

Для формального представления имитационной модели будем использовать аппарат раскрашенных сетей Петри.

Сеть Петри [4] – это двудольный ориентированный граф с динамическими элементами – фишками. Динамика сети представляет собой процесс перемещения фишек в результате срабатывания переходов. Переход разрешен, если все его входные позиции имеют фишки. При срабатывании переход изымает фишки из своих входных позиций и помещает в свои выходные позиции. Срабатывание перехода происходит мгновенно. Классическая сеть Петри определяется следующим набором:

$$< (P,T,Z); \gamma; \mu_0; \delta_1; \delta_2 >,$$

где: 1) (P,T,Z) - ориентированный двудольный граф с множеством вершин $(P \cup T)$ и множеством дуг $Z = \{z / z \in (P \times X) \cup (T \times P)\}$, при этом: подмножество вершин $P = \{p_1, p_2, ... p_{m(p)}\}$ называется позициями сети Петри, а подмножество вершин $T = \{t_1, t_2, ... t_{m(t)}\}$ называется переходами сети Петри;

2) $\gamma : Z \Rightarrow N^+$ - функция кратности дуг $z \in Z$, где $\gamma(z)$ - целое неотрицательное число;

3) $\mu_0 : P \Rightarrow N$ - функция начальной маркировки, которая ставит в соответствие каждой позиции $p \in P$ неотрицательное целое число $\mu_0(p)$, интерпретируемое как количество маркеров в позиции p;

4) $\delta_1(\mu_i, t)$ - условия возбуждения перехода $t \in T$ в состоянии маркировки μ_i сети Петри, которые имеют вид:

$$\delta_1(\mu_i, t) = \forall p \in P / (p,t) \in Z : \mu_i(p) \geq \gamma(p,t),$$

т.е. во всех входных позициях p для перехода t количество маркеров $\mu_i(p)$ при текущей маркировке должно быть не меньше кратности соответствующей дуги (p,t); если условия возбуждения выполнены, то $\delta_1(\mu_i, t) = 1$ и переход t считается возбужденным, в противном случае $\delta_1(\mu_i, t) = 0$;

5) $\delta_2(\mu_i, t = \mu_{i+1})$ – правило изменения маркировки в результате срабатывания возбужденного перехода t $(\delta_1(\mu_i, t) = 1$, которое для классической сети Петри определяется следующими соотношениями:

$$\forall p \in P : \mu_{i+1}(p) = \mu_i(p) + \tilde{\gamma}(t,p) - \tilde{\gamma}(p,t),$$

(здесь и далее функция со значком ~ тождественно равна функции без этого знака на всей области ее определения и равна 0 вне нее), т.е. из каждой входной позиции p перехода t удаляется количество маркеров, равное кратности соответствующей дуги $\gamma(p,t)$, а в каждую входную позицию p перехода t добавляется соответствующее количество маркеров $\gamma(p,t)$.

Поскольку возможностей классических сетей Петри при спецификации сложных систем оказывается недостаточно, в аппарате сетей Петри имеются их расширения в виде раскрашенных, временных, иерархических и временных сетей Петри. В данной работе использован класс раскрашенных иерархических сетей Петри. Для этого класса сетей Петри в приведенное выше классическое определение дополнительно вводятся следующие функции:
функция цвета $C : P \to \Sigma$, где Σ является конечным множеством непустых типов;
G - функция охраны, которая действует из T в выражения такие, что

$$\forall t \in T : [Type(G(t)) = Z \hat{} Type(Var(G(t))) \subseteq \Sigma],$$

вводится функция временных интервалов

$$Time : T \to Interv(N_+),$$

где $Interv(N_+) = \{[t_1, t_2], [t_3, \infty] \mid t_1, t_2, t_3 \in N_+, t_1 \leq t_2\}$ и границы временного интервала трактуются как раннее и позднее время срабатывания перехода раскрашенной сети Петри.

С точки зрения наглядности восприятия сети Петри и качественного анализа моделируемых процессов, что особенно важно при разработке спецификаций как инструмента

проектирования, особое место занимает графическое представление Сетей Петри. Такой подход (визуального представления) является основой для графических редакторов в специализированных инструментальных программных системах моделирования сетей Петри. В комментариях к графическому представлению и в самом графическом представлении будем придерживаться системы обозначений, используемых в монографии K. Jensen [5], и инструментальной программной системы моделирования раскрашенных сетей Петри Design/CPN [6].

Рассмотрим особенности построения имитационной модели (рис. 1) на примере реализации угроз, направленных на Web–приложения. Как показывает статистика [7], уязвимости в Web-приложениях – одни из наиболее распространенных недостатков защиты сетевой безопасности, на которые приходится приблизительно 70 % всех сетевых атак.

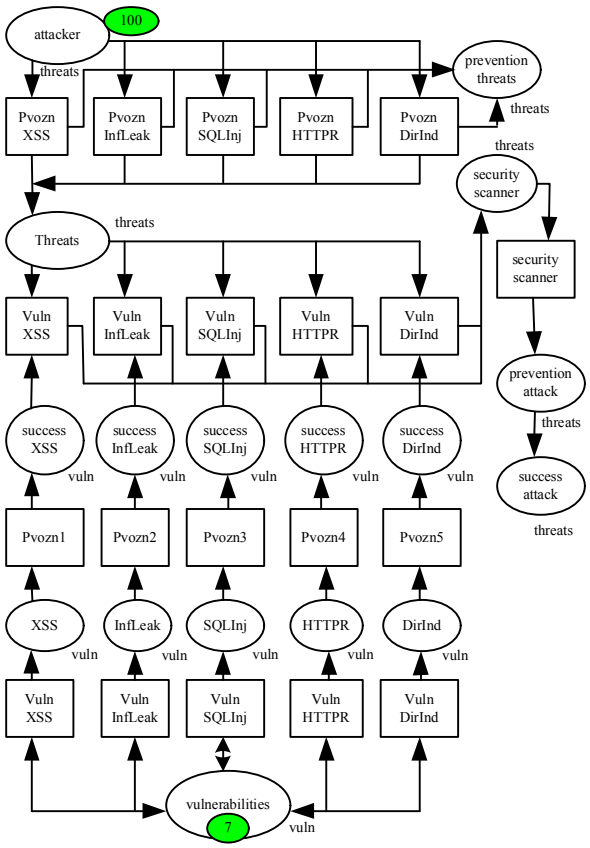

Рис. 1. Имитационная модель управления защищенностью

В представленной модели отражены наиболее часто встречающиеся уязвимости Web-приложений, распределение которых по типам и вероятности появления представлены в табл. 1.

Таблица 1

Тип уязвимости	P_{vozn}
Cross-Site Scripting	0.47
Information Leakage	0.23
SQL injection	0.19
HTTP Response Splitting	0.08
Directory Indexing	0.03

Вероятности появления уязвимостей моделируются с помощью переходов Pvozn1, Pvozn2, Pvozn3, Pvozn4, Pvozn5, при этом числовые значения задаются с помощью функции Bernoulli. Вероятности возникновения угроз, использующих перечисленные типы уязвимостей, представлены в табл. 2.

Моделирование СЗИ ограничено одним из наиболее распространенных классов – сканером защищенности и представлено в модели переходом security scanner.

Таблица 2

Тип уязвимости	P_{vozn}
Cross-Site Scripting	0.46
Information Leakage	0.23
SQL injection	0.15
HTTP Response Splitting	0.11
Directory Indexing	0.05

Вероятность обнаружения угрозы современными сканерами составляет 0,7 [8].

Функция цвета *C* определена для фишек сети на множество цветов двух типов: vuln и threats, уязвимости и угрозы соответственно. Фишки цвета vuln включают в себя два параметра: typevul – тип уязвимости и port – порт, используемый, для активизации данной уязвимости. Фишки цвета threats включают в себя три параметра: class – класс угрозы безопасности (угрозы нарушения целостности, конфиденциальности, доступности), usherb – вероятный ущерб от реализации данной угрозы (высокий, средний, низкий), typ – тип уязвимости используемой для реализации данной угрозы.

Множество используемых для защиты от моделируемых угроз политик безопасности включает следующие.

1. Политики нейтрализации уязвимости SQL injection.
2. Политики нейтрализации уязвимости Cross-Site Scripting.
3. Политики нейтрализации уязвимости Directory Indexing.

4. Политики нейтрализации уязвимости Information Leakage.

5. Политики нейтрализации уязвимости HTTP Response Splitting.

Политики безопасности для каждого класса представляют собой правила по предотвращению возможности использования вышеперечисленных типов уязвимостей. Перечислим основные политики, входящие в каждый из классов.

1.1. Обрабатывать данные перед вставкой в SQL запросы функциями mysql_real escape_string/ Pg_escape_string и обязательно заключать значения в SQL запросе в кавычки.

1.2. Применять специальные конструкторы SQL запросов.

1.3. Использовать по умолчанию механизм обработки ошибок.

1.4. Заблокировать интерфейс ODBC.

1.5. Заблокировать конфигурационные параметры сервера баз данных.

2.1. Фильтрация пришедших извне и публикуемых на сайте данных.

2.2. Запретить запуск Flash приложений.

2.3. Включить XSS фильтр.

3.1. Не указывать при конфигурировании сервера на ошибки в конфигурации.

3.2. Включить фильтрацию запросов.

3.3. Включить очистку КЭШа в базе данных поисковой машины.

4.1. Фильтрация исходящих данных.

4.2. Удаление временных файлов.

5.1. Запретить использование символов перевода строки.

Политики безопасности меняются в зависимости от реализуемой нарушителем в данный момент времени угрозы. Для предложенной модели адекватное угрозе применение политики безопасности приводит к изменению вероятности успешной реализации угрозы, использующей соответствующую уязвимость. Основные результаты моделирования представлены в табл. 3.

Таблица 3

Политики	$t_{\text{îïä}}$	$N_{\text{íáù}}$	$N_{\text{ñçè}}$	$N_{\text{óñï}}$	$P_{\text{ð}}$
Отсутствуют	60 мин	100	83	17	0.17
Присутствуют	60 мин	100	96	4	0.04

В табл. 3 представлены экспериментальные значения следующих показателей, отражающих уровень защищенности ресурсов сети:

$t_{\text{îïä}}$ – время моделирования;

$N_{\text{íáù}}$ – общее количество проведенных атак;

$N_{\text{ñçè}}$ – суммарное количество атак отраженных СЗИ (включая сканер защищенности);

$N_{\text{óñï}}$ – количество успешных атак;

$P_{\text{ð}}$ – вероятность успешной реализации угрозы.

Распределение успешно реализованных атак по типам ущерба представлено в табл. 4.

Таблица 4

Наличие политик	Большой	Средний	Низкий
Отсутствуют	6	8	3
Присутствуют	1	2	1

Таким образом, в работе предложен подход к управлению защищенностью компьютерной сети на основе политик безопасности. Разработана имитационная модель, позволяющая проводить оценку защищенности ресурсов Web-приложений в условиях возникающих угроз при использовании динамически изменяемых политик безопасности. Направлением дальнейших исследований является адаптация механизма шаблонов безопасности операционных систем семейства Windows для реализации изложенного в статье подхода к управлению защищенностью.

Библиографический список

1. *Доля А.* Проактивные технологии для борьбы с вирусами // Экспресс Электроника, 2006.

2. *Котенко И. В., Юсупов Р. М.* Перспективные направления исследований в области компьютерной безопасности // Защита информации. Инсайд. 2006. № 2. - С. 46-57.

3. *Шишкин В. М.* Метамодель анализа, оценки и управления безопасностью // УРСС. М. 2002. С. 92-105.

4. *Котов В. Е.* Сети Петри.– М.: Наука, 1984. - 160 с.

5. *Jensen K.* Colored Petri Nets – Basic Concepts, Analisis Methods and Practical Use.-Vol 1-3, Springer-Verlag, 1997.

6. *Albert K., Jensen K.,* Design/CPN: A Tool Pacckage Supporting the use of Colored Nets// Petri Net Newsletter, April, 1989.pp.22-35

7. Positive Technologies. Статистика уязвимости Web-приложений за 2007 год.

8. *Ермаков А.*. Использование сетевого сканера для повышения защищенности корпоративной информационно-вычислительной сети, М. 2001.