Обнаружение вторжений в компьютерную систему с помощью марковских цепей переменного порядка

Соколов Артем Михайлович

Институт кибернетики им. В.М. Глушкова НАН Украины,
Национальный Технический Университет Украины "КПИ";
адрес: 02091, ул. Тростянецкая 8, кв. 21, Киев, Украина;
sokolov@ukr.net

Аннотация

В работе предлагается подход к построению систем обнаружения аномалий на основе адаптивного варианта модели марковских цепей переменного порядка. Исходная модель была модифицирована для учета динамического характера поведения пользователей. Также предложено несколько методов обнаружения аномалий при помощи новой модели и рассмотрено новый метод борьбы и анализа replay-атак.

Анотація

В роботі пропонується підхід до побудови систем виявлення аномалій на основі адаптивного варіанта моделі марківських ланцюгів із змінним розміром пам’яті. Базова модель була модифікована для врахування динамічного характера поведінки користувачів. Також запропоновано кілька методів виявлення аномалій за допомогою нової моделі і розглянуто новий метод боротьби і аналізу replay-атак.

Abstract

We propose a new approach to the building of anomaly detection systems using an adaptive version of Markov chains with variable memory length. The basic model was modified to take into account the dynamic behaviour of users. Several anomaly detection methods based on the model and a new detection and analysis method of replay attacks were also proposed.

ВВЕДЕНИЕ

За последнее время мир убедился, что даже самые надежные системы безопасности, на построение которых было потрачено много времени и средств, не способны защитить компьютерные системы коммерческих, государственных и военных учреждений от хакерских атак. В большинстве систем применяются обычные механизмы: идентификация и аутентификация, механизмы ограничения доступа к информации согласно правам субъекта и криптографические механизмы. Этот традиционный подход имеет свои недостатки: незащищенность от собственных злонамеренных пользователей, размытость критерия разделения субъектов на "своих" и "чужих" из-за глобализации информационных ресурсов мира, легкость подбора паролей вследствие использования их смысловых разновидностей, применение механизмов ограничения доступа к ресурсам организации, что может снизить производительность и затруднить информационные коммуникации, например в электронном бизнесе.

Из-за недостатков традиционных систем защиты назрела необходимость в механизмах, которые, дополняя традиционные, делали бы возможным обнаружение попыток несанкционированного доступа и своевременно информировали об этом

ответственных за безопасность или активно реагировали в ответ. Желательно, чтобы такие системы могли обнаруживать атаки, даже если хакер уже прошел стандартные защитные системы и с формальной точки зрения соблюдения прав доступа он имеет необходимые полномочия на свои действия в системе.

Для этого предназначены системы обнаружения вторжений (*intrusion detection systems*, или *IDS*). Их можно разделить на системы, которые реагируют на уже известные атаки — системы обнаружения злоупотреблений и системы обнаружения аномалий, которые регистрируют отклонения процессов в системе от нормального хода.

Работа систем обнаружения злоупотреблений базируется на составлении шаблонов, или "подписей", известных атак. Составление шаблонов невысокого уровня абстракции относительно просто, поскольку существует достаточное количество доступных классифицированных баз процессов протекания известных атак. После составления шаблонов события в системе анализируются и проверяются на схожесть с имеющимися в базе. Защитные системы этого типа эффективны на известных атаках, однако в случаях неизвестной атаки или отклонений ее хода от шаблона возникают проблемы. Поэтому приходится поддерживать и непрерывно пополнять большие базы шаблонов на каждую атаку и ее вариации.

Системы обнаружения аномалий более гибки относительно новых атак. Основным их предположением есть то, что все действия злоумышленника обязательно чем-то отличаются от поведения обычного пользователя, т. е. являются аномалиями. Работе таких систем предшествует период накопления информации, когда строится концепция нормальной активности системы, процесса или пользователя. Она становится эталоном, относительно которого оцениваются последующие данные. После составления такой модели система обнаружения аномалий, фиксируя необходимые параметры системы, проверяет их на предмет соответствия модели.

Пионерской работой в отрасли обнаружения аномалий была работа [3], в которой описывались основные понятия и подходы к этой проблеме. Позднее в литературе появилось множество примеров построения таких систем. Практически все можно разделить на такие категории: базирующиеся на перечислении и хранении примеров поведения [4], частотные [3], нейросетевые [1] и строящие конечные автоматы [6,9].

Преимущественное большинство данных, доступное системам аудита, имеет последовательную и недетерминированную природу. Кроме того, для многих источников, которые генерируют эти последовательности, характерно то, что вероятность следующего символа или сигнала зависит от предыдущих элементов, часто — только от небольшого их количества. Это наталкивает на мысль моделировать такие последовательные данные марковскими цепями [2,3,10] или скрытыми марковскими моделями (HMM) [6]. Хотя эти статистические модели способны описывать широкий класс последовательностей и существуют эффективные процедуры их настройки, они имеют серьезные недостатки. Размер марковских цепей экспоненциально увеличивается с ростом их порядка. Из-за этого практическую ценность имеют только модели с небольшим порядком, которые плохо аппроксимировать интересующие нас последовательности. Что касается HMM, то кроме теоретически доказанной сложности их обучения, попытки их реализации показали необходимость весьма больших затрат времени на их настройку.

Как свидетельствует опыт, следующий символ или сигнал редко определяется фиксированным количеством предыдущих элементов; чаще — контекстом переменной

длины, и нет потребности учитывать все возможные контексты — достаточно только те, от которых зависит следующая команда. Поэтому была выбрана модель, которая позволяет получать марковские цепи переменного порядка по сериям примеров работы определенного класса автоматов [7].

Большинство подходов к моделированию поведения пользователей, программ или систем являются неадаптивными. Иногда предлагается периодически перенастраивать модели с целью приспособления к меняющемуся поведению, что приводит у затратам времени и уменьшению надежности моделей с течением времени, которое проходит после очередной перетренировки. Для поддержания реальной адаптивности (когда для получения скорректированной модели модифицируется текущая с учетом поступившей информации) мы применили определенную процедуру изменения параметров нашей модели, в результате чего полученная недавно информация является наиболее весомой. При этом сохраняются полезные аппроксимирующие свойства исходной модели. Единственной известной нам работой, в которой применяется похожий на наш метод адаптивного изменения вероятностей, есть [2]. Однако там за основу взято простейшую марковскую модель первого порядка, а потому не учитывается контекст длины большей, чем одна команда.

ИСХОДНАЯ МАТЕМАТИЧЕСКАЯ МОДЕЛЬ

Рассматривается конечный алфавит символов (в нашем случае пользовательских команд) Σ , конечное множество состояний автомата Q, где каждому состоянию $q \in Q$ соответствует строка-метка $s \in \Sigma^*$. Пустую строку пометим как $\boldsymbol{e}$.

Определение 1 Суффиксным вероятностным автоматом PSA (Probabilistic Suffix Automaton) *называется пятерка* $(Q, \Sigma, \tau, \gamma, \pi)$ *где* $\tau : Q \times \Sigma \to Q$ *— функция переходов между состояниями,* $\gamma : Q \times \Sigma \to [0,1]$ *— вероятность следующего символа,* $\pi : Q \to [0,1]$ *— начальное распределение состояний. Для пары состояний* $q^1, q^2 \in Q$ *, и для каждого* $\sigma \in \Sigma$ *, если* $\tau(q^1, \sigma) = q^2$ *и* q^1 *имеет метку* s^1*, то* q^2 *имеет метку* s^2*, которая есть суффиксом* $s^1\sigma$ *. Для каждого* $q \in Q$ *должно быть* $\sum_{\sigma \in \Sigma} \gamma(q, \sigma) = 1$ *и* $\sum_{q \in Q} \pi(q) = 1$.

Для решения задачи построения автомата по примерам работы некоторого PSA *M*, как класс гипотез выбирается подкласс вероятностных машин — вероятностных суффиксных деревьев PST (*Prediction Suffix Trees*).

Определение 2 PST *T, над алфавитом* Σ *, называется дерево степени* $|\Sigma|$*, где каждое ребро дерева отмечено символом* $\sigma \in \Sigma$ *. С каждым узлом связана пара* (s, γ_s)*, где s — строка, ассоциированная со спуском по дереву, начиная от корня дерева в данный узел в обратном порядке символов в строке, которая помечает этот узел (см. рис. 2.1), а* $\gamma_s : \Sigma \to [0,1]$ *— связанная с s функция вероятностей следующего символа. Требуется, чтобы* $\sum_{\sigma \in \Sigma} \gamma_s(\sigma) = 1$ *для строки s, которая помечает узел.*

Вероятность того, что PST сгенерирует последовательность символов $r = r_1 r_2 \ldots r_N$:

$$P_T^N(r) = \prod_{i=1}^{N} \gamma_{s^{i-1}}(r_i), \qquad (1)$$

где $s^0 = e$ и для каждого $1 < j < (N-1)$, s^j — строка, помечающая самый глубокий узел, в который мы попадаем, сделав проход по дереву, отвечающий $r_j r_{j-1} \cdots r_1$, начиная от корня дерева. Например, для дерева на рис. 1 вероятность строки 00101 будет $0.5 \cdot 0.5 \cdot 0.25 \cdot 0.5 \cdot 0.6$, и метки узлов, в которых были последовательно сгенерированы символы — $s^0 = e, s^1 = 0, s^2 = 00, s^3 = 1, s^4 = 010$.

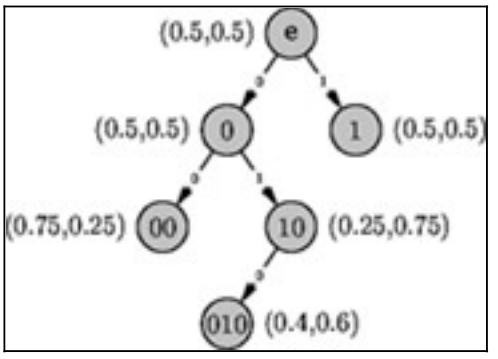

Рис. 1. PST над алфавитом $\Sigma = \{0,1\}$.

Для оценки построенной гипотезы-дерева $\hat{T}$ вводится понятие качества гипотезы.

Определение 3 *PST T называется ε-хорошей гипотезой относительно PSA M, если для любого* $N \in N$,

$$\frac{1}{N} D_{KL}[P_M^N \| P_T^N] = \frac{1}{N} \sum_{r \in \Sigma^N} P_M^N(r) \log \frac{P_M^N(r)}{P_T^N(r)} \le \varepsilon ,$$

где D_{KL} *— это расстояние Kullback-Leibler.*

Требуется, чтобы алгоритм выдавал PST-гипотезу $\hat{T}$, которая с вероятностью как минимум $1-\delta$ была бы ε-хорошей гипотезой относительно PSA *M*.

Используя примеры работы целевого автомата *M*, определим эмпирическую вероятность $\tilde{P}(\cdot)$. Определим $\chi_j(s)$ как 1, когда $r_{j-|s|+1} \cdots r_j = s$ и 0 в другом случае. Тогда, если есть m' строк, каждая длиной $l \ge L+1$, где L — максимальная длина *s*, то положим:

$$\tilde{P}(s) = \frac{1}{m'(l-L+1)} \sum_{i=1}^{m'} \sum_{j=L}^{m-1} \chi_j(s), \qquad \tilde{P}(s) = \frac{\sum_{i=1}^{m'} \sum_{j=L}^{m-1} \chi_{j+1}(s\sigma)}{\sum_{i=1}^{m'} \sum_{j=L}^{m-1} \chi_j(s)} \quad (2)$$

Неформально опишем работу базового алгоритма построения PST. Его работа начинается с дерева с одним корневым узлом ***e***. Потом в дерево добавляются узлы, которые, по нашему мнению, должны ему принадлежать. Так, узел с меткой *s* становится листом дерева, если эмпирическая вероятность $\tilde{P}(s)$ не является пренебрежимой и для некоторого символа σ эмпирическая вероятность $\tilde{P}(\sigma \mid s)$ существенно отличается от эмпирической вероятности получить его после суффикса *s* $\tilde{P}(\sigma \mid suffix(s))$, то есть *s,* а не *suffix(s)* является определяющим контекстом для σ. Алгоритм завершает свою работу, если больше не существует листа, для которого справедливы приведенные условия или он достиг ограничения на максимальную глубину дерева *L*. В работе алгоритма также используется вспомогательный набор величин: $\varepsilon_0, \varepsilon_1, \varepsilon_2, \varepsilon_3$ и $\gamma_{\min}$, которые являются простыми функциями от $\varepsilon, \delta, n, L$ и $|\Sigma|$, конкретный вид которых устанавливается в ходе анализа алгоритма (подробнее см. [7]).

Теорема 2 [7]. *Для любых PSA M и параметров* $\varepsilon > 0$ *и* $0 < \delta < 1$ *алгоритм строит PST* $\hat{T}$ *такое, что с вероятностью как минимум* $(1-\delta)$, *дерево* $\hat{T}$ *— ε-хорошая гипотеза относительно M. Если алгоритм имеет доступ к источнику независимо сгенерированных строк, то время его работы ограничено полиномом от* $L, n, |\Sigma|, 1/\varepsilon, 1/\delta$

.

Для доказательства этой теоремы используется вспомогательная

Лемма 1. *Существует полином m_0 такой, что набор примеров количеством $m > m_0(L, n, |\Sigma|, \frac{1}{\delta}, \varepsilon_0, \varepsilon_1, \varepsilon_2)$ строк, каждый длиной большей (L+1), является типичным с вероятностью, не меньшей $(1 - \delta)$* .

Неформально говоря, набор строк считается типичным, если для каждой подстроки символов сгенерированной *M* ее эмпирические вероятность и вероятность следующего символа, при условии этой подстроки, близки к своим математическим ожиданиям.

Надо указать, что в определении эмпирических вероятностей и, соответственно, в определении 4 вероятность $\tilde{P}(s)$ в случае нескольких строк является суммой независимых случайных ограниченных величин. Для определения $\tilde{P}(s)$ все слагаемые добавляются с одинаковыми весовыми коэффициентами $1/m'(l - L + 1)$. Это дает возможность применить известные неравенства для сумм независимых случайных величин. Так, для доказательства леммы 1 используется вариант неравенства Хеффдинга [5], которое дает оценку сверху вероятности того, что сумма независимых ограниченных случайных величин не превышает своего математического ожидания.

МОДИФИКАЦИЯ ИСХОДНОЙ МОДЕЛИ

Мы модифицировали алгоритм вычисления эмпирических вероятностей таким образом, чтобы вклад более ранних примеров в совокупную эмпирическую вероятность с каждым шагом уменьшался. Более поздние примеры будут учитываться с большими весовыми коэффициентами, и так можно смоделировать "забывание" ранних примеров.

Задавшись определенным коэффициентом обучения $0 < \alpha < 1$, положим эмпирическую вероятность состояния с меткою *s* в момент времени *t*

$$\tilde{P}_1(s) = P'_1(s), \tilde{P}_t(s) = \alpha \tilde{P}_{t-1}(s) + (1 - \alpha) P'_t(s) = \alpha^{t-1} P'_1(s) + (1 - \alpha) \sum_{\tau=2}^{t} \alpha^{t-\tau} P'_\tau(s), \qquad (3)$$

где $P'_t(s)$ — эмпирическая вероятность состояния в строке, которая поступила в момент времени *t*. Величина коэффициента α регулирует скорость забывания: значения, близкие к нулю, приведут к быстрому забыванию, а значения, близкие к единице — к медленному.

Предположим, что в момент времени *t=1* вероятностное суффиксное дерево, по примерам работы которого мы стараемся построить к нему аппроксимацию, скачкообразно изменилось. Будем считать, что в нашей предметной области смоделировано ситуацию, когда пользователь изменяет свое поведение или мы начали наблюдать поведение другого субъекта системы, который идентифицирован как наш пользователь. Надо ответить на вопрос: способен ли наш измененный алгоритм приспособиться к новому поведению и обеспечить необходимую адаптивность модели, оставляя правильным результат теоремы 2.

Обозначая эмпирические вероятности на шаге *t* как X_t, положим $S_t = \alpha^t X_0 + (1 - \alpha) \sum_{\tau=1}^{t} \alpha^{t-\tau} X_\tau$. Нами доказывается вариант неравенства Хеффдинга:

Теорема 4. *Для любых $\varepsilon > 0$ и $0 < \delta < 1$ существуют такие $0 < \alpha < 1$ и $t \geq 1$, что в случае скачкообразного характера изменения распределения величин X*

$$P\{|S_t - MX_t| \le \varepsilon\} \ge 1 - \delta$$

Доказательство похоже на доказательство теоремы 3 из работы [5] и не приведено из соображений экономии места.

Применив в доказательстве теоремы 2 (см. [7]) вместо леммы 1 результат теоремы 4 получим, что результат теоремы 2 остается правильным. С изложенной процедурой изменения вероятностей состояний, при скачкообразном характере изменения характеристик автомата *M*, алгоритм перестроит дерево $\hat{T}$ таким образом, что оно станет ε-хорошей гипотезой по отношению к новому измененному PSA *M*.

Модифицированный алгоритм

1. Получить T_{t-1} из предыдущего шага, текущую пример-сессию *r* и обновить эмпирические вероятности всех узлов дерева согласно правилу (3).

2. Рекурсивно удалить все листы из дерева для меток *s,* которых

$$\tilde{P}_t(\sigma) < (1 - \varepsilon_1)\varepsilon_0 \qquad (4)$$

3. Инициализировать множество $\bar{S} = \{s \mid s \in \Sigma^*, suffix(s) \in L(\hat{T}_{t-1}), \tilde{P}_t(\sigma) \ge (1 - \varepsilon_1)\varepsilon_0\}$, где $L(\hat{T}_{t-1})$ — множество листьев дерева $\hat{T}_{t-1}$.

4. Пока $\bar{S}$ не пусто, выполнять: Выбрать любую $s \in \bar{S}$ и

(a) удалить *s* из $\bar{S}$

(b) если существует $\sigma \in \Sigma$, такое что

$$\tilde{P}_t(\sigma \mid s) \ge (1 + \varepsilon_2)\gamma_{\min} \qquad (5) \qquad \text{и} \quad \frac{\tilde{P}_t(\sigma \mid s)}{\tilde{P}_t(\sigma \mid suffix(s))} > 1 + 3\varepsilon_2, \qquad (6)$$

тогда добавить узел, соответствующий *s*, в дерево и (возможно) все узлы на пути от самого глубокого узла в $\hat{T}$, метка которого является суффиксом *s*, до узла с меткой *s*.

(c) Если $|s| < L$, то для каждого $\sigma' \in \Sigma$, если $\tilde{P}(\sigma' s) \ge (1 - \varepsilon_1)\varepsilon_0$, то добавить строку с меткой $\sigma' s$ в $\bar{S}$.

Модифицированный алгоритм адаптивно перестраивает дерево согласно изменившимся вероятностям состояний. При этом состояния, которые по обновленным значениям вероятностей стали пренебрежимыми, (см. (4)) удаляются, а состояния, для которых условия (5) и (6) начали выполняться — добавляются.

ЭКСПЕРИМЕНТЫ

Из-за сложности получения искусственных данных со следами вторжений и спорности непосредственной экстраполяции результатов, полученных таким путем на реальные ситуации, было решено использовать аудиты сессий реальных пользователей. Эксперименты проводились на данных, полученных за год с UNIX сервера физико-технического факультета НТУУ "КПИ". Механизмами аудита операционной системы FreeBSD отслеживались все процессы, которые запускались от имени зарегистрированных в системе пользователей. Всего было получено данных для более чем 800 пользователей. Среди более 900 процессов, которые были использованы в течение периода наблюдения, мы оставили только те, интенсивность использования которых превышала 200 запусков за весь период (350 команд).

Интересно, что если бы мы использовали "полные" марковские цепи с фиксированным порядком, например $L = 5$, то понадобилось бы

$350^5 = 5\,252\,187\,500\,000$ состояний для построения соответствующего автомата, тогда как с использованием PST с таким же параметром L среднее количество состояний эквивалентного автомата среди всех пользователей не превышает нескольких тысяч.

Для каждой сессии пользователя вычислялась ее вероятность по формуле (1). Из-за того, что вероятность — это произведение чисел, меньших единицы, сессии с меньшим числом команд будут иметь бóльшую вероятность, чем с бóльшим. Поэтому коэффициентом "нормальности" было выбрано величину, которая (при одинаковых множителях) не зависит от длины последовательности — $K = \sqrt[N]{P_T^N(r)}$, где $P_T^N(r)$ — вероятность сессии r длины N.

Для каждого пользователя было построено отдельное PST по модифицированному алгоритму. На рисунках изображается процесс эволюции значения коэффициента K в течение определенного времени (полгода). По оси абсцисс откладывался номер сессии в порядке поступления, а по оси ординат — значения K. Для всех графиков было принято $L = 5$ и $\alpha = 0.99$.

Перекрестный тест

Как один из способов обнаружения аномалий можно предложить перекрестный тест, т.е. для текущей сессии вычисляется ее вероятность как если бы она была сгенерирована PST остальных пользователей и самого пользователя, от которого она была получена. Если максимальное значение вероятности достигается не на PST, отвечающем "хозяину" сессии, то могло иметь место вторжение. Мы провели эксперимент, показывающий, насколько надежной будет подобная классификация сессий (рис. 3). На рисунке цветом отмечены значения вероятностей — темные цвета соответствуют большим значениям, светлые — меньшим. Диагональные значения, соответствующие "своим" сессиям на "своих" PST, более чем в три раза превышают значения вне диагонали.

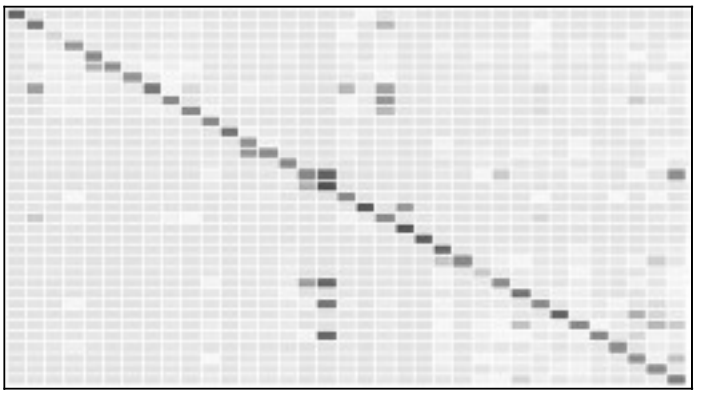

Рис. 3. Перекрестный тест для 35 случайных пользователей

Моделирование вторжения

Целью работы системы обнаружения аномалий на уровне пользователей есть ответ на вопрос, сгенерирована ли текущая сессия пользователем, от которого получено аудит сессии, или же от его имени выступало другое лицо. Распространенным примером вторжений, когда возникает потребность именно в таком ответе, является ситуация с украденным паролем. На рис. 2 изображены результаты обработки сессии с умышленно вставленными частями сессий, полученных от других пользователей.

Как хорошо видно на рисунках, "чужим" сессиям соответствуют меньшие значения K. Поэтому можно ввести, например, пороговую классификацию на нормальную или

аномальную сессии. Если значение коэффициента K превышает этот порог, сессия отмечается как нормальная, в противном случае — как аномальная. Однако если оба пользователя принадлежат к одному классу, то такая классификация может быть не очень надежной (последний график на рис. 2; оба пользователя — студенты с похожими привычками).

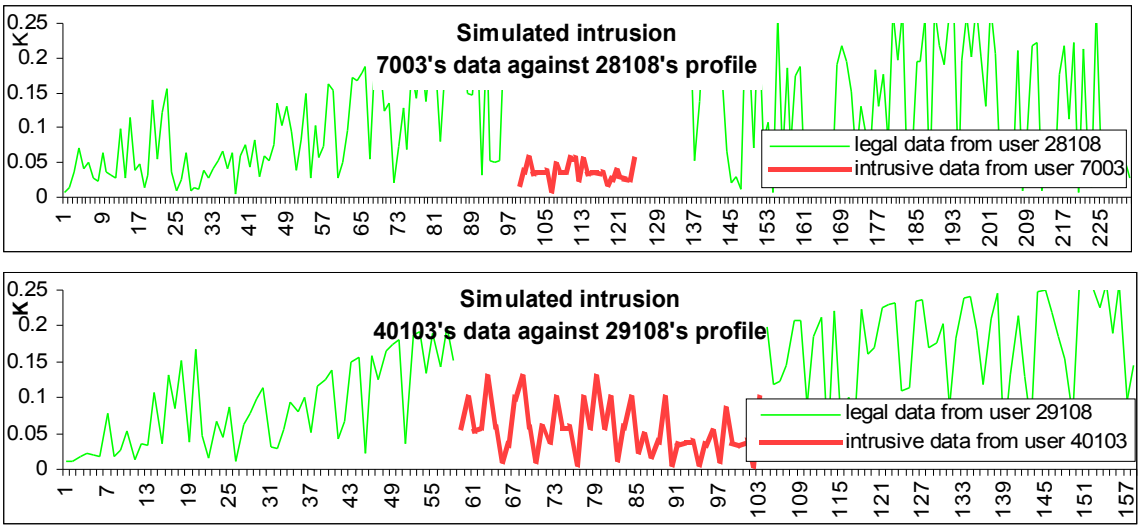

Рис. 2. Примеры подмены пользователя. Моделирование ситуации с украденным паролем.

Replay-атаки

Одним из недостатков существующих систем обнаружения аномалий есть уязвимость по отношению к т.н. replay-атакам. Они характеризуются тем, что злоумышленник, получив доступ к аудит-файлам легального пользователя, "повторяет" его сессии, заменив незначительную их часть или вставив нужные ему команды. Из-за пренебрежимо малого количества добавленных команд IDS пропускают такую сессию как нормальную.

Возможность преобразования PST в PSA [7] позволяет нам не только указать на модифицированную сессию, но и выделить из нее "чужие" команды. Идея — рассматривать полученную сессию как зашумленную и стараться восстановить наиболее вероятную последовательность состояний автомата, через которые он прошел, чтобы восстановить ее незашумленный вариант. Эта и подобные задачи сводятся к известной задаче на разметку структур типа цепочки или простой цепи [8].

Предположим, что экспертным путем получено информацию, по которой вероятность замены команды в сессии составляет $0 < \rho < 1$. То есть, интерпретируя PST как автомат, генерирующий последовательности команд, имеем, что, находясь в состоянии k с меткою s, автомат с вероятностью $(1-\rho)P(\sigma^* \mid k)$ перешел в состояние k^{**} с меткою $suffix(s\sigma^*)$, где σ^* — следующая команда в текущей сессии; или с вероятностью $\rho P(\sigma' \mid k)$ — в некоторые состояния k' с метками $suffix(s\sigma')$, где $\sigma' \in \Sigma \setminus \{\sigma^*\}$. После построения такого автомата ставится задача найти наиболее вероятную последовательность его состояний $\bar{k} = \arg\max_{k \in \Sigma^N} \sum_{k_0 \in \Sigma} P_0(k_0) \prod_{i=1}^{N} P(x_i, k_i \mid k_{i-1})$, где

$P_0(k_0)$ — вероятность первой команды, которая определяется из переходных вероятностей узла с меткой начала сессии. Решая эту задачу по известным алгоритмам [8], получаем искомую последовательность состояний $\bar{k}$. Если для некоторого i состояние $\bar{k}_i$ имеет метку s, последний символ которой не совпадает с имеющимся для этого места в сессии, значит, здесь имела место замена команды.

Когда вероятно как изменение команды в неизвестном месте, так и вставка новой команды с "раздвиганием" соседних команд, алгоритм поиска наиболее вероятной последовательности состояний обобщается с топологий типа цепь на простые сети [8].

Значения параметров модели

Как и ожидалось, близкие к единице значения α приводят к медленной адаптации и медленному росту значений коэффициента K, тогда как с меньшими значениями α процесс адаптации ускоряется. Выбирая конкретное значение коэффициента α, надо придерживаться индивидуального подхода, поскольку у разных людей поведение меняется по-разному. Важно не задавать слишком малое α поскольку это может привести к повышению уровня ошибок типа false negatives — ошибок, когда аномальное поведение не замечено.

Оптимальным значением L для данного пользователя можно считать такое значение, при котором характеристики построенного PST уже не улучшаются, т. е. при дальнейшем увеличении L значения коэффициента K не увеличиваются. Как и параметр α — это индивидуальная характеристика каждого пользователя.

ЗАКЛЮЧЕНИЕ

Применение адаптивной модели марковсих цепей с переменным размером памяти в системах обнаружения аномалий предлагается впервые. В работе мы предложили несколько возможных технологий обнаружения аномалий и привели их экспериментальное обоснование, а именно пороговую классификацию, перекрестный классификационный подход и подход, который дает возможность обнаруживать аномальные replay-сессии и конкретизировать, в чем именно заключается аномальность. Все предложенные технологии могут быть использованы в реальных системах обнаружения аномалий.

ЛИТЕРАТУРА

[1] А. М. Резник, Н. Н. Куссуль и А. М. Соколов. Нейросетевая идентификация поведения пользователей компьютерных систем. *Кибернетика и вычислительная техника*, (123): 70-79, 1999.

[2] B. D. Davison and H. Hirsh. Predicting sequences of user actions. In *Predicting the Future: AI Approaches to Time-Series Problems*, pages 5-12, Madison, WI, July 1998. AAAI Press. Proceedings of AAAI-98/ICML-98 Workshop

[3] D. E. Denning. An intrusion-detection model. In *Proc. IEEE Symposium on Security and Privacy*, pages 118-131, 1986.

[4] S. Forrest, S. A. Hofmeyr, A. Somayaji, and T. A. Longstaff. A sense of self for Unix processes. In *Proceedings of the 1996 IEEE Symposium on Research in Security and Privacy*, pages 120-128. IEEE Computer Society Press, 1996.

[5] W. Hoeffding. Probability inequalities for sums of bounded random variables. *Journal of the American Statistical Association*, 58 (301): 13-31, 1963.

[6] T. Lane. Hidden markov models for human/computer interface modeling. In *IJCAI-99 Workshop on Learning About Users*, pages 35-44, 1999.

[7] D. Ron, Y. Singer, and N. Tishby. The power of amnesia: Learning probabilistic automata with variable memory length. *Machine Learning*, 25 (2-3): 117-149, 1996.

[8] M. I. Schlesinger and V. Hlavač. *Deset prednasek z teorie statistickeho a strukturniho rozpoznavani*. Vydavatelstvi CVUT, Praha, 1999.

[9] D. Wagner and R. Dean. Intrusion detection via static analysis. In F. M. Titsworth, editor, *Proceedings of the 2001 IEEE Symposium on Security and Privacy*, pages 156-169, Los Alamitos, CA, May 14-16 2001. IEEE Computer Society.

[10] N. Ye. A markov chain model of temporal behavior for anomaly detection. In *Proceedings of the 2000 IEEE Systems, Man, and Cybernetics, Information Assurance and Security Workshop*, 2000.