УДК 621.391

ИНТЕЛЛЕКТУАЛЬНЫЙ КЛАССИФИКАТОР ОБНАРУЖЕНИЯ ВТОРЖЕНИЙ НА БАЗЕ ГЕНЕТИЧЕСКИХ АЛГОРИТМОВ

М.Г. МОЗДУРАНИ ШИРАЗ, В.А. ВИШНЯКОВ

Белорусский государственный университет информатики и радиоэлектроники, Республика Беларусь

Поступила в редакцию 01 ноября 2019

Аннотация. Предложена модель обнаружения вторжений на основе генетических алгоритмов. Найдена лучшая хромосома в цикле эволюции, которая применена к тестовому набору данных типовых атак KDD. Построен классификатор на базе модели, для которого получена хорошая степень распознавания атак и показана высокая скорость обнаружения вторжений, продемонстрировав результаты лучше, чем аналогичные. Также этот классификатор может обнаружить значительный процент атак R2L, который выше, чем у аналогов, разработанных ранее.

Ключевые слова: обнаружение вторжений, генетические алгоритмы, набор данных атак KDD, классификатор атак.

Введение

С развитием новых инфокоммуникационных технологий (ИКТ) и расширением сетевых компьютерных систем при значительном приросте интернет-трафика, конфиденциальные данные находятся под постоянной угрозой от различных атак. Вторжения являются распространенными угрозами для сетей, и с ростом количества злоумышленников, разработка эффективных инструментов безопасности затрудняется. Современные инструменты безопасности, такие как межсетевые экраны (файрволлы), системы обнаружения вторжений (IDS) и другие средства, недостаточны для обеспечения целостности безопасности. Возникает необходимость в обнаружении неизвестных вредоносных вторжений, т.е. вторжения становятся все более важной технологией, которая отслеживает сетевой трафик и распознает незаконное использование локальных компьютерных систем.

Разница между известными продуктами безопасности и системами обнаружения вторжений заключается в том, что последним требуется больше интеллекта. Они должны анализировать собранную информацию и выводить результаты в отсутствии сигнатур атак в базе. Основная проблема заключается в том, что эти системы не могут обнаружить новые атаки без известных шаблонов, а также производные шаблоны известных атак [1]. В последнее время IDS стали одним из важных предметов исследований в области информационной безопасности (ИБ), новые исследования сосредоточены на том, как сделать IDS интеллектуальными. Т. е. улучшить их таким образом, чтобы иметь возможность автоматически обнаруживать новые формы атак. Эти исследования сосредоточены на объединении методов искусственного интеллекта с механизмами обнаружения вторжений для повышения точности идентификаторов (т. е. производительности обнаружения), эффективности и удобства использования. Некоторые исследования используют методы интеллектуального анализа данных, другие – нейронные сети или генетические алгоритмы. Анализ систем обнаружения вторжений (СОВ) приведен в работе [2]. Ниже рассматривается построение интеллектуальной СОВ на базе генетических алгоритмов.

Описание наборов данных KDD99

Набор данных KDD99 типовых атак [2] очень важен для оценки идентификаторов обучения, он является эталоном для тестирования IDSs. Атаки этого набора подразделяются на четыре основные категории:

− DOS: отказ в обслуживании, например, SYN flood;

− R2L: несанкционированный доступ с удаленного компьютера, например, угадывание пароля;

− U2R: несанкционированный доступ к привилегиям локального пользователя (root), например, различные атаки переполнения буфера;

− Зондирование: наблюдение и другое зондирование, например, сканирование портов.

Набор данных KDD99 имеет два типа данных:

− набор данных обучения: он используется для обучения любого алгоритма идентификаторов обучения, поскольку он содержит тип атаки для каждого соединения. Этот набор данных имеет 24 типа обучающих атак и содержит 4898431 соединения.

− набор данных тестирования: используется для проверки эффективности идентификаторов в исследованиях, например, для проверки способности нового алгоритма обнаружения выявлять атаки в наборе данных обучения, кроме новых 19 атак, которые включены в набор данных тестирования. Набор данных тестирования содержит 311 029 соединений.

В данных KDD99 каждое соединение имеет 42 функции (включая метку класса), которые содержат информацию о сеансе. Эти функции используются для распознавания нормальных соединений от атакующих; их можно разделить на четыре категории.

1. Basic: содержит основные характеристики записи подключения.

2. Содержимое: создаются из полезной нагрузки пакетов трафика и содержат сведения, связанные с хостом, такие как количество сбоев входа в систему.

3. Трафик: содержит статистические данные, такие как количество подключений к одному и тому же хосту в двухсекундном временном окне.

4. Класс: указывает, является ли соединение нормальным или интрузивным; используется для обучения и оценки.

Алгоритм обучения классификатора

В задаче проектирования интеллектуальной системы обнаружения вторжений она была подвержена воздействию атак из наборов данных KDD99 [2]. Набор обучающих данных содержит около 5 миллионов записей подключения, каждая запись содержит 41 функцию. Процесс занимает много времени, чтобы обучить систему с учетом всех особенностей. Но если бы можно уменьшить количество функций, например, до 8 функций, сильно уменьшается количество вычислений.

Пространство набора данных можно разделить на:

− 2 основных класса: обычный трафик и атака трафик;

− более точно – 5 классов: нормальный, DoS, R2L, U2R, зонд.

Таким образом, пространство набора данных может быть описано следующими величинами:

X: случайная величина = {нормальный, DOS, U2R, R2l, зонд}

Y: функция подключения: (41 значение независимых случайных величин)

Пример: *Y*(Protocol_type) = {ICMP, TCP, UDP}

Теперь, чтобы извлечь значение функции соединения (*X*. potocol_type), можно вычислить «объем информации о *X* (нормальное соединение или DOS, R2L, U2R, зонд), содержащийся в *Y* (функция соединения *X* тип протокола)».

Оценка эффективности алгоритма

Подход к генетическому алгоритму (ГА) зависит от результатов функций селекции, имеющих параметры: длительность, обслуживание, байты источника (src_bytes), байты результата (dst_bytes) и количество [4]. Генетический классификатор называется *GACL* и может быть записан в виде:

$$GACL(conn_j) = \begin{cases} if \ \sum_{i=1}^{n} W_i f_i(conn_j) < threshold \ \ then \ \ conn_j \ \ is \ \ ATTACK \\ else \ \ conn_j \ \ is \ \ NORMAL \end{cases},$$

где $conn_j = connection_j = (\text{duration, service, src\_bytes, dst\_bytes, count})$.

Неравенство было выбрано по причине того, что для предыдущих признаков большинство атакующих соединений в обучающем наборе KDD имеют низкие значения. Например, набор данных обучения KDD содержит 396743 записи атак, 396083 из которых имеют значение длительности, равное 0.

Шесть параметров управления доступом кодируются следующим образом:

– пять параметров представляют веса объектов $(W1, W2, W3, W4, W5)$;

– один параметр представляет собой пороговое значение.

Каждый параметр представляет собой двоичный ген длиной 11 бит: 10 бит для значения параметра, один бит для знака. Число 10 бит (значение параметра) было выбрано с точностью $210 \approx 0,001$. Пороговое значение остается всегда положительным.

В качестве примера можно получить двоичное значение гена следующим образом:
$Gene1 = 11000000010 = 1\&1000000010$ (11-й бит = 1, так что знак отрицательный).
Десятичное значение гена равно:

$$Gene1 = 1000000010 = -1 \cdot 29 + 0 \cdot 28 + 0 \cdot 27 + 0 \cdot 26 + 0 \cdot 25 + 0 \cdot 24 + 0 \cdot 23 + 0 \cdot 22 + 1 \cdot 21 + 0 \cdot 20 = -514.$$

В табл. 1 показаны хромосомные гены.

Таблица 1. **Кодирование параметров**

Гены	Параметры генов	Границы генов	Значения представлений	
			Двоичное	Десятичное
*Gene*1	$W_{duration}$	[–1023,1023]	11000000010	–514
*Gene*2	$W_{service}$	[–1023,1023]	01000000010	514
*Gene*3	$W_{src\ bytes}$	[–1023,1023]	00000000010	2
*Gene*4	$W_{dst\ bytes}$	[–1023,1023]	11111111111	–1023
*Gene*5	W_{count}	[–1023,1023]	01111111111	1023
*Gene*6	*Threshold*	[0,1023]	00101011010	532

Хромосома предыдущих генов имеет вид, представленный в табл. 2.

Таблица 2. **Хромосома предыдущих генов**

*Gene*6	*Gene*5	*Gene*4	*Gene*3	*Gene*2	*Gene*1
00101011010	01111111111	11111111111	00000000010	01000000010	11000000010
532	1023	–1023	2	514	–514

Таким образом значение *GACL* для этой хромосомы:

$$GACL(conn_j) = \begin{cases} if \ \ -514\, f_{duration} + 514\, f_{service} + 2\, f_{src\_byte} - 1023\, f_{dst\_byte} + 1023\, f_{count} < 532 \\ then \ \ conn_j \ \ is \ \ ATTACK \ \ \ else \ \ conn_j \ \ is \ \ NORMAL \end{cases}.$$

Перед определением структуры программы для работы классификатора приведем основные понятия из теории ГА [3]. Гены – это параметры равные сумме веса и порога. Хромосома – это решение – генетический классификатор *GACL*. Индивид – это то же самое, что хромосома с дополнительными переменными для хранения значения пригодности, показателей производительности и других параметров. Потомство – это ребенок, полученный в результате операции скрещивания (кроссовера). Популяция – совокупность индивидов.

Поколение – это то же самое, что и популяция с дополнительными переменными для хранения всей пригодности поколения, лучшего и худшего индивидуума и других.

Запуск: программа запуска операции эволюции начинается с генерации начальной случайной популяции, проходит через создание нескольких поколений и заканчивается созданием последнего поколения, которое содержит искомое решение, если оно существует. На рис. 1 представлена структура программы.

```
Type INDIVIDUAL      // Структура
  crom (1..gensNb*genLength) As Byte  // Классификатор GACL   11010101000010011000 1100
  decCode (1. . gensNb) As Integer        // десятичное значение (512  340  211 -20 -10 20)
  fitVal As Double                        // значение функции пригодности
  fitProb As Double                       // вероятность пригодности общей приспособленности популяции
  reFit As Boolean                        //индикатор – нужно ли пересчитывать значение пригодности
  TP As Long        // number количество атакующих соединений, обнаруженных данной хромосомой
  TN As Long        // # of нормальных связей, которые были обнаружены этой хромосомой
  FP As Long        // # ложноположительные связи этой хромосомы
  FN As Long        // # ложноотрицательные связи этой хромосомы
```

Рис. 1. Структура программы

При запуске операции эволюции все результаты будут сохранены в базе данных в соответствии с предыдущей структурой данных, что позволит извлечь и построить много полезных статистических данных о найденных решениях. Предложенная функция пригодности совпадает со скоростью классификации

$$fitVal = fitnessValue = CR = \frac{number\ of\ \ correctly\ detected\ connections}{size\ of\ \ dataset} = \frac{TP + TN}{size\ of\ \ dataset}.$$

Поскольку каждая хромосома в популяции является решением (классификатором), ее значение CR будет вычисляться в соответствии с обучающим набором данных KDD – 10 %, который содержит объекты соединений и метки (normal, attack).

Ниже приводится краткое представление о генетических операциях, используемых в предлагаемом обучении ГА: отбор, кроссовер и мутация [4].

Отбор – это действие выбора родителей, т.е. как выбрать индивидов в популяции, которые создадут потомство для следующего поколения, и сколько потомков создаст каждый. Цель отбора состоит в том, чтобы выделить более приспособленных особей в популяции в надежде, что их потомство, в свою очередь, будет иметь еще более высокую приспособленность. Существует множество стратегий выбора, таких как: «колесо рулетки», «масштабирование Сигмы», элитизм и др. [3].

Был принят метод выбора «колеса рулетки», потому что он прост в реализации и быстро сходится, но недостатком которого является то, что, если, например, индивидуальная пригодность составляет 90 %, существует низкий шанс для выбора других особей. Основная идея этого метода отбора заключается в том, что лучшие особи получают более высокие шансы, т.е. шансы пропорциональны значению пригодности. Отметим, что главной отличительной особенностью ГА является использование кроссовера (скрещивания).

Одноточечный кроссовер: это самая простая форма. Одна позиция пересечения выбирается случайным образом, и части двух родителей после позиции пересечения обмениваются, чтобы сформировать два потомства.

Двухточечный кроссовер: две позиции выбираются случайным образом, и сегменты между ними обмениваются.

N-точечное пересечение: выбирается *n* случайных точек пересечения, затем их разделяют вдоль этих точек и делают пересечение. Это обобщение одного точечного кроссовера

Использован двухточечный кроссовер для создания потомства, поскольку он лучше, чем одноточечный кроссовер и быстрее, чем *n*-точечный или равномерный кроссовер. Операция кроссовера будет выполнена в соответствии с вероятностью – Pc.

Предложенный ГА был выполнен с использованием следующих параметров:

Длина хромосомы – 66, количество генов – 6, размер популяции – 500, число поколений – 160, мутационная способность *Pm* – 0,02, вероятность кроссовера *Pc* – 0,7.

HighCR – это самая высокая скорость классификации и она представляет собой критерий остановки, может быть, например, в диапазоне от 0,975 до 1. Если это не достигнуто, то ГА остановится, когда будет достигнут номер предопределенного поколения. В процессе моделирования лучшая найденная хромосома имела вид:

00111010001111100101011000110000001110111001011010010111000011001111.

Для нее значение классификатора:

$$GACL(conn_j) = \begin{cases} if \;\; 103 f_{duration} + 843 f_{service} + 953 f_{src\_byte} - 96 f_{dst\_byte} - 917 f_{count} < 465 \\ \\ \quad then \;\; conn_j \;\; is \;\; ATTACK \quad else \;\; conn_j \;\; is \;\; NORMAL \end{cases}.$$

Этот классификатор был применен для обнаружения атак в тестовом наборе данных KDD99, и были получены результаты, представленные в табл. 3.

Таблица 3. **Показатели эффективности классификатора линейного генетического алгоритма**

Класс атаки	Записи	Обнаружено	Соотношение	
			DR	NDR
NORMAL	60593	58330	96,27 %	–
DOS	229853	222142	96,65 %	16,34 %
PROBE	4166	3227	77,46 %	92,34 %
R2L	16347	4953	30,30 %	0,40 %
U2R	70	53	75,71 %	77,42 %
TP = DR	91,99 %			
NDR	14,89 %			
TN	96,27 %			
CR	92,82 %			
FP	3,73 %			
FN	8,01 %			

Сравнивая эти результаты с предыдущими идентификаторами, можно увидеть, что *GACL* имеет меньшую частоту обнаружения для нормальных, DOS и зондовых соединений, но имеет и другие преимущества:

– скорость выше, чем SF-5NN и SUS-5NN;

– лучшее обнаружение новых атак равно 14,89 %, в то время как в SF-5NN оно составляет 11,81 %;

– лучше частота выявления R2L, равная 30,30 %, а в SF-5NN – 7,99 %, причем атака R2l более опасна, чем DOS и зондовые атаки.

Заключение

1. Построен линейный классификатор на базе теории генетических алгоритмов, оценивающий суммы взвешенных признаков связи в сети. Если сумма была меньше пороговой, то соединение рассматривалось как атакуемое. Параметры, определяющие работу фильтра – сумма весов и порога. Для получения этих параметров был использован бинарный генетический алгоритм, в котором гены равны параметрам. Функция пригодности была определена как скорость классификации (*CR*), цикл эволюции занял 160 поколений, каждое поколение состояло из 500 хромосом.

2. Найденная лучшая хромосома *GACL* была применена к тестовому набору данных типовых атак KDD. Получена хорошая скорость классификации – 92,82 %, другим важным результатом была скорость обнаружения атаки, которая оказалась лучше, чем у известных решений (14,89 %). Также этот классификатор может обнаружить 30,30 % опасных атак R2L. Этот показатель лучше, чем имеют аналогичные подходы, разработанные ранее.

INTELLIGENT CLASSIFIER OF INTRUSION DETECTION ON BASES OF GENETIC ALGORITHMS

U.A. VISHNIAKOU, M.M. MOSDURANY SHIRAS

Abstract. Some detection model was proposed using genetic algorithms and KDD99 data sets. The best chromosome in evolution cycle was founded, which to test set of typical attacks was used. The classifier on the model base was designed, for which good result of attack detection was received and high speed of intrusion detection was shown. These results are better than analogous. So this classifier may detect some percent of R2L attacks high than analogue was designed.

Keywords: intrusion detection, genetic algorithms, attack set data KDD99, classifier.

Список литературы

1. Seagren S. Secure Your Network for Free. Syngress Publishing. 2007.
2. Lazarević A., Srivastava J., Kumar V. Data Mining For Intrusion Detection, Army High Performance Computing Research Center, Department of Computer Science, University of Minnesota, 2003.
3. Melanie M. An Introduction to Genetic Algorithms. Cambridge, Massachusetts London, England. 1999.
4. Вишняков В.А., М.Г. Моздурани Шираз. // Современные средства связи: материалы 24-й междунар. науч. конф. Минск. РИВШ. 2019. С.160.