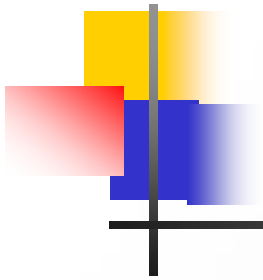


СПИИРАН

ОБНАРУЖЕНИЕ ВРЕДОНОСНОГО ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ НА БАЗЕ МЕТОДОВ ИНТЕЛЛЕКТУАЛЬНОГО АНАЛИЗА ДАННЫХ

Котенко И.В., Шоров А.В.

Санкт-Петербургский институт информатики и автоматизации РАН

Комашинский Д.В.

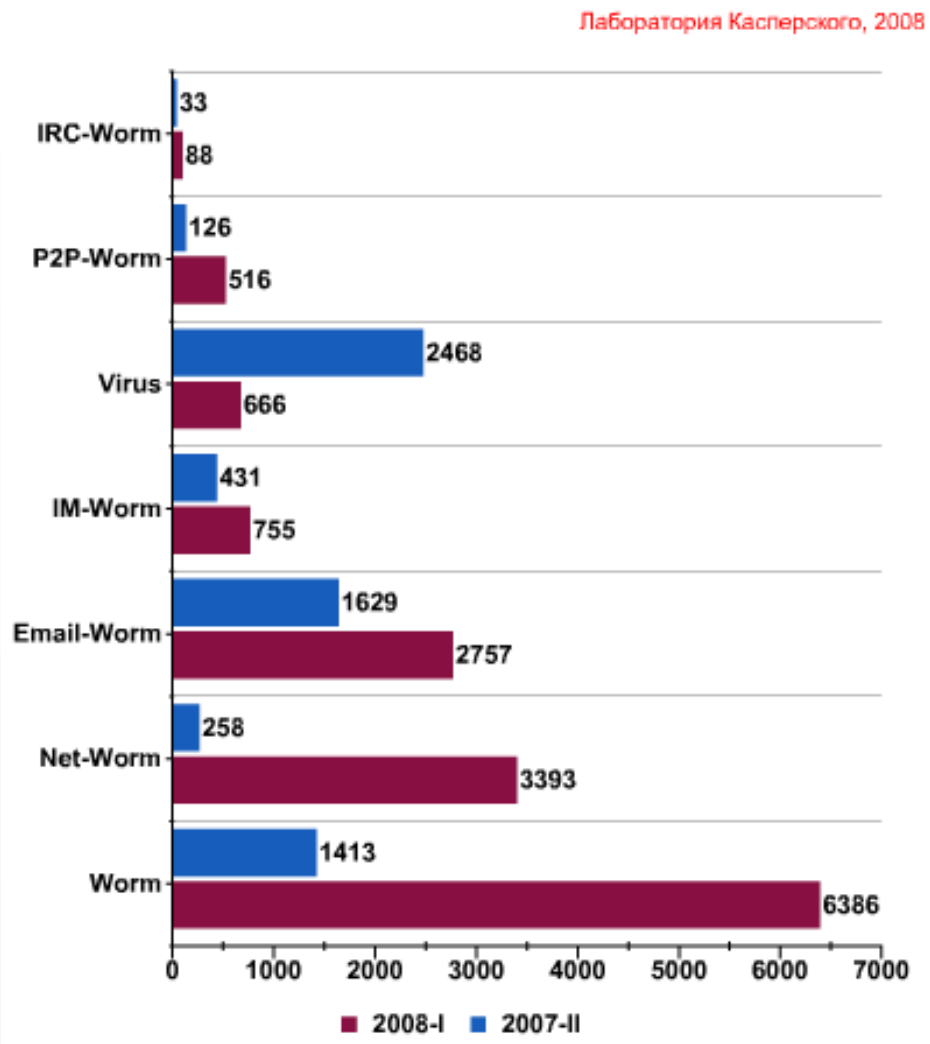
ЗАО “Аркадия”

РусКрипто’2009, 2-5 апреля 2009 г.

Содержание

- Введение. Сущность и особенности задачи обнаружения malware на основе методов интеллектуального анализа данных
- Релевантные работы
- Сущность предлагаемого подхода
- Комплекс моделирования и эксперименты
- Заключение

Основные тенденции в области разработки и противодействия malware



Несмотря на наличие позитивных сдвигов, статистические данные в целом показывают тенденцию роста количества зарегистрированных вредоносных приложений, и, как следствие, подтверждение наличия проблем противодействия ему.

Актуальность задачи обнаружения malware

Ввиду долговременности процессов совершенствования законодательной базы и продвижения принципов соблюдения «гигиены» в вопросах информационной безопасности, а так же продолжения развития malware-технологий в ближайшей перспективе **наиболее актуальным направлением исследований и разработок остается совершенствование технологического базиса противодействия malware.**

Актуальность задачи обнаружения malware

Принимая во внимание фактор **недостаточности существующих методов обнаружения malware** (использование баз сигнатур не является панацеей из-за наличия временной дельты от момента обнаружения malware аналитиками до момента поставки обновления конечному пользователю) и **сложности точной настройки** существующих проактивных средств детектирования, **задача совершенствования эвристических средств обнаружения malware представляется необходимой.**

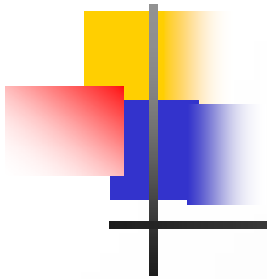
Сущность и особенности задачи обнаружения malware

Типовое программное обеспечение, обеспечивающее **защиту от malware**, выполняет **три основные функции**:

- обнаружение;
- идентификация;
- дизинфекция.

Функция обнаружения обеспечивает возможность принятия решения о том, является ли некоторый программный код вредоносным или нет.

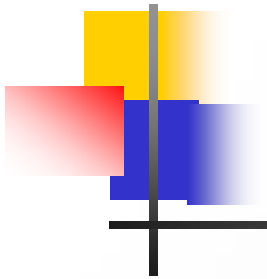
Точное обнаружение новых вредоносных программ по статическим признакам или поведению (динамическим признакам) **невозможно**, так как автор malware всегда обладает большим количеством степеней свобод в части реализации.



Сущность и особенности задачи обнаружения malware

Очевидно, что более эффективным с точки зрения безопасности является обнаружение malware без его выполнения, так как в данном случае атакуемый хост не подвергается деструктивному воздействию.

Данное условие породило группу статических методов обнаружения.



Сущность и особенности задачи обнаружения malware

Однако существуют мощные средства сокрытия кода (упаковщики, протекторы), наличие которых существенно затрудняет использование статических методов.

Симметричным ответом на данные трудности явилось появление методов, обеспечивающих обнаружение при выполнении проверяемого кода или как его еще называют, «на лету».

Решение проблемы безопасности для методов второй группы решается, как правило, за счет формирования изолированной вычислительной среды программным или организационным путем.

Треугольник качества задачи детектирования

Начальное состояние:

$S_0 = [A_0, C_0, P_0]$;

Оптимизация:

O target: A to max; P to save.

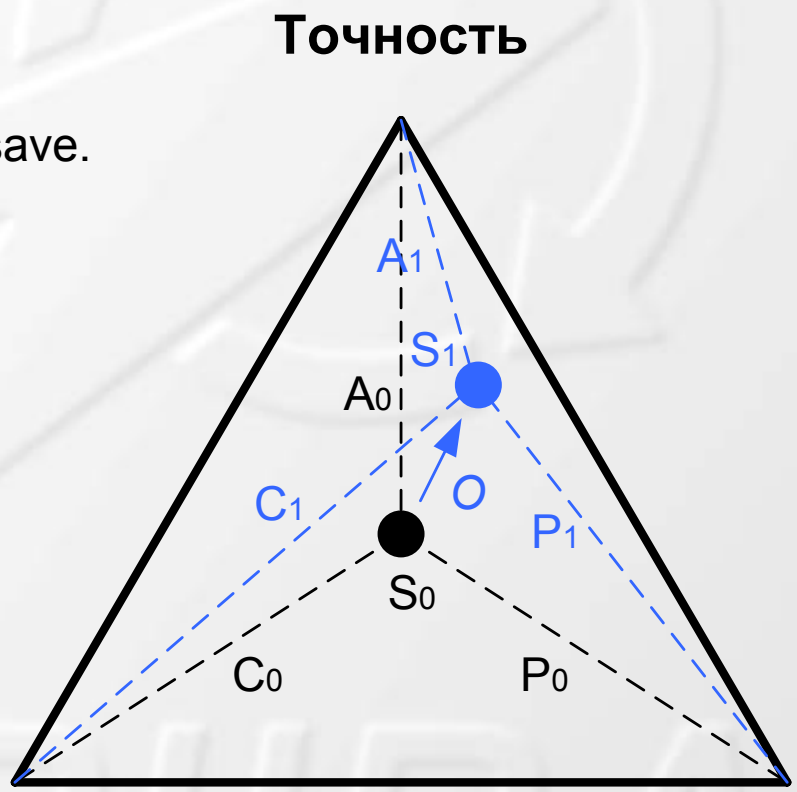
$S_1 = O(S_0) = \{A_1, C_1, P_1]$;

Результат:

A_1 better than A_0;

P_1 is close to P_0;

S_1 worse than S_0.



Задача проактивного обнаружения malware

- **Проактивность** <-> реактивность
- **Проактивное обнаружение основано** на *автоматических механизмах использования информации об "истории" анализируемых событий и прогнозе будущих событий, а также автоматической подстройке (обучения) параметров и методов обнаружения*
- Для обнаружения malware необходимо решать **задачу "добычи" (data mining) и объединения (data fusion) данных большой размерности**.
- **Задача обучения обнаружению malware** может рассматриваться как приложение knowledge and data discovery (KDD), но является достаточно **специфической задачей.**

Подходы к обнаружению malware

- **Подход, ориентированный на данные,** - основан на выявлении полезных паттернов из зафиксированных данных, представляющих собой примеры обнаруженных злоупотреблений и аномалий. Задача обучения обнаружению вторжений рассматривается в виде процесса KDD (S.Stolfo, P.Chan, W.Lee, *et al*).

- **Подход, основанный на компьютерной иммунологии,** - основан на использовании базовых принципов функционирования иммунной системы человека для обучения обнаружению аномалий (S. Forrest, D.Dasgupta, etc.) посредством нахождения паттернов нормального использования в зафиксированных данных регистрации (offline audit data) и сравнения их с паттернами, обнаруженными в текущих соединениях (online audit data).

Релевантные работы. Пример 1 (1/3)

Источник: Schultz M.G., Eskin E., Zadok E., Stolfo S.J. Data Mining Methods for Detection of New Malicious Executables // Informatics and Computer Science, Volume 172, Issue 1-2, 2005. P.241-261.

Цель: обнаружение неизвестных экземпляров malware, распространяющихся по каналам электронной почты

Обучающий набор: файлы формата PE32 (Portable executable), 3265 файлов malware, 1001 безопасных.

Выделение признаков (feature extraction):

статические данные, полученные при парсинге файлов:

- данные импорта (используемые функции модулей операционной системы);
- строки;
- байтовые последовательности из секции кода.

Релевантные работы. Пример 1 (2/3)

Выделение значимых признаков (feature selection): на основе оценки частоты проявления тех или иных признаков для вредоносных и безопасных PE32 файлов из обучающей выборки.

Использованные классификаторы:

- RIPPER, индуктивный основанный на правилах (inductive rule-based);
- Naive Bayes, статистический;
- Multi-Naive Bayes, статистический мультипликативный.

Оценка построенных математических моделей обнаружения: вычисление значений ошибок первого и второго рода (false negative, false positive) кросс-теста.

Релевантные работы. Пример 1 (3/3)

Результаты оценки:

Profile Type	True Positives (TP)	True Negatives (TN)	False Positives (FP)	False Negatives (FN)	Detection Rate	False Positive Rate	Overall Accuracy
Signature Method — Bytes	1102	1000	0	2163	33.75%	0%	49.28%
RIPPER							
— DLLs used	22	187	19	16	57.89%	9.22%	83.62%
— DLL function calls	27	190	16	11	71.05%	7.77%	89.36%
— DLLs with counted function calls	20	195	11	18	52.63%	5.34%	89.07%
Naive Bayes — Strings	3176	960	41	89	97.43%	3.80%	97.11%
Multi-Naive Bayes — Bytes	3191	940	61	74	97.76%	6.01%	96.88%

Релевантные работы. Пример 2 (1/3)

Источник: Wang J.-H., Deng P.S., Fan Y.-S., Jaw L.-J., Liu Y.-C. Virus Detection using Data Mining Techniques // Proceedings. IEEE 37th Annual 2003 International Carnahan Conference, 2003. P.71-76.

Цель: обнаружение неизвестных экземпляров malware.

Обучающий набор: файлы формата PE32 (Portable executable), ~600 файлов malware, ~90 безопасных.

Выделение признаков (feature extraction):

статические данные, полученные при парсинге файлов:

- байтовые последовательности из секции кода длиной 1 и 2 байта.

Релевантные работы. Пример 2 (2/3)

Выделение значимых признаков (feature selection): на основе вычисления значения функции “информационного усиления” (“information gain”) каждого признака и выбора наиболее значимых.

Использованные классификаторы:

- Naive Bayes, статистический;
- Деревья решений.

Оценка построенных математических моделей детектирования: вычисление значений ошибок первого и второго рода (false negative, false positive), тестовые наборы:

~260 файлов malware, ~40 безопасных файлов.

Релевантные работы. Пример 2 (3/3)

Результаты оценки классификаторов:

Algorithms	Feature Element length	TP	TN	FP	FN	DR (%)	FPR (%)	ACY (%)
Naïve Bayesian	1 Byte	67	10	7	21	76.1	41.2	73.3
	2 Bytes	71	10	7	17	80.7	41.2	77.1
Decision Tree	1 Byte	82	12	5	6	93.2	29.4	89.5
	2 Bytes	83	13	4	5	94.3	23.5	91.4

Релевантные работы. Пример 3 (1/3)

Источник: Zhang B.-Y., Yin J.-P., Hao J.-B., Zhang D.-X., Wang S.-L. Using Support Vector Machine to Detect Unknown Computer Viruses // International Journal of Computational Intelligence Research, Vol.2(1), 2006. P.100-104.

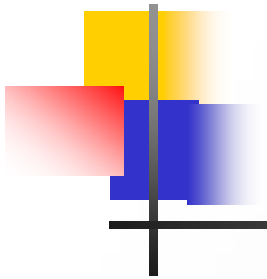
Цель: обнаружение неизвестных экземпляров malware.

Обучающий набор: файлы формата PE32 (Portable executable), 50 файлов malware, 50 безопасных.

Выделение признаков (feature extraction):

поведенческие данные, полученные при выполнении файлов в изолированной вычислительной среде:

- трассы вызовов функций библиотек операционной системы в User Mode.



Релевантные работы. Пример 3 (2/3)

Выделение значимых признаков (feature selection): выделение «аномальных» цепочек вызовов разной длины на основе вычисления расстояния Хэмминга между всеми собранными цепочками для заведомо безопасных приложений и malware.

Использованные классификаторы:

- метод опорных векторов (Support Vector Machines, SVM), относящийся к группе классификаторов, основанных на теории разделимости множеств.

Оценка построенных математических моделей детектирования: вычисление значений ошибок первого и второго рода (false negative, false positive), тестовый набор: 373 файла malware, 159 безопасных.

Релевантные работы. Пример 3 (3/3)

Результаты оценки классификатора:

C	σ^2	False Negative		False Positive	
		k = 6	k = 7	k = 6	k = 7
50	10	3.21%	4.02%	5.66%	7.54%
100	1	4.82%	5.63%	6.28%	5.66%
200	0.5	6.97%	7.50%	10.06%	11.32%

Релевантные работы. Пример 4 (1/3)

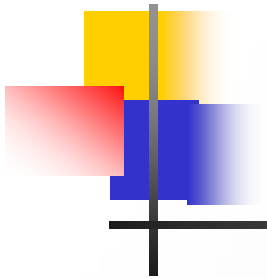
Источник: Kolter J.Z., Maloof M.A. Learning to Detect Malicious Executables in the Wild // Proceedings of the 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2004.

Цель: обнаружение неизвестных экземпляров malware.

Обучающий набор: файлы формата PE32 (Portable executable), 1651 файл malware, 1971 безопасный.

Выделение признаков (feature extraction):

- статические признаки, выделение всех существующих бинарных последовательностей длиной в 4 байта.



Релевантные работы. Пример 4 (2/3)

Выделение значимых признаков (feature selection):

Для каждого признака (4-байтовой последовательности) был вычислен коэффициент информационного усиления. 500 наиболее значимых были использованы для проведения обучения.

Использованные классификаторы:

(Boosted) Naive Bayes, (Boosted) SVM, (Boosted) J48, TFIDF, kNB.

Оценка построенных математических моделей детектирования: вычисление значений ошибок первого и второго рода (false negative, false positive), визуализация ROC-кривой; использовалась 10-кратная cross-проверка на урезанном и полном наборах данных.

Релевантные работы. Пример 4 (3/3)

Результаты оценки классификатора:

Method	AUC
Naive Bayes	0.8850±0.0247
J48	0.9235±0.0204
Boosted Naive Bayes	0.9461±0.0170
TFIDF	0.9666±0.0133
SVM	0.9671±0.0133
IBk, $k = 5$	0.9695±0.0129
Boosted SVM	0.9744±0.0118
Boosted J48	0.9836±0.0095

Method	AUC
Naive Bayes	0.9366±0.0099
J48	0.9712±0.0067
TFIDF	0.9868±0.0045
Boosted Naive Bayes	0.9887±0.0042
IBk, $k = 5$	0.9899±0.0038
Boosted SVM	0.9903±0.0038
SVM	0.9925±0.0033
Boosted J48	0.9958±0.0024

Примечание: левая таблица демонстрирует результаты проверок на урезанном объеме данных (476 malicious / 561 безопасный); правая таблица отображает результаты для проверки на всем объеме данных.

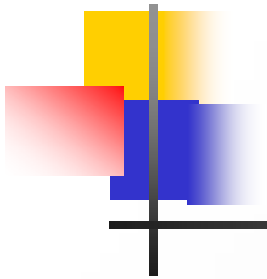
Сущность предлагаемого подхода (1/3)

Предлагаемый подход основан на использовании методов классификации (отнесения объектов к тому или иному классу на базе формируемой математической модели) приложений на пространстве статических (структура и данные файлового контейнера) и поведенческих признаков.

Цель: обнаружение неизвестных экземпляров malware.

Требования:

- точности;
- скрытности;
- соответствия ожиданиям пользователя

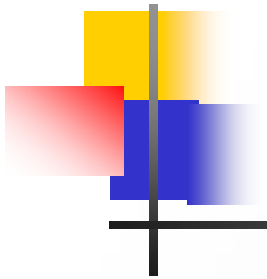


Требование точности

Требование точности: минимальные значения ошибок первого и второго рода при обработке данных о приложениях, не находившихся в обучающем наборе.

Идеальным, но на практике недостижимым, является возможность гарантированного обнаружения известных и неизвестных экземпляров malware с нулевым показателем ложных срабатываний.

Приближенные к практике значения: обнаружение 95% известных и неизвестных экземпляров malware с показателем ложных срабатываний не более 10%.

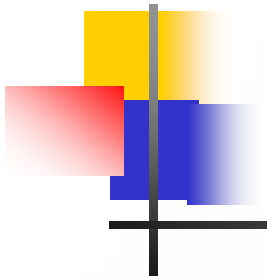


Требование скрытности

Требование скрытности: скрытность мониторинга вследствие наличия разнообразных методов обнаружения мониторинга и противодействия ему запущенными экземплярами malware.

Современные технологии malware позволяют достаточно легко определять факт слежения (использование Debugging API, модификация виртуального адресного пространства процесса посредством методик Byte Stealing, API redirection, etc.).

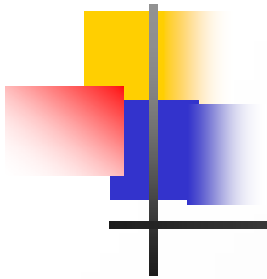
Необходимо минимизировать возможность обнаружения факта слежения за «подозреваемым» приложением.



Требование соответствия ожиданиям пользователя

Требование соответствия ожиданиям пользователя: минимальное влияние реализации подхода на работу пользователя (требования оперативности и устойчивости работы пользователя).

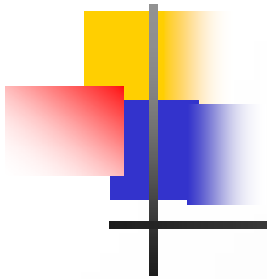
Подход может быть широко применим только в тех случаях, когда он не имеет ярко выраженного негативного эффекта на показатели работы пользователя, приложений, операционной системы.



Сущность предлагаемого подхода (2/3)

Минимальные неделимые информационные единицы при формировании пространства признаков:

- некоторые структурные атрибуты;
- понятие терминального инцидента (события) .



Структурная информация

Использованные структурные атрибуты:

- количество секций (NtHeaders.FileHeader.NumberOfSections);
- расположение EntryPoint - (NtHeaders.OptionalHeader.EntryPoint) - (секция / пространство м-ду окончанием заголовков и первой секцией);
- максимальная энтропия секции;
- названия секций;
- количество импортируемых модулей;
- количество импортируемых функций.

Событийная информация

Описание терминального инцидента (события) включает в себя:

- имя вызванной функции;
- значения переданных на вход функции операндов;
- значения возвращенных функцией результатов.

Сущность предлагаемого подхода (3/4)

Процесс выделения из всего набора структурных атрибутов наиболее существенных (**множества признаков)** строится на базе оценивания степени значения информационного усиления каждого из них и учитывает:

- данные из заголовка DOS HEADER;
- данные из заголовков NT HEADERS (File, Optional);
- данные из заголовков секций (SECTIONS);
- данные из директорий импорта и экспорта.

Сущность предлагаемого подхода (4/4)

Процесс выделения из набора хронологически упорядоченных инцидентов **множества признаков** учитывает:

- количество вызовов каждой функции;
- количество обращений к ресурсам, обладающим одинаковым рангом значимости;
- факты запросов специфических ресурсов (загрузка системных библиотек, попытки обращения к разделам системного реестра, системным файлам и т.д.);
- наличие и количество специфических цепочек вызовов.

Выделение специфических для malware вызовов (feature selection) обеспечивается за счет вычисления *расстояния Левенштейна* (мера различия двух последовательностей символов, определяемая как минимальное количество операций вставки, удаления и замены, необходимых для перевода одной строки в другую).

Задачи исследований

В рамках данного этапа исследования реализуются следующие задачи:

- применимость методов ориентированных на генерирование набора правил классификации (OneR)
- применимость канонического Naive Bayes (NB) классификатора и расширенных классификаторов на базе NB с использованием к качестве дополнительного набора признаков статических характеристик исполняемых файлов;
- применимость индуктивного метода классификации Decision Tree (дерево решений);
- применимость нейронных сетей (многослойный перцептрон).

Комплекс моделирования (1/3)

Используемая платформа:

- Для проведения эксперимента выбрана одна из наиболее популярных как с точки зрения пользователей, так и с точки зрения злоумышленников операционная система Windows XP (NT5.1).
- В целях обеспечения изолированной вычислительной среды и ускорения восстановления используемой операционной системы после отработки вредоносной программы использовался программный пакет виртуализации VmWare Workstation 5.5.3.

Исходные данные:

Обучающий и тестовый наборы приложений были сформированы с использованием файлов заведомо безопасных и вредоносных программ из следующих источников:

- архив вредоносных программ был извлечен с сайта VX Heavens (vx.netlux.org);
- набор безопасных приложений был сформирован из исполняемых файлов, предустанавливаемых с ОС Windows XP и Microsoft Office.

Комплекс моделирования (2/3)

Использованное средство мониторинга

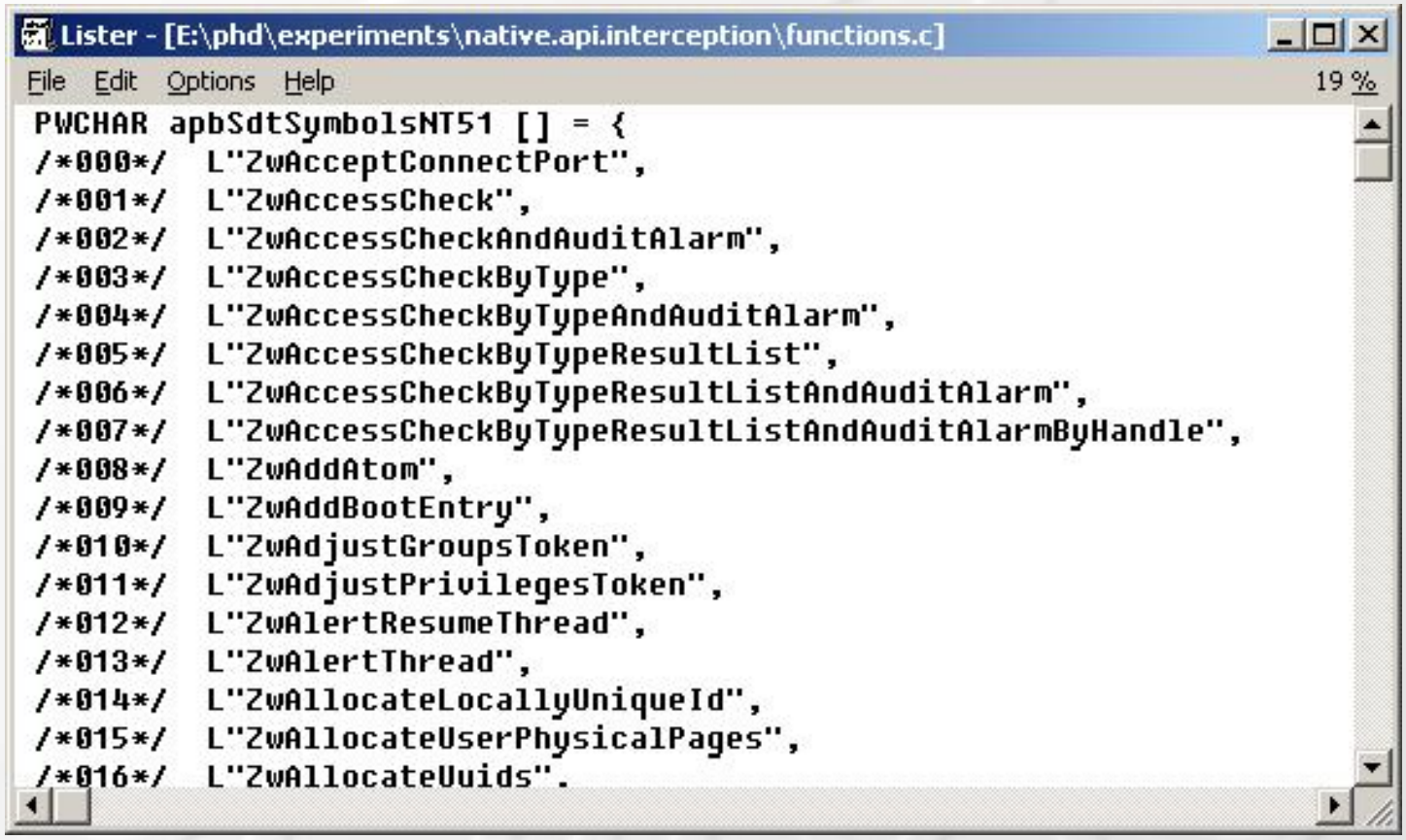
Контроль доступа к ресурсам ОС осуществляется средствами перехватчика вызовов Native API, являющего основным интерфейсом между ядром Windows XP и пользовательским приложением.

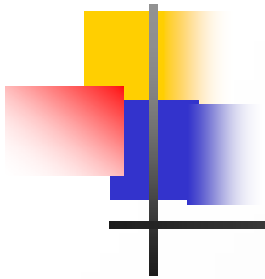
В основу метода перехвата заложены идеи, представленные Марком Руссиновичем и Брюсом Когвеллом и развитые Свеном Шрайбером (Шрайбер С. Недокументированные возможности Windows 2000. Издательство «Питер», 2002.).

Для проведения обучения, кросс-проверок, контрольных проверок и визуализации результатов используется специализированный программный комплекс **Weka Classifier**; входные данные подаются в arff (Attribute-Relation File Format) формате.

Комплекс моделирования (3/3)

Фрагмент перечня функций Native API Windows XP





Основные функции, выбранные для перехвата

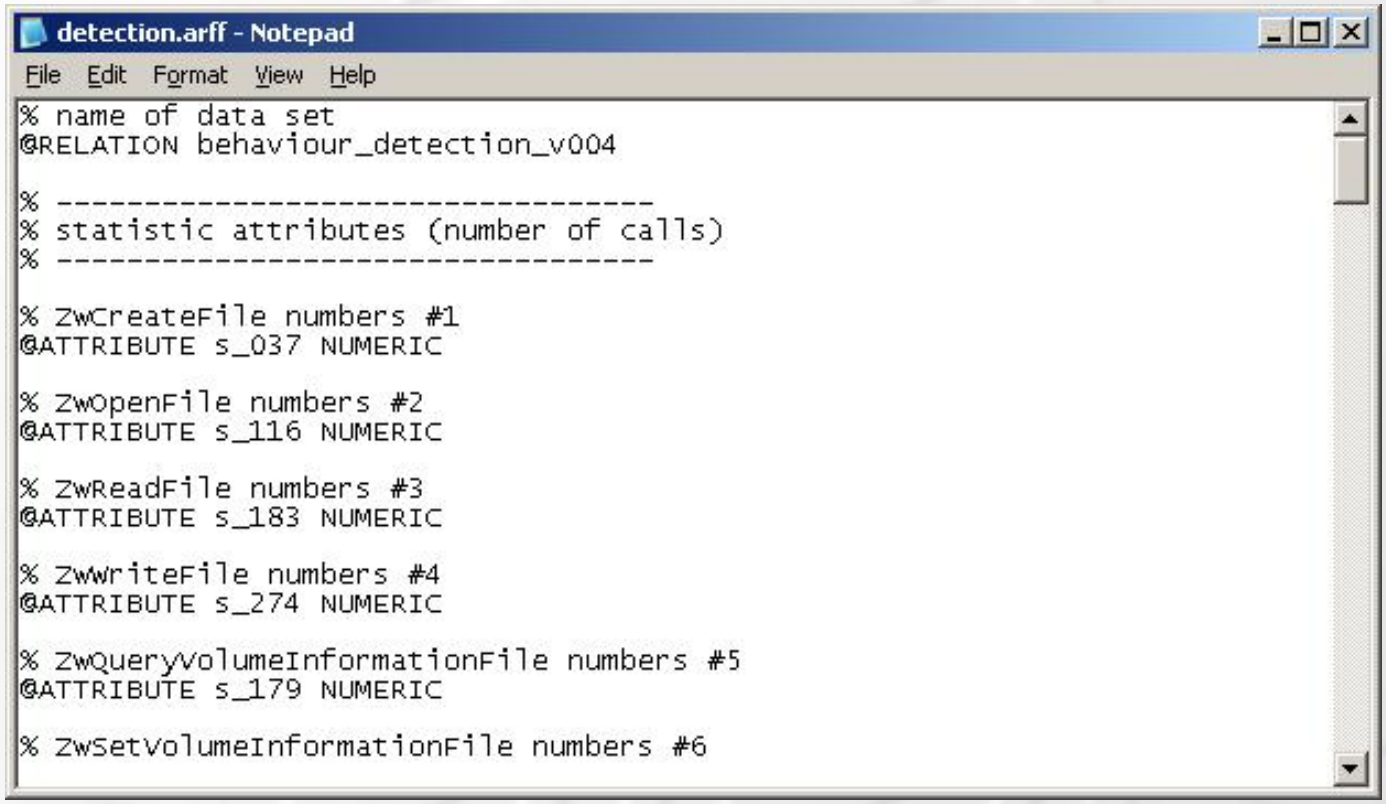
- файловая система (функции получения доступа к файлам, чтения и модификации контента и атрибутов - *File);
- подсистема создания процессов (объектов новых приложений в операционной системе - *Process);
- подсистема поддержки системного реестра (получение доступа к ключам реестра, и их содержимому - *Key).

Id	Function name	Id	Function name
037	ZwCreateFile *	224	ZwSetInformationFile
116	ZwOpenFile *	145	ZwQueryDirectoryFile
183	ZwReadFile *	047	ZwCreateProcess
274	ZwWriteFile *	048	ZwCreateProcessEx
179	ZwQueryVolumeInformationFile	041	ZwCreateKey *
248	ZwSetVolumeInformationFile	063	ZwDeleteKey *
139	ZwQueryAttributesFile	065	ZwDeleteValueKey
149	ZwQueryFullAttributesFile	119	ZwOpenKey
151	ZwQueryInformationFile	247	ZwSetValueKey *

(* - осуществляется контроль операндов)

Данные для обучения классификаторов (1/6)

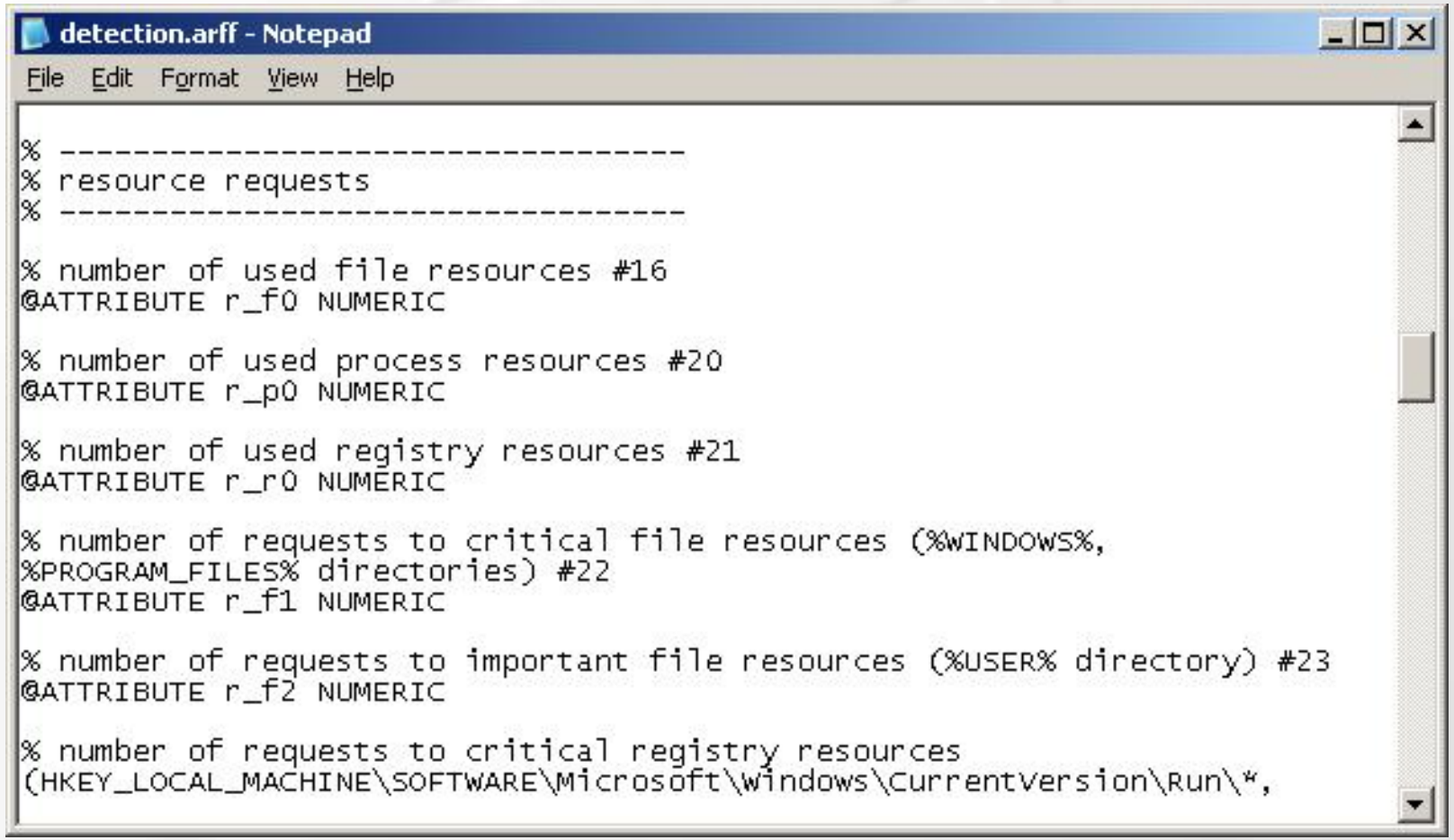
Фрагмент arff-файла о описанием входных данных (**атрибуты количества вызовов контролируемых функций Native API**).



файл ARFF (Attribute-Relation File Format) содержит данные, подготовленные для проведения обучения классификатора.

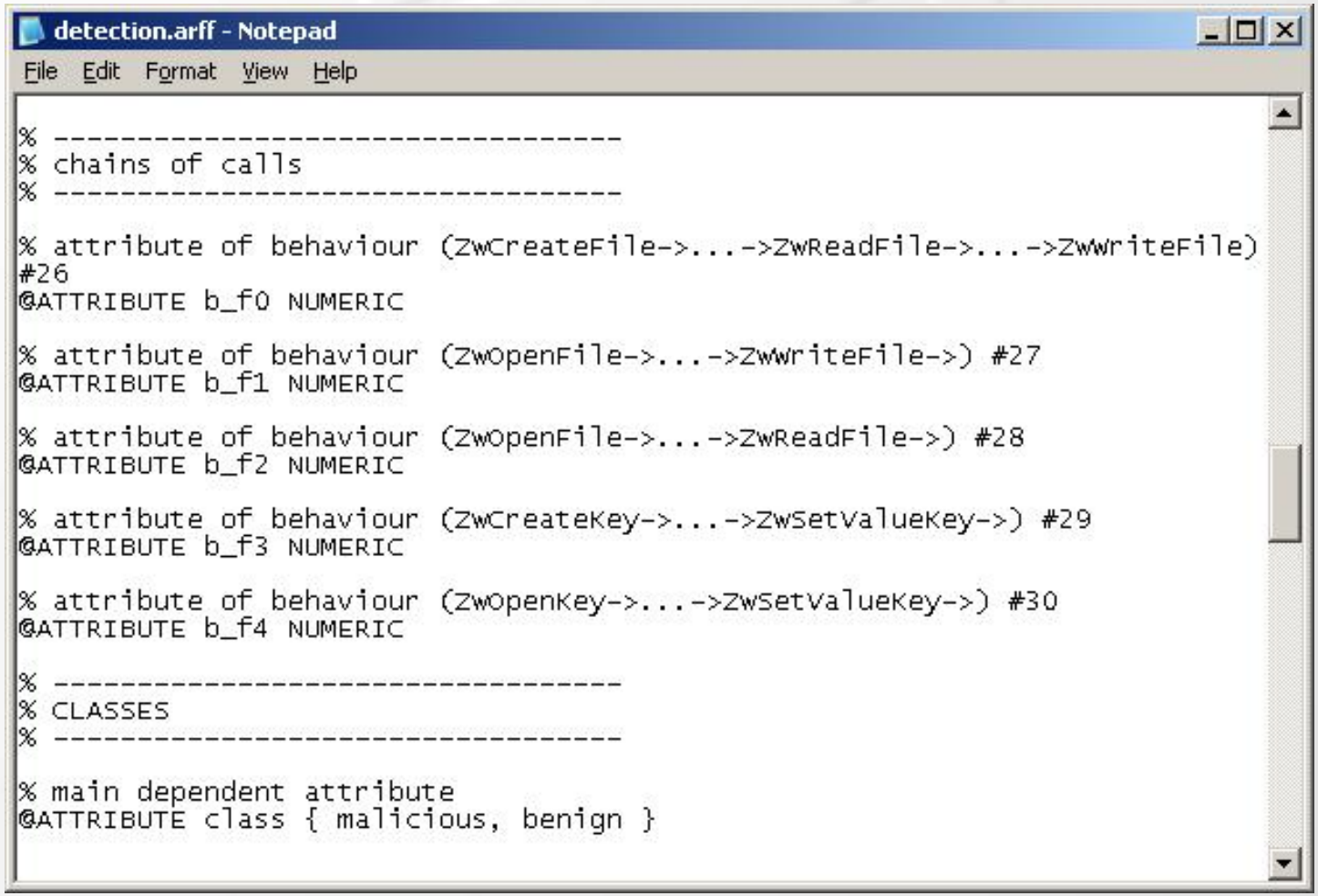
Данные для обучения классификаторов (2/6)

Фрагмент arff-файла о описанием входных данных (**атрибуты обращения к ресурсам**).



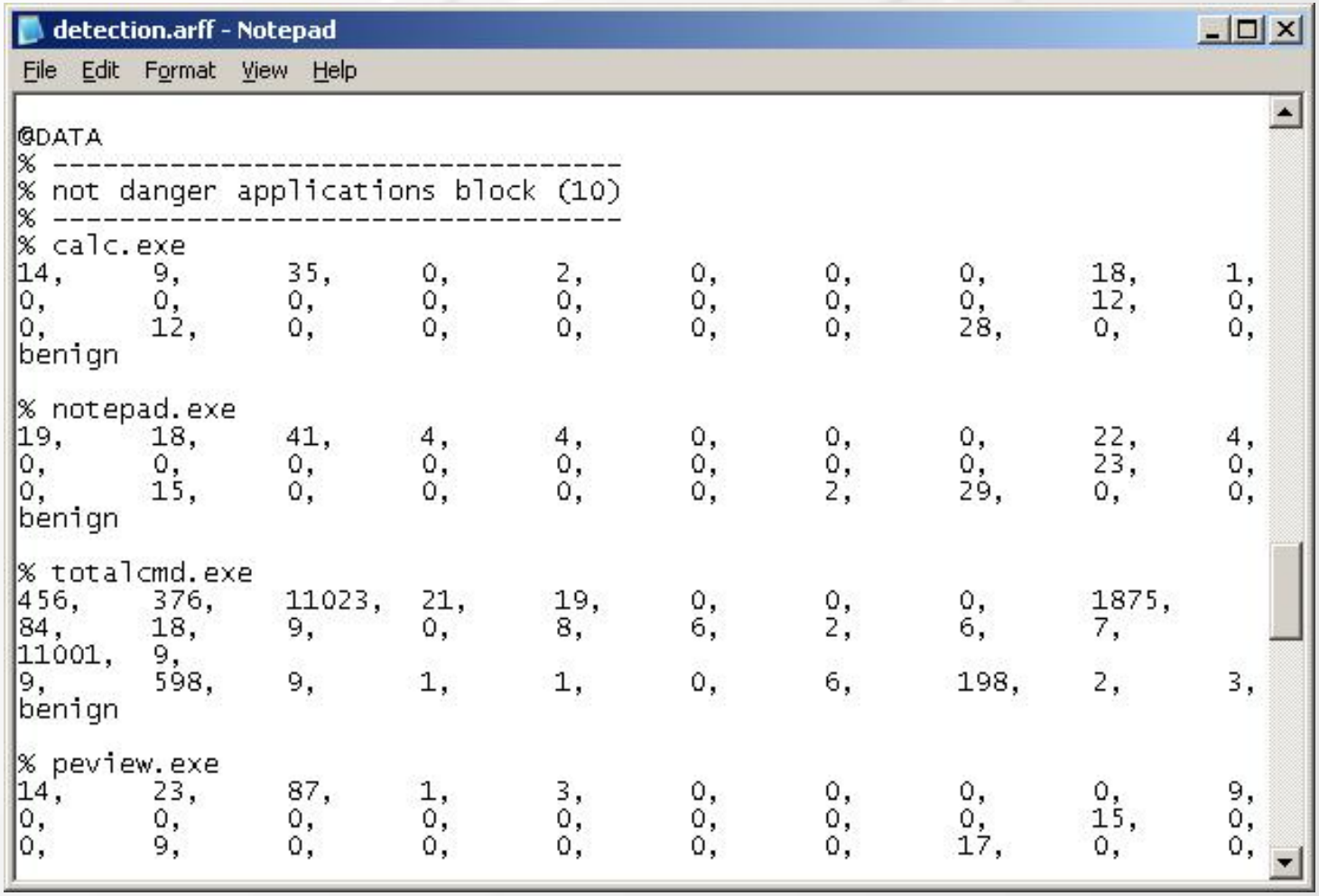
Данные для обучения классификаторов (3/6)

Фрагмент arff-файла с описанием входных данных (**атрибуты цепочек вызовов**).



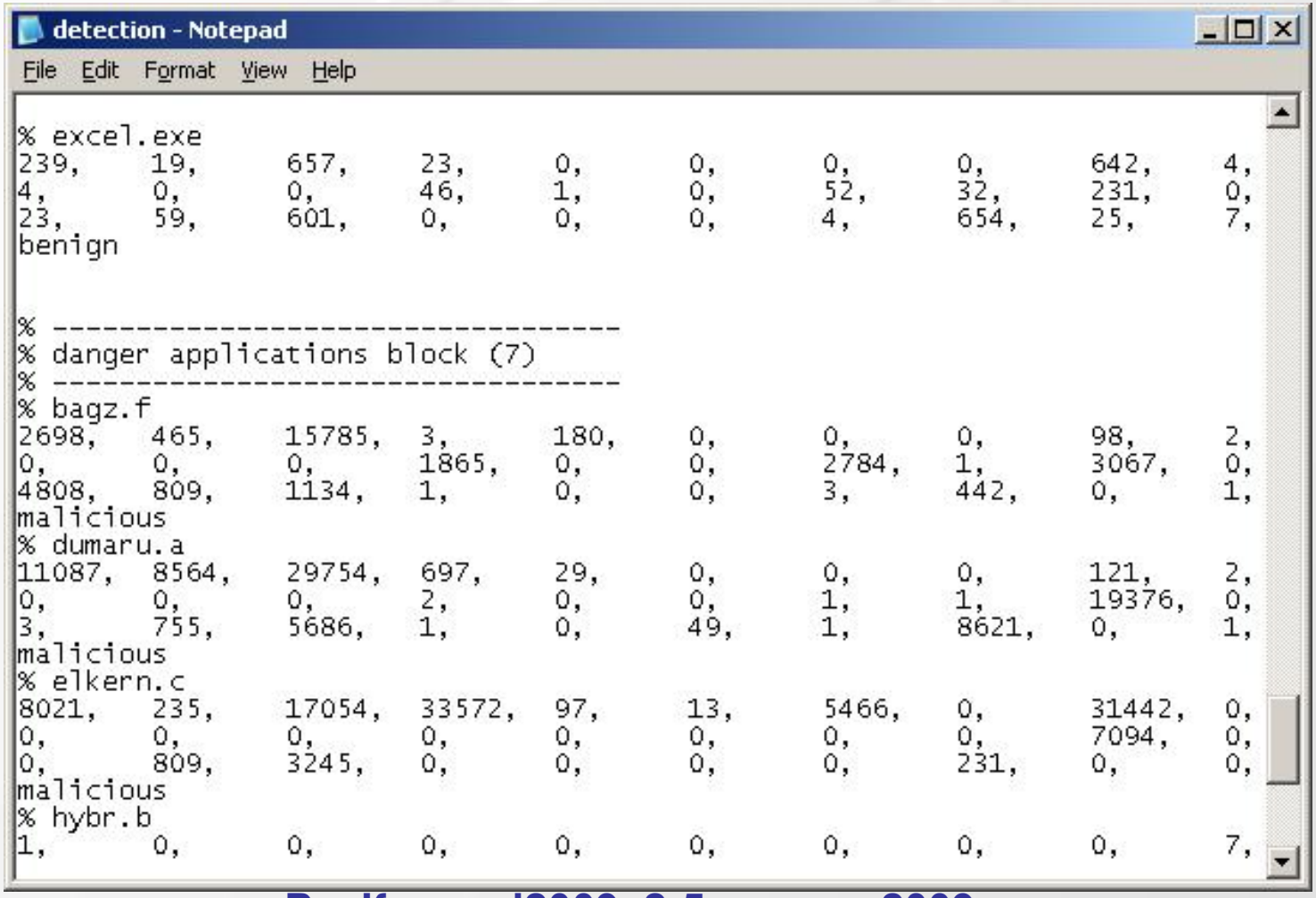
Данные для обучения классификаторов (4/6)

Фрагмент arff-файла с описанием входных данных (**значения атрибутов**).



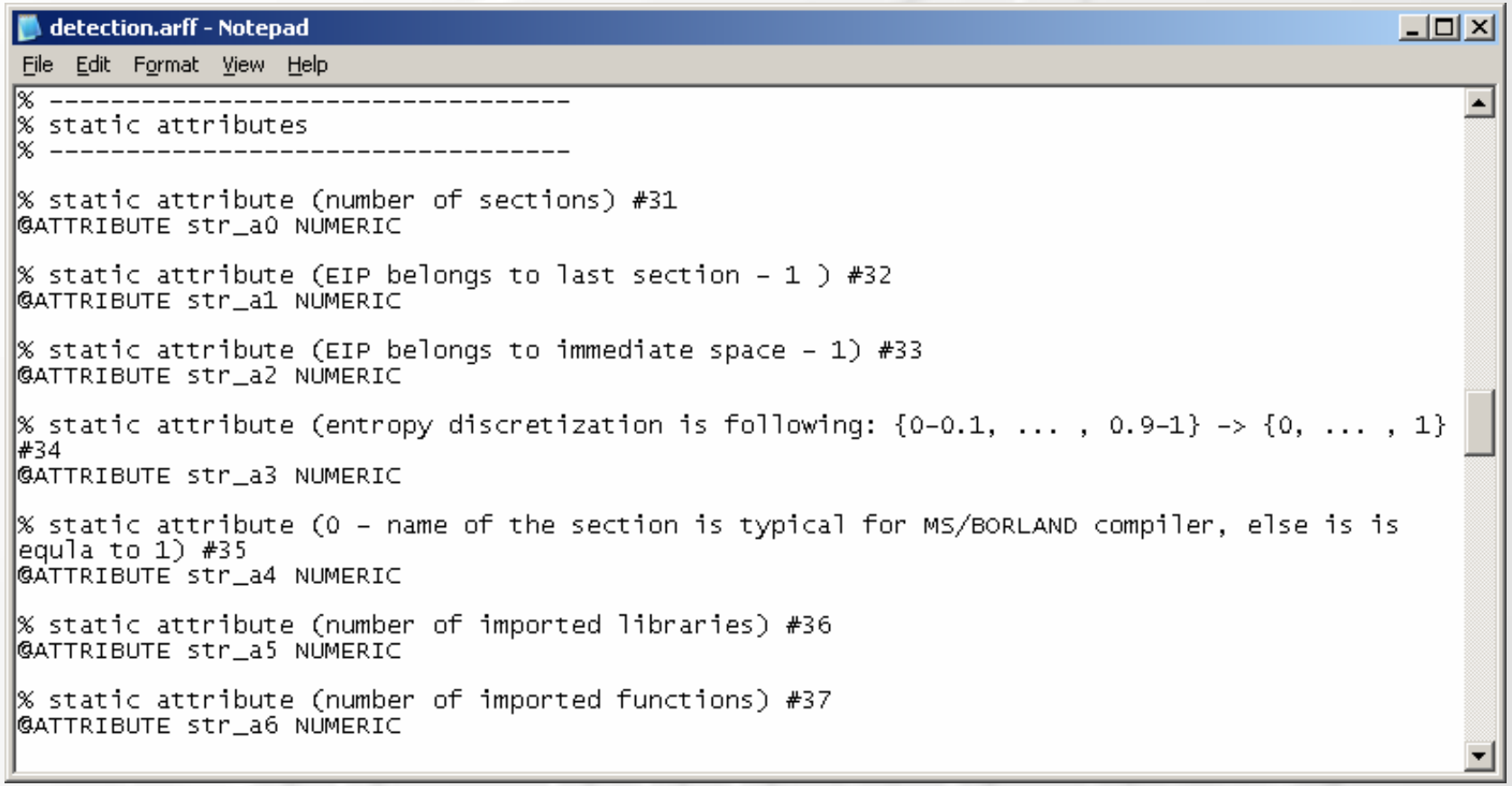
Данные для обучения классификаторов (5/6)

Фрагмент arff-файла с описанием входных данных (**значения атрибутов**).



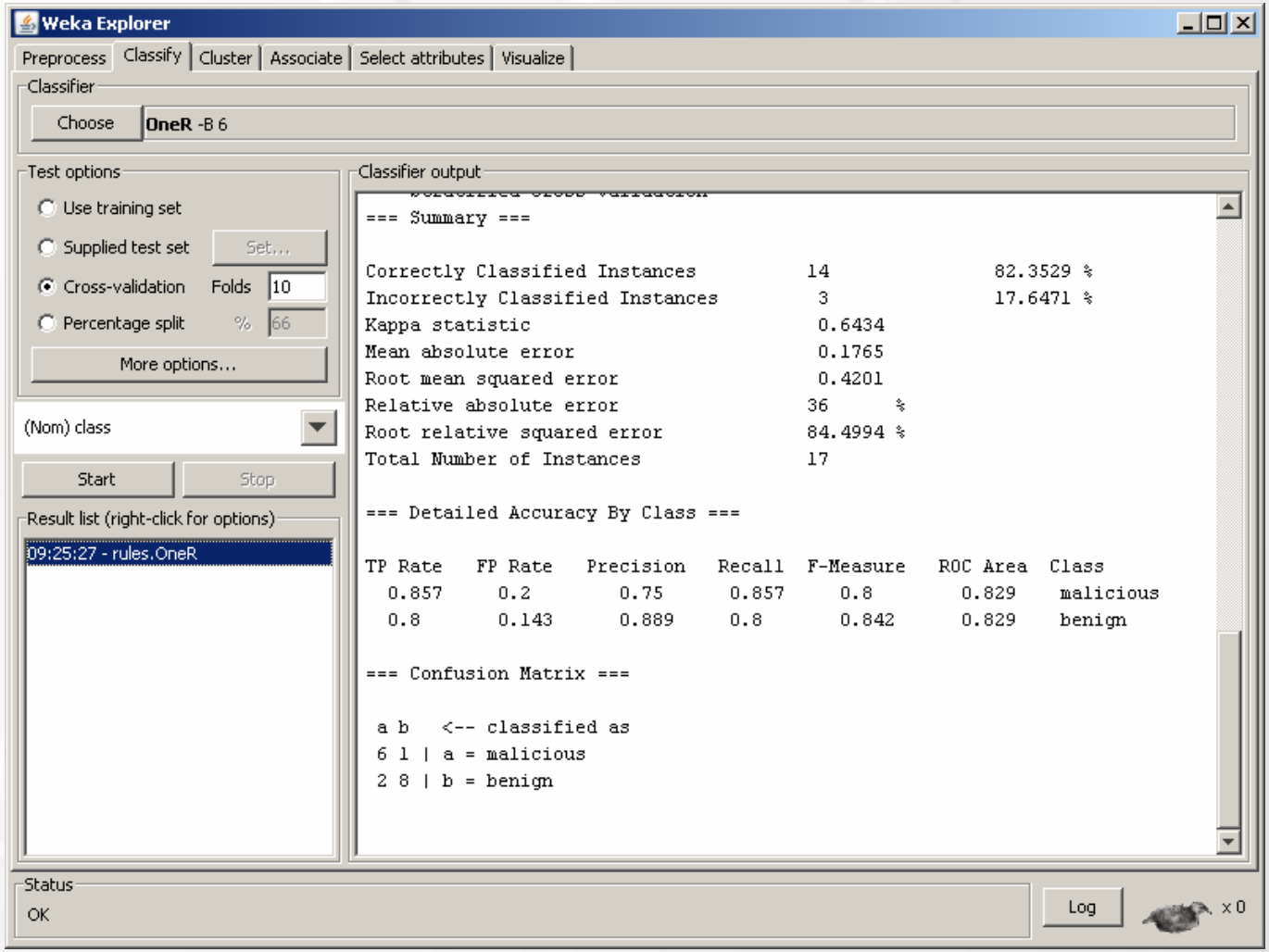
Данные для обучения классификаторов (6/6)

Фрагмент arff-файла с описанием входных данных (**значения атрибутов**).



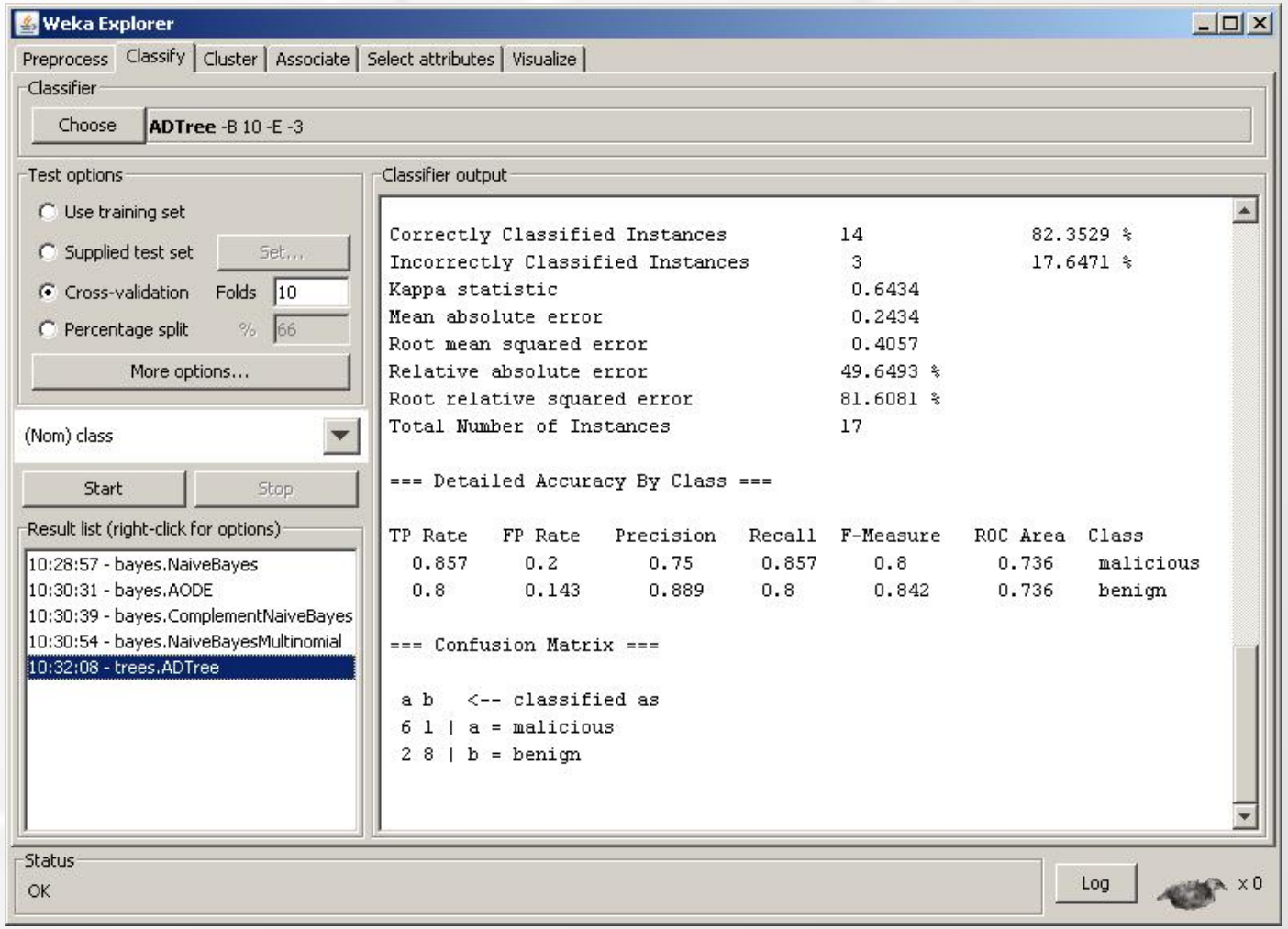
Результаты экспериментов (1/7)

One-R



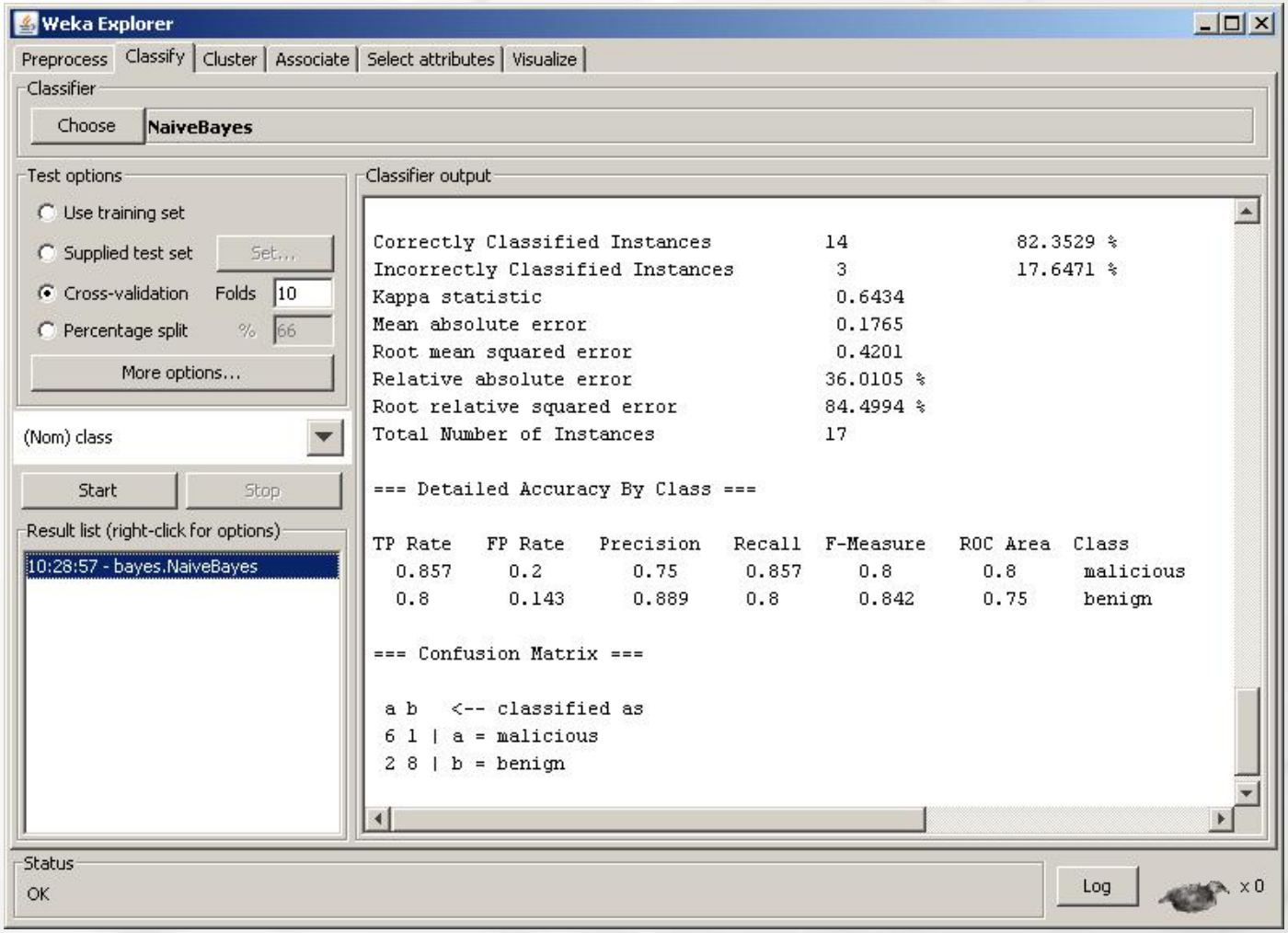
Результаты экспериментов (2/7)

Decision Tree



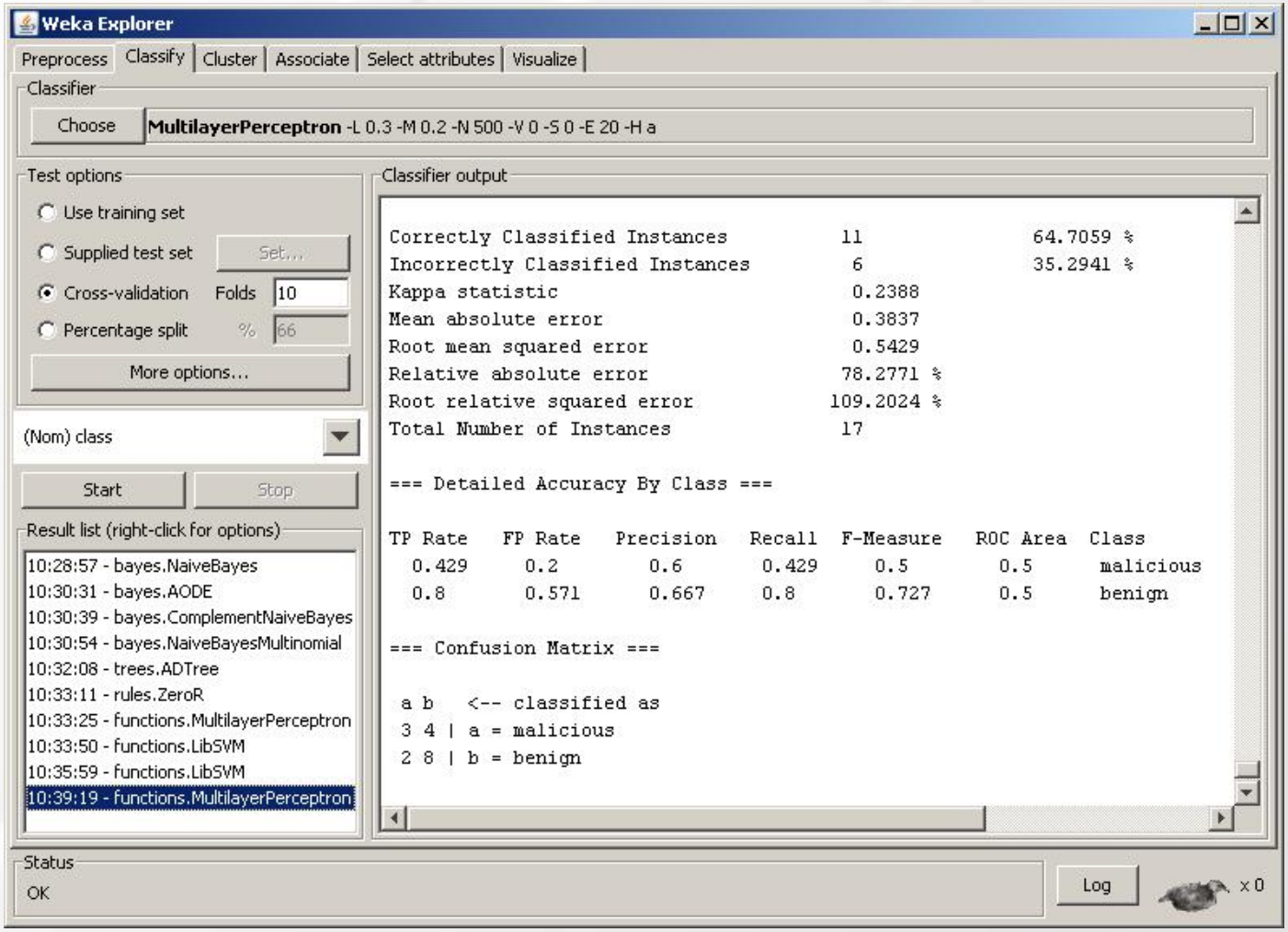
Результаты экспериментов (3/7)

Naive Bayes



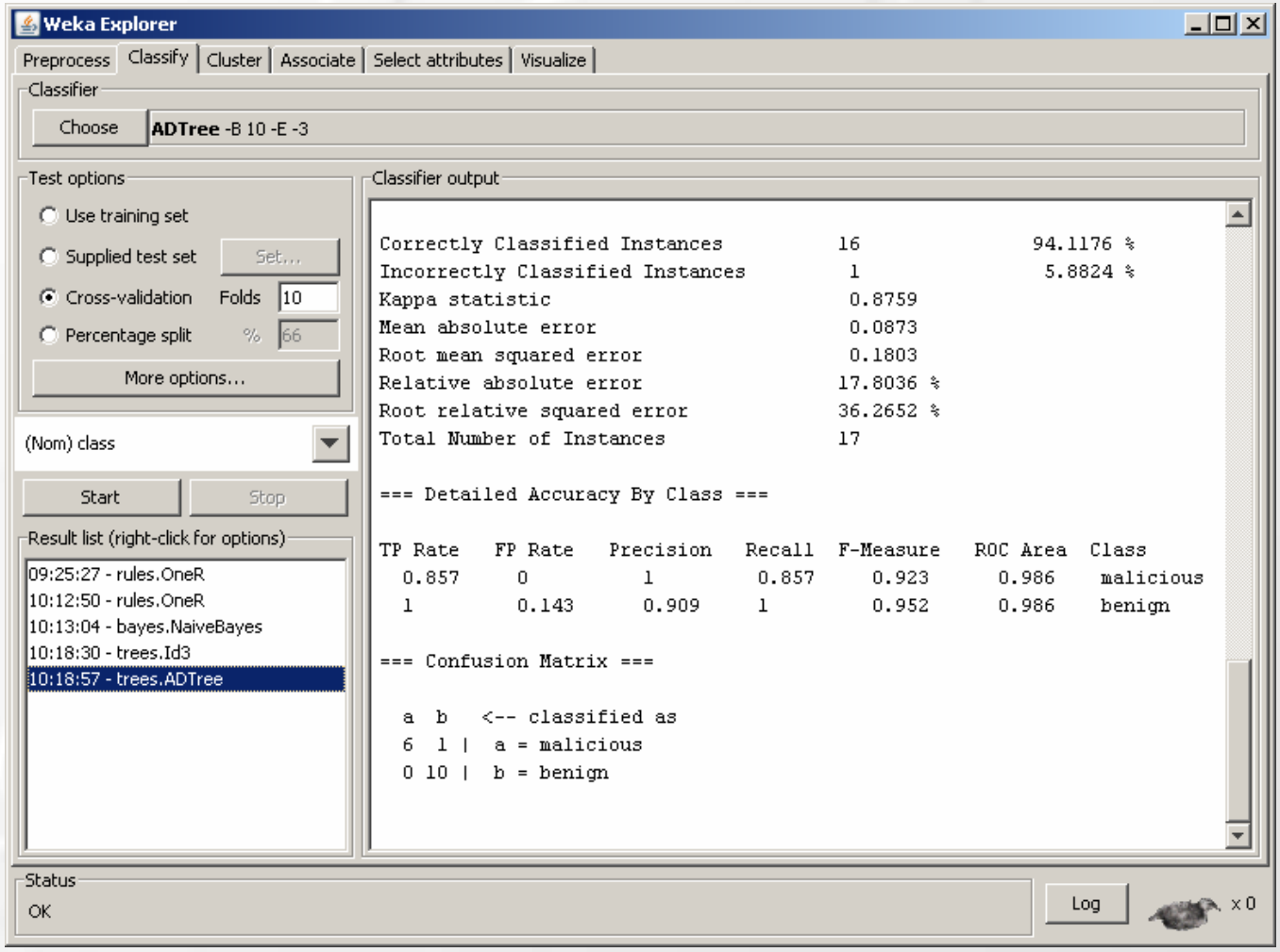
Результаты экспериментов (4/7)

Multilayer Perceptron (NeuralNets)



Результаты экспериментов (5/7)

Decision Tree (с включением статических признаков)



Результаты экспериментов (6/7)

На текущий момент, являющийся по сути стартовым для данной работы, в среднем точность детектирования (True positive rate) malware образцов составляет **80%** при показателе ложных срабатываний (False positive rate) в пределах **15%.**

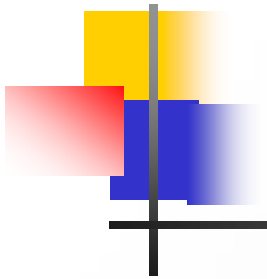
При этом классификаторы, использующие вероятностный (Naive Bayes), индуктивный (DTree) и Rule-ориентированный (OneR) подходы значительно превосходят классификаторы, построенные на базе принципов разделимости множеств (SVM, NeuralNets).

К наиболее значащим признакам, обеспечивающим текущую точность классификации, относятся те, которые описывают интенсивность файлового перебора (на фазе подготовки к репликации) и использование механизмов автозапуска (системный реестр).

Результаты экспериментов (7/7)

Данные по статическим аттрибутам в целом позволяют с большей степенью вероятности находить вредоносные программы (см. «Результаты экспериментов», слайд 5/7). Однако это достигается за счет того, что, как оказалось, большая часть вредоносных программ из использованных обучающей и тестовой коллекций защищена средствами, влияющими на структурные характеристики файлов-контейнеров malware.

Следует отметить, что **набор безопасных приложений**, использованных для обучения классификаторов, **этими свойствами в такой мере не обладал** (был упакован только один тестовый файл – totalcmd.exe)..



Заключение

В работе:

- проведен анализ задачи использования методов Data Mining для обнаружения вредоносного ПО (malware);
- предложен подход к комбинированному обнаружению вредоносного ПО, учитывающий как статические так и поведенческие аспекты;
- разработан комплекс моделирования и проведены эксперименты.

Указанный подход позволяет осуществлять обнаружение вредоносного программного обеспечения во время его выполнения за счет классификации по структурным (статическим) признакам и по выделенным признакам его поведения.

Вместе с тем, текущие результаты экспериментов показывают необходимость в дальнейшем более тщательного выделения наборов поведенческих признаков, специфичных для каждого класса вредоносного ПО, расширения программного комплекса моделирования и продолжения совершенствования наборов поведенческих признаков за счет учета дополнительных данных.

Контактная информация

Комашинский Дмитрий Владимирович (ЗАО «Аркадия»)
komashinskiy@comsec.spb.ru
http://comsec.spb.ru/Komashinskiy/

Котенко Игорь Витальевич (СПИИРАН)
ivkote@comsec.spb.ru
http://comsec.spb.ru/Kotenko/

Шоров Андрей Владимирович (СПИИРАН)
ashorov@comsec.spb.ru
http://comsec.spb.ru/Shorov/

Благодарности

Работа выполнена при финансовой поддержке РФФИ (проект №07-01-00547), программы фундаментальных исследований ОНИТ РАН и при частичной финансовой поддержке, осуществляемой в рамках проекта Евросоюза RE-TRUST (контракт № 021186-2).

РусКрипто'2009, 2-5 апреля 2009 г.