# Введение

С начала 2000-х годов и по сегодняшний день одним из ключевых инструментов киберпреступности являются ботнеты[]. Ботнетом называется сеть зараженных узлов, на которых запущен автономный процесс, выполняющий команды злоумышленника. Узел, на котором запущен такой процесс, принято называть ботом или узлом-зомби.

В настоящей работе рассматривается проблема обнаружения и фильтрации ботнетов на этапе их распространения, эффективная для борьбы с ботнетами в виду своей превентивности. В тот момент, когда осуществляется стадия вредоносной активности ботнета, ущерб атакуемым узлам уже нанесен либо полностью, либо частично. В этом случае задача обнаружения и фильтрации ботнетов сводится к борьбе с последствиями вредоносной активности. Превентивное обнаружение позволит минимизировать наносимый ботнетом ущерб.

В данной рассматривается проблема обнаружения и фильтрации ботнетов, распространяющихся посредством удаленного эксплуатирования уязвимостей работы с памятью. Уязвимости работы с памятью возникают тогда, когда некоторый код в программе записывает в память больше данных, чем было предусмотрено разработчиком приложения. Типичными примерами таких уязвимостей являются переполнение стека, переполнение кучи, а так же некоторых других служебных структур.

Вредоносный код, эксплуатирующий уязвимости работы с памятью, традиционно называется шелл кодом (shellcode).

Возможный пример применения решения поставленной проблемы- монитор в рамках IDS/IPS. Каждый пакет, проходящий по каналу, анализируется на предмет содержания в нем шелл кодов. Если шелл код в пакете обнаружен, то пакет сбрасывается. Это, в свою очередь, накладывает определенные ограничения на решение рассматриваемой проблемы. В виду того, что высокоскоростные каналы характеризуются большим объемом передаваемой информации, проходящей по каналу в единицу времени, алгоритмическая сложность решения крайне критична. В случае высокой алгоритмической сложности возможно два сценария. При первом проходящие по каналу данные будут обработаны частично, что не дает гарантий обнаружения ботнета в обработанном объеме данных, даже при условии его передачи по каналу. Обработка же всего объема передаваемых данных влечет за собой уменьшение пропускной способности канала до значения пропускной способности средства обнаружения распространения ботнетов, либо приводит к запозданию решения о вредоносности данных. Последнее означает, что даже если в проходящем трафике были обнаружены шелл коды, то данные уже были переданы далее, а ущерб, возможно, уже нанесен.

Кроме того, для средства обнаружения и фильтрации шелл кодов крайне важен показатель ложных срабатываний. При больших объемах анализируемых данных, даже незначительный процент ложных срабатываний метода влечет за собой значительное абсолютное значение ложных сигналов, что может сказаться на качестве обслуживания легитимных пользователей.

# Аналитический обзор вредоносных кодов

Жизненный цикл любого ботнета включает в себя несколько стадий[]:

* *Разработка*

Планирование архитектуры ботнета и его реализация, выбирается механизм распространения ботнета и конкретная уязвимость, которая будет эксплуатироваться при распространении.

* *Распространение*

Осуществляется эксплуатирование уязвимостей узлов и внедрения на них ботов.

* *Взаимодействие*

Взаимодействие между ботами и их мастером (в случае его наличия в архитектуре ботнета).

* *Вредоносная активность*

Осуществление вредоносной активности.

* *Успешное завершение атаки*

Типичная структура шелл кода:

1. *Активатор*

Участок тела шелл кода, гарантирующий корректное выполнение полезной нагрузки шелл кода

1. *Декриптор*

Часть тела шелл кода, необходимая для расшифровки зашифрованной части шелл кода, в случае ее присутствия.

1. *Полезная нагрузка*

Участок кода, выполняющий вредоносную активность

1. *Зона адресов возврата*

Зона адресов возврата содержит предполагаемый злоумышленником абсолютный адрес.

# Методы обнаружения вирусов

Как правило, методы обнаружения делят на две группы: обнаружение конкретных вредоносных кодов и универсальное обнаружение. Однако обнаружение конкретных вредоносных кодов на практике редко реализуем, т.к. существует огромное число модификаций одного и того же вируса. Вторая группа методов - универсальное обнаружение - позволяет обнаружить вредоносный код независимо от его типа и она более востребована на практике.

Универсальный подход очень важен ввиду его гибкости и способности быстро приспосабливаться к новым или полностью неизвестным вредоносным программам. Основным преимуществом универсальных методов обнаружения является возможность простой адаптации, основанной на переобучении классификатора на новые вредоносные коды и форматы контейнеров без необходимости разработки их явных моделей.

Универсальное обнаружение является методом мета-обнаружения, т.е. это метод, который может быть настроен (после обучения на исходных и модифицированных изображениях) на обнаружение любого вредоносного кода. Цель универсального обнаружения состоит в классификации заданных кодов в два класса: пустые (данные, не несущие вред) контейнеры и модифицированные (вредоносные коды).

Задача универсального обнаружения может быть рассмотрена как задача статистического распознавания некоторых признаков вредоносных кодов. Этот подход основан на использовании статистических методов классификации с обучением. Основная идея применения этих методов в обнаружении вредоносных кодов состоит в следующем.

На основании обучающей выборки, представляющей собой наблюдения из двух классов: класса пустых контейнеров и класса модифицированных контейнеров формируются признаки классификации, которые позволяют отличить пустой контейнер от модифицированного и строятся статистические решающие правила, которые в дальнейшем применяются для классификации вновь поступивших контейнеров в один из рассматриваемых классов.

Можно предложить общую схему универсального стеганоанализа, которая состоит из следующих основных частей:

* Подбор пустых и модифицированных контейнеров в достаточном для анализа количестве.
* Выделение признаков.
* Выбор классификатора.
* Классификация.

# Признаки вредоносного кода

Некоторые признаки вредоносных кодов[]:

* *Наличие GetPC кода.*

GetPC - набор исполнимых инструкций, вычисляющих свое рас-

положение в адресном пространстве исполнимого процесса. GetPC код, как правило, необходим, чтобы возможно было подменить значение программного указателя на адрес самого кода непосредственно. Этот признак специфичен для вредоносных исполнимых инструкций, использующих техники самодекодирования и самомодификации.

* *Адрес возврата находится в определенном диапазоне значений.*

В шелл коде адрес возврата перезаписывается значением, которое находится в диапазоне адресного пространства исполнимого процесса.

* *Использование шаблонов.*

В шелл кодах часто используются зарезервированные ключевые слова или априорно известные числовые константы. Такие признаки являются специфичными для вредоносных объектов, использующих конкретные уязвимости[].

* *Если последняя инструкция в цепочке заканчивается командой перехода с прямой или абсолютной адресацией.*

Как правило, при эксплуатировании уязвимости задачей злоумышленника является не аварийное завершение запущенного процесса, а например, получения контроля над консолью с уровнем доступа ядра или какой-либо другой цели. Таким образом, код вредоносного объекта должен передавать управление системным вызовам, которые могут быть доступны явным вызовом библиотеки или путем прямого прерывания.

# Математическая модель обнаружения на основе дискриминантного анализа

Рассмотрим математическую модель обнаружения на основе дискриминантного анализа[]. Пусть  – обучающая выборка, состоящая из пустых контейнеров (класс ), c распределением , - обучающая выборка, состоящая из модифицированных контейнеров (класс ), с p-мерным Гауссовским распределением , с векторами математических ожиданий и и ковариационными матрицами

, p – число признаков. Для модели линейного дискриминантного анализа предполагается, что ковариационные матрицы равны между собой: .

Линейная дискриминантная функция для j-го класса записывается в виде:



где - априорная вероятность того, что контейнер x принадлежит классу .

Параметры дискриминантной функции  и константы  выражаются через вектора математических ожиданий  и ковариационную матрицу .

Решающее правило классификации в случае модели линейного дискриминантного анализа состоит в следующем: контейнер относится к классу , если выполняется условие; в противном случае x относится к классу .

Отметим, что эффективность построенных решающих правил дискриминантного анализа, определяется путем переклассификации обучающей выборки или путем классификации экзаменационной выборки.

Число параметров линейной дискриминантной функции определяется следующим выражением:



Между числом параметров дискриминантной функции *l* и объемом обучающей выборки *n* должно выполняться соотношение. В противном случае вероятность ошибочной классификации при использовании дискриминантных функций существенно возрастет.

|  |  |  |  |
| --- | --- | --- | --- |
| Матрица классификации (table.sta) Строки: наблюдаемые классы Столбцы: предсказанные классы | | | |
|  | **Процент** | **G\_1:1** | **G\_2:2** |
| **G\_1:1** | 100.0000 | 54 | 0 |
| **G\_2:2** | 100.0000 | 0 | 52 |
| **Всего** | 100.0000 | 54 | 52 |

|  |  |  |  |
| --- | --- | --- | --- |
| Матрица классификации (table.sta) Строки: наблюдаемые классы Столбцы: предсказанные классы | | | |
|  | **Процент** | **G\_1:1** | **G\_2:2** |
| **G\_1:1** | 100.0000 | 40 | 0 |
| **G\_2:2** | 100.0000 | 0 | 40 |
| **Всего** | 100.0000 | 40 | 40 |

|  |  |  |  |
| --- | --- | --- | --- |
| Апостериорные вероятности (table.sta) Неправильные классификации отмечены \* | | | |
|  | **Наблюд.** | **G\_1:1** | **G\_2:2** |
| **1** | G\_1:1 | 1.000000 | 0.000000 |
| **2** | G\_1:1 | 1.000000 | 0.000000 |
| **3** | G\_1:1 | 1.000000 | 0.000000 |
| **4** | G\_1:1 | 1.000000 | 0.000000 |
| **5** | G\_1:1 | 1.000000 | 0.000000 |
| **6** | G\_1:1 | 1.000000 | 0.000000 |
| **7** | G\_1:1 | 1.000000 | 0.000000 |
| **8** | G\_1:1 | 1.000000 | 0.000000 |
| **9** | G\_1:1 | 1.000000 | 0.000000 |
| **10** | G\_1:1 | 1.000000 | 0.000000 |
| **11** | G\_1:1 | 1.000000 | 0.000000 |
| **12** | G\_1:1 | 1.000000 | 0.000000 |
| **13** | G\_1:1 | 1.000000 | 0.000000 |
| **14** | G\_1:1 | 1.000000 | 0.000000 |
| **15** | G\_1:1 | 1.000000 | 0.000000 |
| **16** | G\_1:1 | 1.000000 | 0.000000 |
| **17** | G\_1:1 | 1.000000 | 0.000000 |
| **18** | G\_1:1 | 1.000000 | 0.000000 |
| **19** | G\_1:1 | 1.000000 | 0.000000 |
| **20** | G\_1:1 | 1.000000 | 0.000000 |
| **21** | G\_1:1 | 1.000000 | 0.000000 |
| **22** | G\_1:1 | 1.000000 | 0.000000 |
| **23** | G\_1:1 | 1.000000 | 0.000000 |
| **24** | G\_1:1 | 1.000000 | 0.000000 |
| **25** | G\_1:1 | 1.000000 | 0.000000 |
| **26** | G\_1:1 | 1.000000 | 0.000000 |
| **27** | G\_1:1 | 1.000000 | 0.000000 |
| **28** | G\_1:1 | 1.000000 | 0.000000 |
| **29** | G\_1:1 | 1.000000 | 0.000000 |
| **30** | G\_1:1 | 1.000000 | 0.000000 |
| **31** | G\_1:1 | 1.000000 | 0.000000 |
| **32** | G\_1:1 | 1.000000 | 0.000000 |
| **33** | G\_1:1 | 1.000000 | 0.000000 |
| **34** | G\_1:1 | 1.000000 | 0.000000 |
| **35** | G\_1:1 | 1.000000 | 0.000000 |
| **36** | G\_1:1 | 1.000000 | 0.000000 |
| **37** | G\_1:1 | 1.000000 | 0.000000 |
| **38** | G\_1:1 | 1.000000 | 0.000000 |
| **39** | G\_1:1 | 1.000000 | 0.000000 |
| **40** | G\_1:1 | 1.000000 | 0.000000 |
| **41** | --- | 1.000000 | 0.000000 |
| **42** | --- | 1.000000 | 0.000000 |
| **43** | --- | 1.000000 | 0.000000 |
| **44** | --- | 1.000000 | 0.000000 |
| **45** | --- | 1.000000 | 0.000000 |
| **46** | --- | 1.000000 | 0.000000 |
| **47** | --- | 1.000000 | 0.000000 |
| **48** | --- | 1.000000 | 0.000000 |
| **49** | --- | 1.000000 | 0.000000 |
| **50** | --- | 1.000000 | 0.000000 |
| **51** | --- | 1.000000 | 0.000000 |
| **52** | --- | 1.000000 | 0.000000 |
| **53** | --- | 1.000000 | 0.000000 |
| **54** | --- | 1.000000 | 0.000000 |
| **55** | G\_2:2 | 0.000000 | 1.000000 |
| **56** | G\_2:2 | 0.000000 | 1.000000 |
| **57** | G\_2:2 | 0.000000 | 1.000000 |
| **58** | G\_2:2 | 0.000000 | 1.000000 |
| **59** | G\_2:2 | 0.000000 | 1.000000 |
| **60** | G\_2:2 | 0.000000 | 1.000000 |
| **61** | G\_2:2 | 0.000000 | 1.000000 |
| **62** | G\_2:2 | 0.000000 | 1.000000 |
| **63** | G\_2:2 | 0.000000 | 1.000000 |
| **64** | G\_2:2 | 0.000000 | 1.000000 |
| **65** | G\_2:2 | 0.000000 | 1.000000 |
| **66** | G\_2:2 | 0.000000 | 1.000000 |
| **67** | G\_2:2 | 0.000000 | 1.000000 |
| **68** | G\_2:2 | 0.000000 | 1.000000 |
| **69** | G\_2:2 | 0.000000 | 1.000000 |
| **70** | G\_2:2 | 0.000000 | 1.000000 |
| **71** | G\_2:2 | 0.000000 | 1.000000 |
| **72** | G\_2:2 | 0.000000 | 1.000000 |
| **73** | G\_2:2 | 0.000000 | 1.000000 |
| **74** | G\_2:2 | 0.000000 | 1.000000 |
| **75** | G\_2:2 | 0.000000 | 1.000000 |
| **76** | G\_2:2 | 0.000000 | 1.000000 |
| **77** | G\_2:2 | 0.000000 | 1.000000 |
| **78** | G\_2:2 | 0.000000 | 1.000000 |
| **79** | G\_2:2 | 0.000000 | 1.000000 |
| **80** | G\_2:2 | 0.000000 | 1.000000 |
| **81** | G\_2:2 | 0.000000 | 1.000000 |
| **82** | G\_2:2 | 0.000000 | 1.000000 |
| **83** | G\_2:2 | 0.000000 | 1.000000 |
| **84** | G\_2:2 | 0.000000 | 1.000000 |
| **85** | G\_2:2 | 0.000000 | 1.000000 |
| **86** | G\_2:2 | 0.000000 | 1.000000 |
| **87** | G\_2:2 | 0.000000 | 1.000000 |
| **88** | G\_2:2 | 0.000000 | 1.000000 |
| **89** | G\_2:2 | 0.000000 | 1.000000 |
| **90** | G\_2:2 | 0.000000 | 1.000000 |
| **91** | G\_2:2 | 0.000000 | 1.000000 |
| **92** | G\_2:2 | 0.000000 | 1.000000 |
| **93** | G\_2:2 | 0.000000 | 1.000000 |
| **94** | G\_2:2 | 0.000000 | 1.000000 |
| **95** | --- | 0.000000 | 1.000000 |
| **96** | --- | 0.000000 | 1.000000 |
| **97** | --- | 0.000000 | 1.000000 |
| **98** | --- | 0.000000 | 1.000000 |
| **99** | --- | 0.000000 | 1.000000 |
| **100** | --- | 0.000000 | 1.000000 |
| **101** | --- | 0.000000 | 1.000000 |
| **102** | --- | 0.000000 | 1.000000 |
| **103** | --- | 0.000000 | 1.000000 |
| **104** | --- | 0.000000 | 1.000000 |
| **105** | --- | 0.000000 | 1.000000 |
| **106** | --- | 0.000000 | 1.000000 |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| T-критерии; Группир.: Var1 (table.sta) Группа 1:1 Группа 2:2 | | | | | | | | |
|  | **Среднее** | **Среднее** | **t-знач.** | **ст.св.** | **p** | **N набл.** | **N набл.** |
| **Var5** | 0.007614 | 0.003972 | 27.5642 | 78 | 0.000000 | 40 | 40 |
| **Var8** | 0.004265 | 0.002035 | 12.8273 | 78 | 0.000000 | 40 | 40 |
| **Var17** | 0.002043 | 0.014213 | -11.9902 | 78 | 0.000000 | 40 | 40 |