

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ**

Федеральное государственное автономное образовательное учреждение

высшего образования

**«Дальневосточный федеральный университет»**

|  |  |
| --- | --- |
|  | **ШКОЛА ЕСТЕСТВЕННЫХ НАУК** |

**Кафедра информационной безопасности**

Пузан Сергей Александрович

|  |
| --- |
| РАЗРАБОТКА ПЛАТФОРМЫ МОНИТОРИНГА И СКАНИРОВАНИЯ ВРЕДОНОСНЫХ ФАЙЛОВ |

**КУРСОВАЯ РАБОТА**

по дисциплине «Защита в операционных системах»

Специальность 10.05.01 «Компьютерная безопасность»

Очной формы обучения

|  |
| --- |
| Студент группы С8117 ммзи \_\_\_\_\_\_\_\_\_  (подпись) |
| Руководитель  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись) (и.о.ф) |
| Регистрационный № \_\_\_\_\_\_\_\_  \_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  подпись И.О.Фамилия  « \_\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2021г. | Оценка \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись) (и.о.ф)  «\_\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2021г. |

г. Владивосток

2021

Оглавление

[Введение 3](#_Toc74490481)

[1. Классификация вредоносных файлов, их свойства. 5](#_Toc74490482)

[2. Методы анализа вредоносных файлов 9](#_Toc74490483)

[2.1 Статический анализ 9](#_Toc74490484)

[2.2 Динамический анализ 9](#_Toc74490485)

[3. Подготовка системного окружения для проведения анализа вредоносного ПО 11](#_Toc74490486)

[Список литературы 14](#_Toc74490487)

# Введение

Любая инфраструктура, реализуемая с использованием информационных технологий, в той или иной степени подвержена угрозе нарушения информационной безопасности, причем атакам злоумышленников подвергаются как компании коммерческого, так и государственного секторов. Зачастую авторы вредоносных программ даже не нацеливают свое ПО против конкретных компаний – их целью является нанесение максимального вреда и придание огласке такого рода происшествий.

Подверженность различных инфраструктур атакам вредоносных программ объясняется не только наличием уязвимостей в используемом ими программном обеспечении (далее – ПО), но и в неправильной или недостаточной его настройке. Распространенным решением по защите компьютерной системы от различных видов вредоносного ПО является использование антивирусных программ, однако на данный момент ни один антивирусный продукт не может гарантировать абсолютную защиту от вредоносных файлов.

Одним из способов по уменьшению рисков заражения и получению более точных результатов анализа файлов является объединение нескольких технологий в одну платформу. Существуют сервисы, использующие на своей основе практически все известные антивирусные программные модули сканирования, что в конечном счете позволяет проанализировать отчеты о сканировании экземпляра от разных продуктов и сформировать наиболее точный вердикт. Основная проблема, встающая на пути использования подобных сервисов – конфиденциальность данных. В силу различных ограничений многие организации не могут свободно выгружать имеющиеся у них файлы на проверку.

В связи с этим целью данной работы является разработка платформы для локального мониторинга и анализа экземпляров вредоносных файлов. Для достижения цели поставлены следующие задачи:

1. Изучить основные классы вредоносных файлов, их свойства;
2. Изучить основные методы анализа файлов;
3. Подготовить системные окружения для безопасного анализа вредоносного ПО;
4. Разработать программу, реализующую управление задачами по сканированию файлов.

# Классификация вредоносных файлов, их свойства.

Вредоносное ПО можно классифицировать по достаточно большому спектру признаков: способам размножения, особенностям алгоритма работы и сложности кода, деструктивным возможностям, использованию интернет-технологий, способам запуска, средам обитания, способам проникновения в систему, времени воздействия и целевым операционным системам [1, 2]. Так как не существует единой классификации для всех видов и семейств вредоносных объектов, для примера приведем лишь некоторые примеры из классификации, представленной на сайте компании «Лаборатория Касперского» [3]. Вредоносные программы делятся на:

* Черви
  + Net-Worm
  + Email-Worm
  + IM-Worm
* Вирусы
* Троянские программы
  + Backdoor
  + Rootkit
  + Trojan-Spy
  + Trojan-Downloader
  + Trojan-Dropper
  + Trojan-Mailfinder
  + Trojan-DDoS
  + Trojan-Clicker
  + Trojan-Ransom
* Вредоносные утилиты
  + Exploit
  + VirTool
  + HackTool
  + Constructor

В первую очередь программы делятся по способам их распространения – наиболее опасными являются черви, так как они могут распространяться через сети – по каналам обмена сообщениями, файлами, в одноранговых (P2P) сетях и т.д. Среди червей есть особый класс так называемых сетевых червей (Net-Worm). Их особенность заключается в том, что для распространения им не нужен пользователь, т.е. они могут самостоятельно распространяться и запускаться на уязвимых компьютерах. В основном это становится возможно при наличии критической уязвимости у компьютеров, подключенных к сети, позволяющей послать специально сформированный пакет, который позволит коду червя проникнуть на компьютер жертвы и запустит его на выполнение.

Следующими по способу распространения являются вирусы – они способны распространяться только в пределах локальных ресурсов компьютера. Распространение вирусных программ происходит следующим образом: вирус ищет необходимые ему файлы, открывает их с правами записи и дописывает к ним свой код. Далее, если пользователь запустит уже зараженный файл на своем компьютере, он будет пытаться заражать остальные документы. Вирусы зачастую дописывают себя в начало файла – в секцию, выполняющуюся самой первой при его запуске. В исполняемых файлах такой секцией является точка входа.

Далее – троянские программы. В их функционал не заложены механизмы самораспространения и не могут заразить собой другие файлы. В классификации приведены не все виды, так как некоторые из них представляют собой лишь улучшенные или более узконаправленные их варианты. Ниже дадим краткое пояснение этим видам.

***Backdoor***

Backdoor - это класс вредоносных программ, позволяющий злоумышленнику получить контроль на компьютером жертвы и посылать на него различные команды. Специфика этих команд определяется автором трояна, однако при желании он может получить практически полный доступ к конечному узлу.

***Rootkit***

Зачастую в качестве таких программ выступают драйверы, предназначенные для сокрытия определенной активности в системе. В основном используются в связке с другим вредоносным ПО для уменьшения вероятности его обнаружения.

***Trojan-Spy***

К этому классу относятся вредоносные программы, ворующие личные данные жертвы и пересылающие их злоумышленнику. Пересылка может осуществляться практически любыми способами.

***Trojan-Downloader***

Это программы, скрыто загружающие из сети на компьютер жертвы другие вредоносные файлы и запускающие их на выполнение.

***Trojan-Dropper***

Класс программ-дропперов отличается тем, что они не загружают вредоносные программы из интернета, а распаковывают их из своего тела. В качестве вредоносного наполнения может служить не только исполняемый файл, но и самый обыкновенный скрипт, написанный на практически любом языке программирования. В таком случае в теле троянца иногда можно найти даже строчки кода, которые он впоследствии поместит в файл и запустит его на выполнение.

***Trojan-Mailfinder***

Класс программ, описывающий деятельность вредоносов, занимающихся поиском файлов, которые могут содержать в себе адреса электронных почт (к примеру, адресные книги). После сбора адресов она отправляет данные злоумышленнику.

***Trojan-DDoS***

К этому классу относятся вредоносные программы, проводящие атаки типа «отказ в обслуживании» на заранее указанный компьютер жертву.

***Trojan-Clicker***

Используются злоумышленниками либо для искусственной накрутки посещаемости сайтов, либо для привлечения потенциальных жертв заражения другими вредоносными файлами. Такие программы нередко представляют собой небольшой отрывок кода, встроенный в веб-страницы, который обращается к определенным веб-ресурсам в скрытом окне.

***Trojan-Ransom***

К ним относятся вредоносные программы-вымогатели, вынуждающие своих жертв платить выкуп за восстановление поврежденной, украденной информации, за возвращение доступа к заблокированным функциям компьютера и т.д.

Зачастую вредоносные программы сочетают в себе сразу большое количество функций, позволяющих им как распространяться по сети, так и, к примеру, проводить DDoS-атаки, поэтому конечный класс и вид программы определяется сначала ее способом распространение, а затем по вредоносной нагрузке. Так, например программа, загружающая на компьютер жертвы другую вредоносную программу, меняющая все ярлыки на рабочем столе, крадущая личные данные пользователя и распространяющаяся по локальным ресурсам компьютера в рамках данной классификации будет обнаружена антивирусом как “Virus”.

# 2. Методы анализа вредоносных файлов

Анализ вредоносных файлов можно разделить на два этапа – статический и динамический.

## 2.1 Статический анализ

Статический анализ подразумевает под собой исследование программы на наличие вредоносного кода без ее запуска. В него входит поиск необычных для чистых файлов артефактов, строк, ресурсов, чтение дизассемблированного кода программы, сверка сигнатур с базами данных, а также эвристический анализ (выполняется непосредственно антивирусом).

Эвристический анализ — это метод поиска программных дефектов в коде программы, подразумевающий под собой сопоставление фрагментов кода с имеющейся базой данных дефектов безопасности.[4] Здесь под дефектами безопасности понимаются недостатки в реализации программы, потенциально влияющие на состояние защищенности информации. Такой подход может быть очень полезен, к примеру, для обнаружения вредоносных программ с высокой степенью полиморфизма. Авторы некоторых вирусов разрабатывают их таким образом, чтобы вредоносный код формировался в процессе их исполнения и видоизменялся от раза к разу. Таким образом, невозможно создать универсальную сигнатуру для обнаружения подобного вируса – у каждого файла зараженная область будет практически полностью внешне отличаться, что не позволит антивирусу обнаружить ее по своим вирусным базам.

## 2.2 Динамический анализ

Под динамическим анализом понимают исследование поведения программы после ее непосредственного запуска. Такой подход осуществляется с помощью различного ПО, предназначенного для мониторинга процессов, служб, исходящего и входящего интернет-трафика, анализа загрузки памяти и т.д. Динамический анализ позволяет аналитику более широко понять принцип работы программы. В некоторых случаях авторы вредоносного ПО используют настолько сложные алгоритмы упаковки, шифрования, адресации и обфускации собственного кода и встроенных в них ресурсов, что иногда не представляется возможным достоверно определить класс и вредоносную нагрузку такой программы.

Так как динамический анализ подразумевает под собой запуск исследуемого объекта, в целях защиты компьютера от возможного вредоносного воздействия такого ПО используют различные методы, позволяющие изолировать его от внешнего мира. В случае, если с программой работает непосредственно вирусный аналитик, ее запускают на виртуальной машине с предустановленным ПО, необходимого для анализа.

Для автоматизации процесса анализа запущенной программы были созданы так называемые «песочницы» - изолированные среды, выполняющие определенную последовательность действий по запуску и мониторингу деятельности программы и собирающие всю информацию о ней в файлах-логах. Ранее эта технология была популярна среди антивирусного ПО, однако со временем стала реже ими использоваться ввиду высокой ресурсоемкости таких систем. На данный момент «песочницы» являются удобным инструментом для динамического анализа вредоносных программ, так как позволяют запускать в очередь на сканирование сразу несколько файлов и значительно сокращают время обработки результатов мониторинга. Аналитику лишь остается ознакомиться с результатами, которые выдаст ему изолированная среда, чтобы сделать соответствующие выводы.

Конечно, как утверждалось ранее, ни одна автоматизированная система анализа не позволяет быть стопроцентно уверенным в том, что тот или иной файл не является вредоносным, т.к. существует вероятность как ложной тревоги, так и того, что вредоносное ПО не будет обнаружено. К счастью, использование нескольких технологий статического и динамического анализа может позволить значительно сократить шанс подобного инцидента.

# 3. Подготовка системного окружения для проведения безопасного анализа вредоносного ПО

## 3.1 Выбор необходимых для работы инструментов

Для достижения поставленной цели в данной работе были использованы следующие инструменты:

* VMWare Workstation 16 Player – средство виртуализации операционных систем, на которых будет проводиться дальнейший анализ;
* Windows 10 – ОС, на которой будет проводиться статический анализ файлов путем сканирования с помощью инструментов антивирусного ПО;
* Ubuntu 16.04 – ОС семейства Linux, используемая для работы «песочницы». На ней непосредственно будет производиться динамический анализ файлов;
* Kaspersky Security Cloud – бесплатный антивирус Лаборатории Касперского;
* ESET Internet Security 14.1.20 – пробная версия антивируса компании ESET;
* Защитник Windows – встроенное антивирусное ПО компании Microsoft;
* Cuckoo Sandbox – «песочница», используемая для динамического анализа файлов;
* VirtualBox - средство виртуализации операционных систем, используемое в рамках динамического анализа «песочницей» Cuckoo Sandbox;

Приведенные выше антивирусы были выбраны по тем причинам, что не во всех бесплатных версиях антивирусных продуктов различных компаний есть сканирующий модуль, позволяющий инициировать операцию сканирования с использованием командной строки или стороннего ПО. В нашем случае у всех выбранных антивирусов присутствует функция, позволяющая проводить сканирование файлов с указанием определенных флагов прямо из командной строки.

Cuckoo Sandbox – один из бесплатных и наиболее распространенных продуктов по автоматизированному динамическому анализу вредоносного ПО. Помимо того, что он выдает полный отчет о работе файла, продукт также имеет собственный модуль по анализу результатов мониторинга и вынесению вердиктов. В связи с тем, что последняя выпущенная версия ПО была протестирована только на Ubuntu 16.04, было принято решение использовать именно эту версию ОС в целях стабильности работы «песочницы». Для своей работы Cuckoo Sandbox использует средство виртуализации VirtualBox, поэтому установлено оно будет непосредственно на Ubuntu 16.04.

## 3.2 Настройка главной машины

Так как в рамках работы будут использоваться вредоносные экземпляры, необходимо максимально полно исключить вероятность их непроизвольного запуска. Для этого все файлы должны перемещаться внутри основной системы в архиве, защищенном паролем.

При создании виртуальной машины под управлением Ubuntu необходимо включить средства вложенной виртуализации Intel VT или AMD-V в зависимости от используемого в компьютере процессора. Эти средства позволят в дальнейшем Cuckoo Sandbox запускать собственную виртуальную машину для анализа.

Для удобства обмена файлами и отчетами с виртуальными машинами необходимо создать директорию с функцией общего доступа. В данном случае функцию общего доступа достаточно настроить непосредственно в средстве виртуализации, а не глобально на компьютере. Таким образом, доступ к этой папке будут иметь только установленные виртуальные машины. Так как в папке с экземплярами могут оказаться вредоносные файлы, необходимо с помощью настроек групповых политик запретить запуск файлов из этой папки для всех пользователей системы. Для этого в редакторе групповых политик создается правило, запрещающее исполнение файлов из этой папки и указывается список расширений, попадающих под это правило. В качестве дополнительной меры предосторожности стоит всем предназначенным для сканирования файлам изменять расширение на неподходящее для запуска. Для удобства положим, что все файлы должны иметь расширение «.to\_scan\_». Это расширение также необходимо добавить в правило о запрете на запуск из общей папки.

Для удобства работы создаем расшаренную папку и запрещаем в ней выполнение любых файлов (будет использоваться для хранения семплов и их сканирования без пересылки на виртуалку).

Антивирус Касперского:

1. Прописать папку с антивирусом в PATH
2. Команда: avp.com SCAN
3. Флаги:
   1. /i0 – выдать только репорт
   2. /RA:<report\_file> - сохранить все события в репорт
   3. /R:<report\_file> - сохранить только критические события

Можно добавить разные функции: скан памяти (/MEMORY), автозагрузка (/STARTUP) и т.д. типа на выбор.

Пример команды: avp.com SCAN "E:\Programs\ubuntu-16.04.7-desktop-amd64.iso" /i0 /RA:brokendick.txt

ESET:

1. Добавляем в PATH
2. Команда: ecls [опции] “имя\_файла”
3. Используем опции:
   1. /log\_file=LOGPATH(задает пользователь в настройках скрипта)
   2. /no\_quarantine – не пихаем в карантин
   3. /adv-heur – расширенная эвристика (разницы не заметил)

Windows Defender:

1. Будем обращаться не из Path, а напрямую через команду “%ProgramFiles%\Windows Defender\MpCmdRun.exe” -Scan -ScanType 3 -DisableRemediation -File “ПУТЬ К ФАЙЛУ”

По всей видимости, придется файл из шары копировать. Антивирусы не горят желанием сканировать проги из нее. Более того, надо запускать все команды от имени администратора.

Как мы запускаем кукушку:

- Из-под рута пишем: /home/cuckoo/cuckoo/bin/cuckoo rooter –sudo –group cuckoo

- Из-под куку-пользователя пишем: . ~/cuckoo/bin/activate и cuckoo –debug

VMCloak.

Чистый образ – win7clean

Клон с надстройками – win7snap

# Список литературы

1. https://math-it.petrsu.ru/users/semenova/Informatika/DOC/MUM\_65/Lections/Comp\_bez\_Lec.pdf
2. <https://vestnik.igps.ru/wp-content/uploads/V83/9.pdf>
3. <https://encyclopedia.kaspersky.ru/knowledge/the-classification-tree/>
4. <https://cyberleninka.ru/article/n/evristicheskiy-analiz-bezopasnosti-programmnogo-koda/viewer>
5. <https://cyberleninka.ru/article/n/tehnologii-staticheskogo-i-dinamicheskogo-analiza-uyazvimostey-programmnogo-obespecheniya/viewer>
6. <https://cyberleninka.ru/article/n/o-lozhnyh-srabatyvaniyah-sredstv-zaschity-informatsii/viewer>
7. <https://xakep.ru/2020/06/08/malware-sandbox/>
8. <https://dspace.susu.ru/xmlui/bitstream/handle/0001.74/16544/2017_401_goncharukip.pdf?sequence=1?sequence=1>
9. <https://cyberleninka.ru/article/n/sistema-vyneseniya-verdiktov-po-otchetam-cuckoo-sandbox/viewer> (страница 272)