# **Отчет по практическому заданию: Анализ учебного вредоносного образца**

## **1. Теоретическая часть**

### **1.1. Вредоносное ПО и его разновидности**

**Вредоносное ПО (Malware)** — это программы, созданные для нанесения ущерба компьютеру, данным или пользователю. Основные типы:

* **Вирусы** – заражают исполняемые файлы, распространяются при запуске.
* **Трояны** – маскируются под легитимные программы, крадут данные или открывают бэкдоры.
* **Вымогатели (Ransomware)** – шифруют файлы и требуют выкуп за их расшифровку.

### **1.2. Методы анализа вредоносного ПО**

* **Статический анализ** – изучение кода без запуска (дизассемблирование, поиск строк, сигнатур).
* **Динамический анализ** – мониторинг поведения программы в изолированной среде (сетевые запросы, изменения в файловой системе и реестре).

### **1.3. Правила безопасности при анализе**

* Использовать **изолированную виртуальную машину** (без доступа к основной системе).
* Отключать **сеть** (если не требуется для анализа).
* Фиксировать все изменения через **Process Monitor** и **FakeNet-NG**.
* Не запускать подозрительные файлы на **основной ОС**.

## **2. Практическая часть**

### **2.1. Подготовка среды**

1. Установлена **Windows 10** на **VirtualBox**.
2. Установлены инструменты:
   * **FakeNet-NG** (эмуляция сети).
   * **Process Monitor** (мониторинг процессов).
   * **Ghidra** (статический анализ).
3. Сеть в ВМ **отключена** (VirtualBox → Сеть → "Not Attached").

### **2.2. Анализ EICAR-образца (eicar.com)**

#### **Статический анализ в Ghidra**

* Файл открыт в **Ghidra**, обнаружена строка:

text

Copy

Download

EICAR-STANDARD-ANTIVIRUS-TEST-FILE!$H+H\*

* Программа **не содержит вредоносного кода**, только текстовую сигнатуру для теста антивирусов.

#### **Динамический анализ с FakeNet-NG**

1. Запущен fakenet.exe -d .
2. Запущен eicar.com → **сетевых запросов не обнаружено**.

#### **Анализ поведения через Process Monitor**

* Фильтры:
  + Process Name = eicar.com
  + Operation = CreateFile, RegSetValue
* Результаты:
  + **Не создает файлы**.
  + **Не изменяет реестр**.
  + **Не запускает сторонние процессы**.

**Вывод:** EICAR — безопасный тестовый файл, не выполняющий вредоносных действий.

### **2.3. Исследование "трояна" (game.exe)**

#### **Статический анализ в Ghidra**

* Найдена строка: http://example.com (подозрение на отправку данных).
* Обнаружена функция чтения файла secret.txt.

#### **Динамический анализ через Process Monitor**

1. Запущен game.exe.
2. Зафиксированы действия:
   * Попытка **чтения C:\secret.txt**.
   * При наличии файла – **отправка POST-запроса на example.com**.
   * Если файла нет – вывод "Game started!!".

**Вывод:** Программа имитирует поведение трояна, который крадет файл и отправляет его на удаленный сервер.

## **3. Заключение**

* **EICAR** – безопасный тестовый файл, полезен для проверки антивирусов.
* **"Троян" (game.exe)** – демонстрирует типичное вредоносное поведение (кража файлов + отправка в сеть).
* **Инструменты (Ghidra, FakeNet, ProcMon)** эффективны для анализа вредоносного ПО.

# **Отчет по анализу учебных вредоносных образцов**

## **1. Таблица результатов анализа**

| **Образец** | **Создает файлы** | **Меняет реестр** | **Сетевые запросы** |
| --- | --- | --- | --- |
| eicar.com | ❌ Нет | ❌ Нет | ❌ Нет |
| game.exe | ❌ Нет | ❌ Нет | ✅ Да (POST на example.com) |

## **2. Ответы на вопросы**

### **1. Какие признаки вредоносного поведения вы обнаружили?**

* **game.exe** пытается прочитать файл secret.txt и отправить его содержимое на example.com (подозрительная сетевая активность).
* **eicar.com** не проявляет вредоносного поведения, так как это тестовый файл.

### **2. Почему EICAR безопасен для анализа?**

EICAR (eicar.com) — это стандартный **тестовый файл** для проверки антивирусов. Он содержит только текстовую строку, но не выполняет вредоносных действий.

### **3. Как FakeNet помогает анализировать сетевую активность?**

FakeNet эмулирует сетевые сервисы (DNS, HTTP, HTTPS), чтобы:

* Перехватывать запросы вредоносного ПО.
* Определять, куда программа пытается отправить данные.
* Анализировать C2 (Command & Control) серверы.

### **4. Что такое вредоносное программное обеспечение (ВПО) и какие его основные типы?**

**ВПО (Malware)** — программы, созданные для нанесения ущерба. Основные типы:

* **Вирусы** (заражают файлы).
* **Трояны** (маскируются под легитимные программы).
* **Руткиты** (скрывают присутствие зловреда).
* **Рэнсомверы** (шифруют файлы и требуют выкуп).
* **Шпионское ПО** (крадет данные).

### **5. Каковы основные цели анализа вредоносного ПО?**

* Определить **функционал** вредоноса.
* Выявить **индикаторы компрометации (IoC)**.
* Разработать **методы защиты**.
* Помочь в **киберрасследованиях**.

### **6. В чем разница между статическим и динамическим анализом?**

| **Критерий** | **Статический анализ** | **Динамический анализ** |
| --- | --- | --- |
| **Запуск кода** | Без исполнения | С запуском в изолированной среде |
| **Методы** | Дизассемблирование, поиск строк | Мониторинг процессов, сетевой активности |
| **Преимущества** | Безопасен, выявляет скрытый код | Показывает реальное поведение |

### **7. Какие инструменты используются для статического анализа?**

* **Ghidra** (дизассемблирование, декомпиляция).
* **IDA Pro** (анализ бинарного кода).
* **PEiD/Detect It Easy** (определение упаковщиков).
* **Strings** (поиск текстовых строк в бинарниках).

### **8. Что такое обфускация кода и как она используется во вредоносах?**

**Обфускация** — намеренное усложнение кода, чтобы затруднить анализ.

* **Примеры:** шифрование строк, полиморфный код, анти-отладка.
* **Цель:** избежать детектирования антивирусами.

### **9. Возможные признаки вредоносного кода в ПО**

* Подозрительные **сетевые подключения**.
* Попытки **изменения реестра/файловой системы**.
* Использование **анти-отладочных техник**.
* Необычные **импорты функций** (например, VirtualAlloc, CreateRemoteThread).

### **10. Что такое песочница (sandbox) и как она помогает в анализе?**

**Песочница** — изолированная среда для безопасного запуска подозрительных программ.

* **Примеры:** Cuckoo Sandbox, Any.Run.
* **Преимущества:** автоматический анализ поведения без риска для основной системы.

### **11. Методы обхода антивирусов**

* **Полиморфизм** (изменение кода при каждом запуске).
* **Упаковка/шифрование** (использование Packers: UPX, Themida).
* **Инъекция в легитимные процессы** (например, в explorer.exe).

### **12. Что такое "инъекция кода" и как она применяется?**

**Инъекция кода** — внедрение вредоносного кода в процесс другой программы.

* **Пример:** использование CreateRemoteThread + VirtualAllocEx.
* **Цель:** маскировка под легитимный процесс.

### **13. Основные этапы динамического анализа**

1. **Запуск в изолированной среде**.
2. **Мониторинг процессов (ProcMon)**.
3. **Анализ сетевой активности (Wireshark, FakeNet)**.
4. **Изучение изменений в реестре и файлах**.

### **14. Какие данные можно извлечь при статическом анализе?**

* **Строки** (URL, ключи шифрования).
* **Импорты функций** (API-вызовы).
* **Секции кода** (.text, .data).
* **Сигнатуры упаковщиков**.

### **15. Как изучение поведения вредоносов помогает в защите?**

* Позволяет выявлять **уязвимости** в системе.
* Помогает создавать **правила для IDS/IPS**.
* Улучшает **эвристику антивирусов**.

### **16. Распространенные сценарии использования ВПО**

* **Кража данных** (банковские трояны).
* **Шпионаж** (RAT — Remote Access Trojan).
* **Криптоджекинг** (майнинг без ведома пользователя).

### **17. Что такое обратная разработка (reverse engineering)?**

**Реверс-инжиниринг** — исследование кода программы без исходников.

* **Применение:** анализ вирусов, поиск уязвимостей.
* **Инструменты:** Ghidra, IDA Pro, x64dbg.

### **18. Риски анализа вредоносов в реальной среде**

* **Заражение системы**.
* **Утечка данных**.
* **Срабатывание антивируса на анализ**.

### **19. Меры предосторожности при анализе**

* Использовать **ВМ без сети**.
* Фиксировать все изменения (**ProcMon**).
* Не запускать на **основной ОС**.

### **20. Как анализ помогает в расследованиях?**

* Выявляет **атакующие IP/Domains**.
* Определяет **методы атаки (TTPs)**.
* Помогает в **атрибуции киберпреступников**.

### **21. Этические аспекты анализа вредоносов**

* **Не использовать знания для взлома**.
* **Не распространять исследуемые образцы**.
* **Соблюдать законы (например, GDPR при анализе утечек)**.