



# Worm Propagation Visualization Project Documentation

---

## 1. Project Overview

Този проект симулира разпространението на два компютърни червея в контролирана, виртуализирана и сегментирана мрежова среда. Той включва:

- Реалистична симулация на червеи (Code Red & Voyager)
  - Мониторинг на заразяване в реално време
  - Визуализация на заразата
  - Механизми за изолация и контрол
- 

## 2. Completed Phases



### Research and Planning

#### Червеи:

- **Code Red**
  - CVE: CVE-2001-0500
  - Target: Microsoft IIS
  - Разпространение: HTTP GET exploit
- **Voyager (Conficker-like Oracle Worm)**
  - CVE: CVE-2004-0637
  - Target: Oracle 9i/10g
  - Разпространение: SQL Injection чрез TCP/1521

#### Изолация и дизайн:

- Vagrant + VirtualBox
  - Host-only мрежи
  - Планиране с Cisco Packet Tracer
  - Разделяне по роли: web, oracle, bridge, monitor
-

### 3. Virtual Network Architecture

#### Subnets & Roles

Segment	Subnet	Role
Web-net	192.168.56.0/24	Target на Code Red
Oracle-net	192.168.57.0/24	Target на Voyager
Bridge-net	192.168.59.0/24	Кръстосано движение
Monitor-net	192.168.60.0/24	Мониторинг и контрол

#### VMs:

- **web1–web6:** Уязвими IIS сървъри
  - **oracle1–oracle5:** Oracle 9i/10g бази
  - **bridge1, bridge2:** Свързват сегментите
  - **monitor, control:** Наблюдение, администриране
- 




### 4. Scripts and Automation

- `common.sh`: Инсталира системни пакети, създава директории, подготвя среда
  - `codeRed.py`: Симулира зараза по web-мрежата
  - `voyager.py`: Симулира зараза в oracle-сегмента
  - `monitor.py`: Чете логове в реално време
  - `visualizer.py`: Рисува текущата зараза по данни от логовете
  - `setup.sh`: Централизирано пускане на скриптове и мониторинг
- 







### 5. Monitoring and Visualization

- **Мониторинг** (`monitor.py`):
    - Следи `code_red.log` и `voyager.log` в реално време
    - Отделно за всеки червей
  - **Визуализация** (`visualizer.py`):
    - Използва `NetworkX` и `Matplotlib`
    - Оцветява заразените възли
    - По избор: обновява при нова зараза
-

## 6. Testing and Validation

-  **Containment Tests:** Проверено, че VM-ите нямат външна връзка
-  **Functional Tests:** Заразата се разпространява, логовете се попълват
-  **Reset Mechanism** (в процес):
  - `vagrant snapshot`
  - Или ръчно през `reset.sh`

## 7. Remaining Tasks

Task	Status
Финализиране на <code>monitor.py</code>	 Завършен
Финализиране на <code>visualizer.py</code>	 Завършен
DFS анимация на заразата	 Завършен
Демонстрационно видео	 Завършен
Финален доклад/отчет	 Готов
Обобщение на AI помощта	 Завършено

## 8. Containment Strategy

- Host-only мрежи
  - Без интернет, NAT, bridge или споделени папки
  - Контрол само през `monitor/control`
  - pfSense VM (по избор) за бъдещо ниво на контрол
  - Само bridge VM-ите имат мулти-сегментен достъп
- 

## 9. Resource Management

- Стартиране на VM-и на групи
- $\text{RAM} \leq 384\text{MB}$  на VM
- Спиране на неизползвани машини
- Snapshot-и за бързо възстановяване

## 10. AI Usage

- ChatGPT used for:
  - Script fixing and provisioning help
  - Security containment ideas
  - Network architecture validation

- Drafting of documentation
- All results **reviewed manually** before implementation
- The AI used **only for assistance**, not decision-making