

Переполнение буфера

Цель

Получение контроля над объектом атаки

Механизм реализации

Запуск кода на атакуемом узле

Местонахождение атакующего

В разных сегментах с объектом атаки

Используемые уязвимости

Ошибки реализации

Степень риска **Высокая**

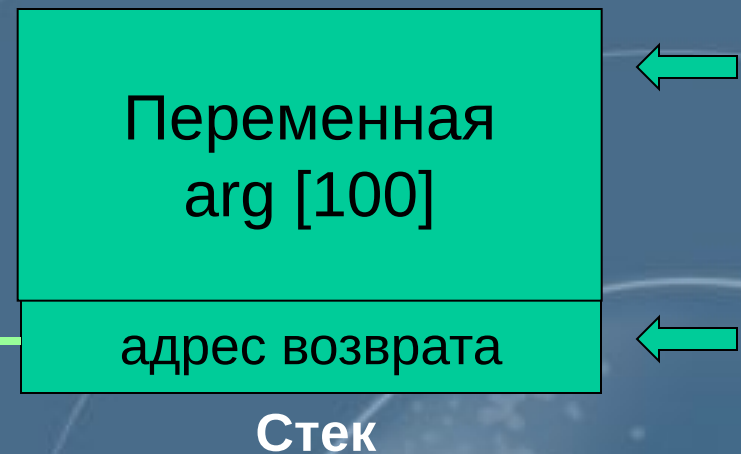
Пример атаки на IP - сеть: «Переполнение буфера»

```
→ int f_vulner (char arg)
{
→ char local[100]
→ //обработка
→ return 0
}
```

strcpy(local, arg)

```
→ void main()
{
→ char arg[200]
→ gets (arg)
→ .
→ .
→ f_vulner (arg)
→ printf(arg)
→ return 0
}
```

Обычный ход выполнения программы



Пример атаки на IP - сеть: «Переполнение буфера»

→ int f_vulner (char arg)

{

→ char local[100]

→ //обработка

→ return 0

}

void main()

{

char arg[200]

gets (arg)

.

.

→ f_vulner (arg)

printf(arg)

return 0

}

strcpy(local, arg)

Ошибка !

Переполнение стека

Вредоносный

код

[200]

Стек

Вместо возврата
запуск кода

Причины переполнения буфера

Отсутствие необходимых проверок на корректность аргументов

strcpy(local, arg)

Отсутствие средств вычисления длины буфера при работе с указателями



Abcd.....?

Последствия переполнения буфера

Чтение ячеек памяти, не принадлежащих массиву

Модификация ячеек памяти

- *Системные данные (адрес возврата и т. д.)*
- *Другие переменные*
- *Исполняемый код*
- *Несуществующая (свободная область)*

Этапы проведения атаки «Переполнение буфера»

Подготовка враждебного кода

Под видом команд или параметров уязвимого приложения

В адресном пространстве уязвимого приложения (без параметров)

В адресном пространстве уязвимого приложения (с параметрами)

Передача управления враждебному коду

Предотвращение ошибок переполнения

Использование механизма структурных исключений

Несуществующая область

Буфер

Несуществующая область

Использование языков программирования, делающих невозможным переполнение буфера

Использование «Неар» для указателей

Отказ от индикатора завершения

Методы защиты

Установка пакетов исправления

Исправление исходного кода с
перекомпиляцией

Тестирование программ специальными
утилитами

Пример атаки на IP - сеть: «Троянский конь»

Цель

Получение контроля над объектом атаки

Механизм реализации

Запуск кода (приложения) на объекте атаки

Местонахождение атакующего

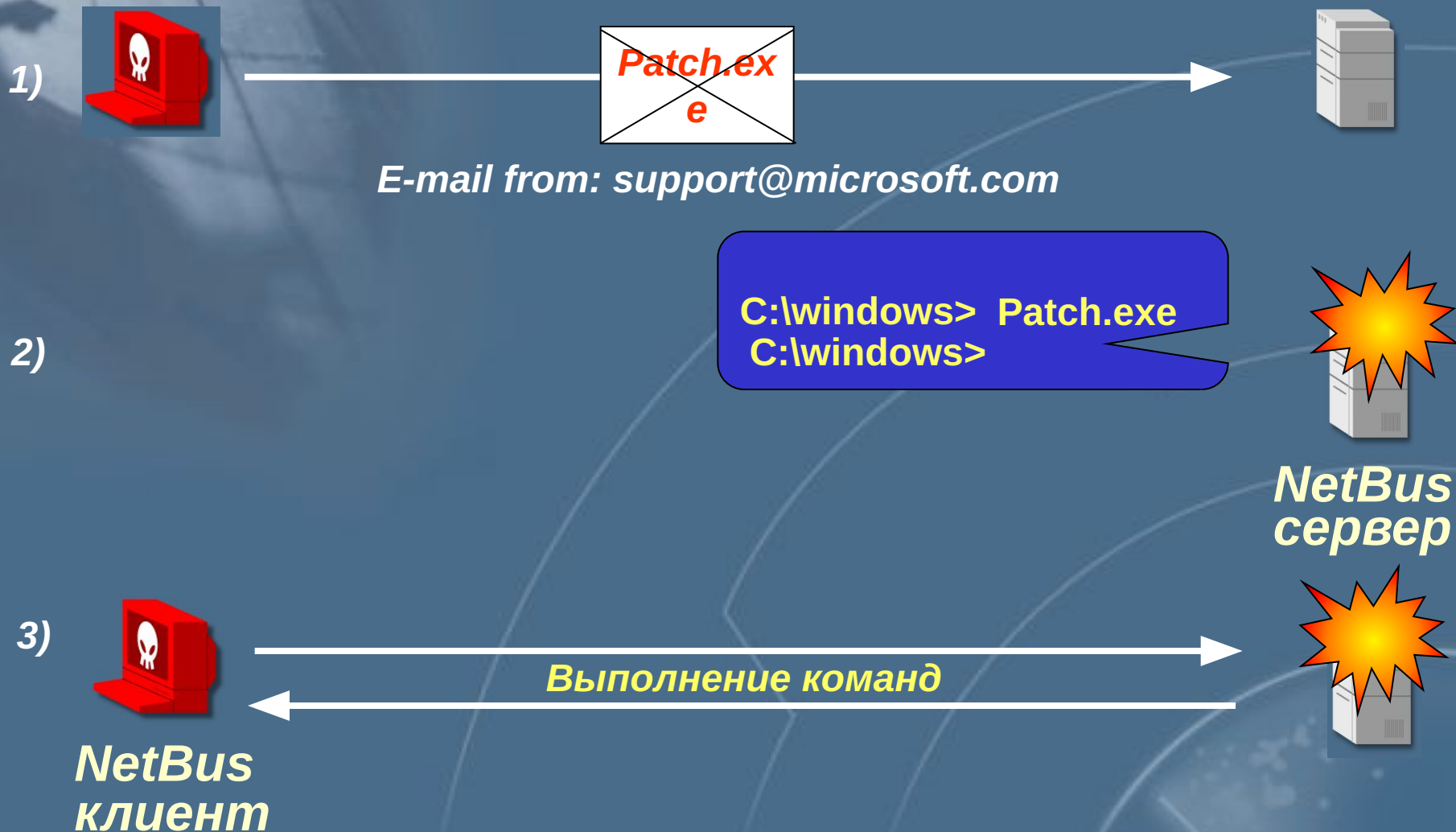
В разных сегментах с объектом атаки

Используемые уязвимости

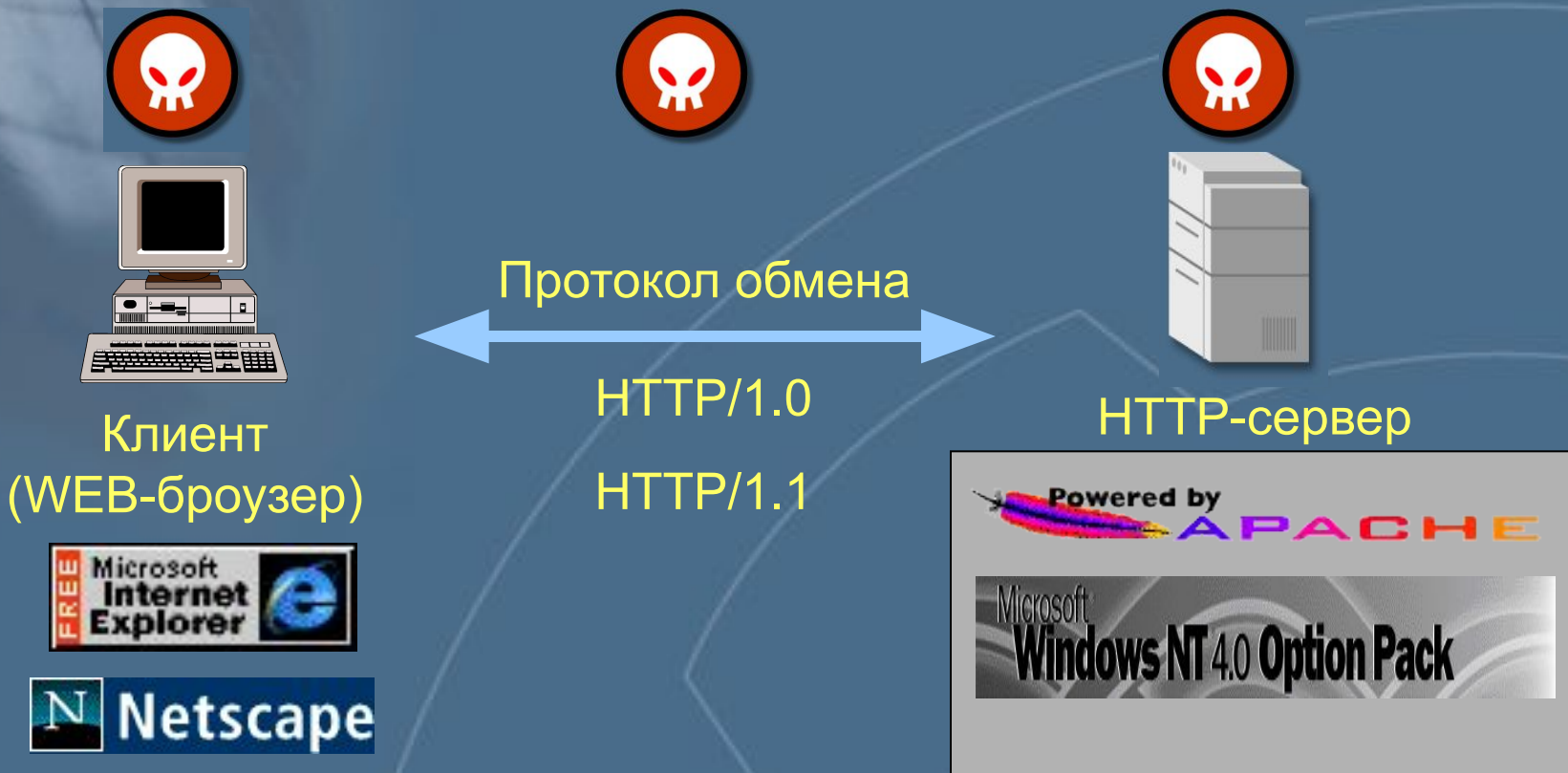
Ошибки эксплуатации (особенности психологии)

Степень риска **Высокая**

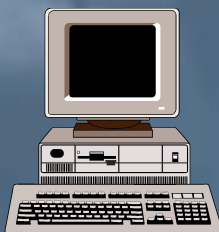
Пример атаки на IP - сеть: «Троянский конь»



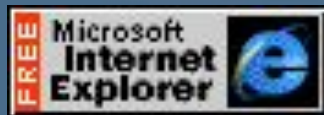
Реализация WWW-службы



Пример уязвимости WWW-клиента



Клиент
(WEB-браузер)



Hacker's
Web site



C:\...\StartUp\ RunMe.hta

Уязвимости WWW-серверов

- *Уязвимости программной реализации (ошибки кода)*
- *Уязвимости информационного наполнения*
- *Ошибки обслуживания (настройки)*

Отказ в обслуживании «IIS_DoS»

Цель

Нарушение нормального функционирования объекта атаки

Механизм реализации

*Бесполезное расходование вычислительного ресурса
(посылка некорректных HTTP-запросов)*

Местонахождение атакующего

В разных сегментах с объектом атаки

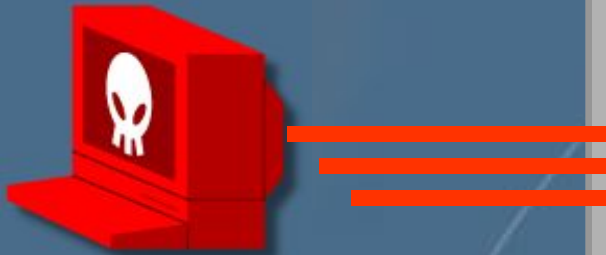
Используемые уязвимости

Ошибка в реализации MS Internet Information Server

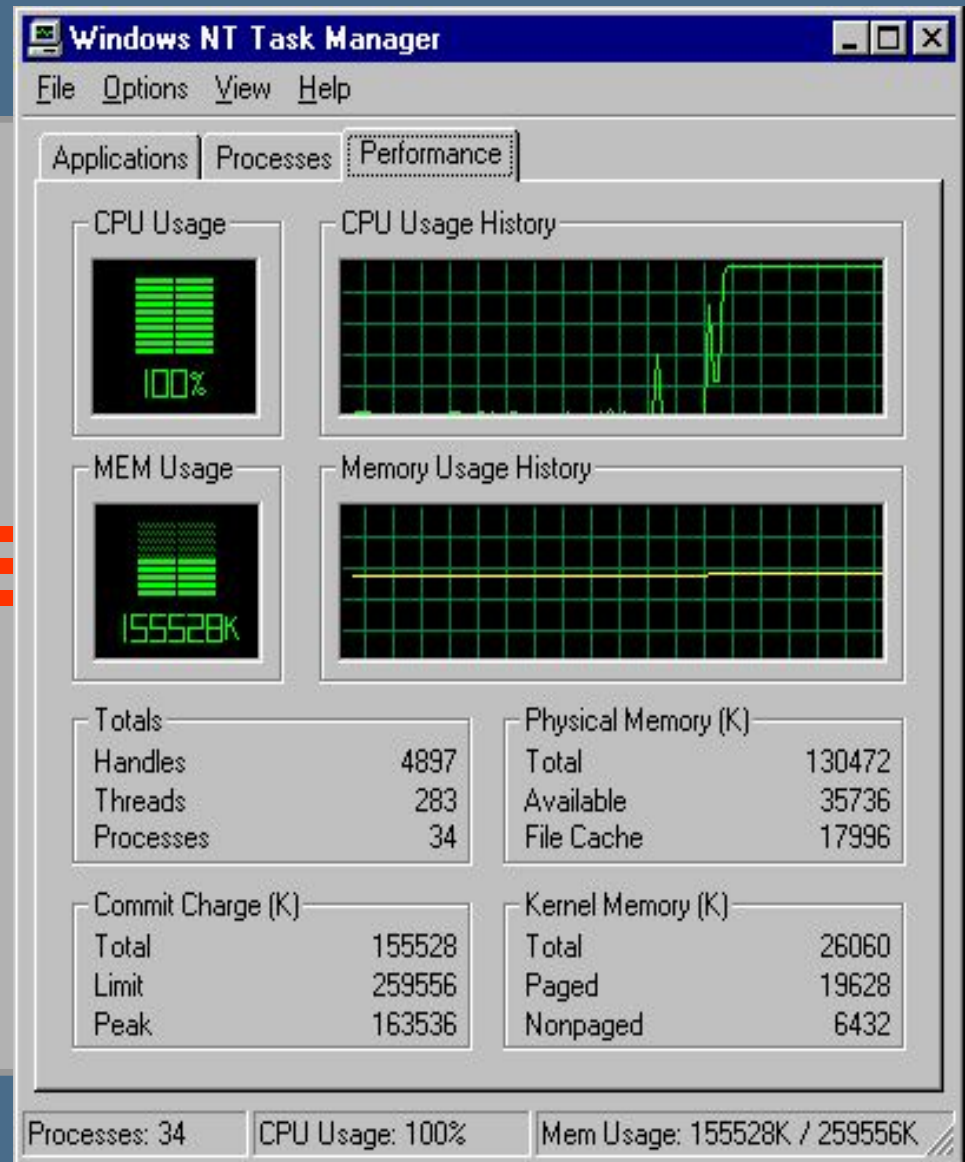
Степень риска Средняя

Отказ в обслуживании «IIS_DoS»

HACKER
200.0.0.12



C:\HackTools \ iisdos.exe



Ошибка обработки имён CGI - скриптов

Цель

Получение контроля над объектом атаки

Механизм реализации

Запуск кода на объекте атаки

Местонахождение атакующего

В разных сегментах с объектом атаки

Используемые уязвимости

Ошибка в реализации MS Internet Information Server

Степень риска Средняя

Ошибка обработки имён CGI - скриптов

Описание уязвимости

HTTP



CVE: CAN-2000-0886

C:\dir

Опубликовано на сайте:

NSFOCUS INFORMATION TECHNOLOGY CO.,LTD

(<http://www.nsfocus.com>)

ИНФОРМЗАЩИТА

НАУЧНО-ИНЖЕНЕРНОЕ ПРЕДПРИЯТИЕ

Ошибка обработки имён НТР - файлов

Описание уязвимости

HTTP

<http://site/scripts/test.bat+.httr>



Содержимое файла
test.bat