



Национальный аэрокосмический университет им. М.Е. Жуковского
"Харьковский авиационный институт"
Кафедра компьютерных систем и сетей

Средства автоматической настройки функций безопасности веб-сервера

ВЫПОЛНИЛ: СТУДЕНТ 545-И ГРУППЫ ПОДДУБНЫЙ К.А.

РУКОВОДИТЕЛЬ: СТ. ПРЕПОДАВАТЕЛЬ КАФ. 503 ЦУРАНОВ М.В.

Постановка задачи

Цель – разработка программного средства для автоматической настройки функций безопасности веб-сервера Apache.

Задачи:

1. Проанализировать наиболее распространенные веб-сервера.
2. Разработать bash-скрипт для автоматической настройки функций безопасности веб-сервера Apache.
3. Проверить настроенные функции безопасности веб-сервера Apache.
4. Рассчитать себестоимость разработанного программного средства.

Сравнительный анализ веб-серверов

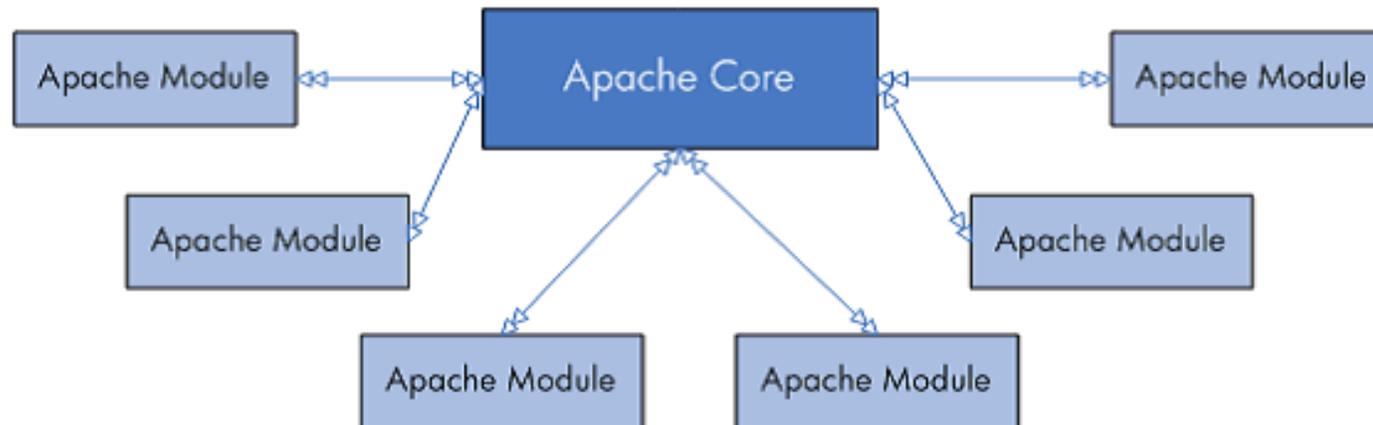
Таблица 1 – Краткое сравнение веб-серверов Apache, IIS и nginx

Название	Apache	IIS	nginx
Автор, год создания	Apache Software Foundation, 1995	Microsoft, 1995	Игор Сисоев, 2002
Распространение	Бесплатное	Включен в Windows NT	Бесплатное
Открытое ПО	+	-	+
Особенности	Акцент на надежность и гибкость. Модульность	Поддержка .NET и сценариев ASPX	Разрабатывался для серверов с высокой нагрузкой

Таблица 2 – Поддержка ОС веб-серверами Apache, IIS и nginx

Название	Apache	IIS	nginx
Windows	+	+	+
Linux	+	-	+
Mac OS X	+	-	+
BSD	+	-	+

Архитектура веб-сервера Apache



Популярные модули безопасности веб-сервера Apache

Название модуля	Выполняемые функции
mod_security	Модуль, с помощью которого создается первичный «щит» защиты веб-сервера.
mod_evasive	Модуль для организации защиты веб-сервера от DoS и brute force атак. Модулем осуществляется контроль интенсивности запросов с одного IP за период времени и числа параллельных запросов. После превышения определенного в конфигурации лимита, осуществляется временная блокировка доступа злоумышленника, возможен вызов внешнего скрипта для блокирования через фаервол.
mod_qos	При нехватке серверных ресурсов mod_qos может блокировать неприоритетные запросы, динамически изменять значения таймаутов, добавлять искусственную задержку перед выполнением запроса и принудительно завершать TCP-соединения.
mod_headers	Этот модуль обеспечивает директивы управления и изменения заголовков запроса и ответа HTTP.
mod_ssl	Модуль включающий поддержку протоколов безопасной передачи информации Secure Sockets Layer и Transport Layer Security.

Выбор инструмента разработки и программного обеспечения

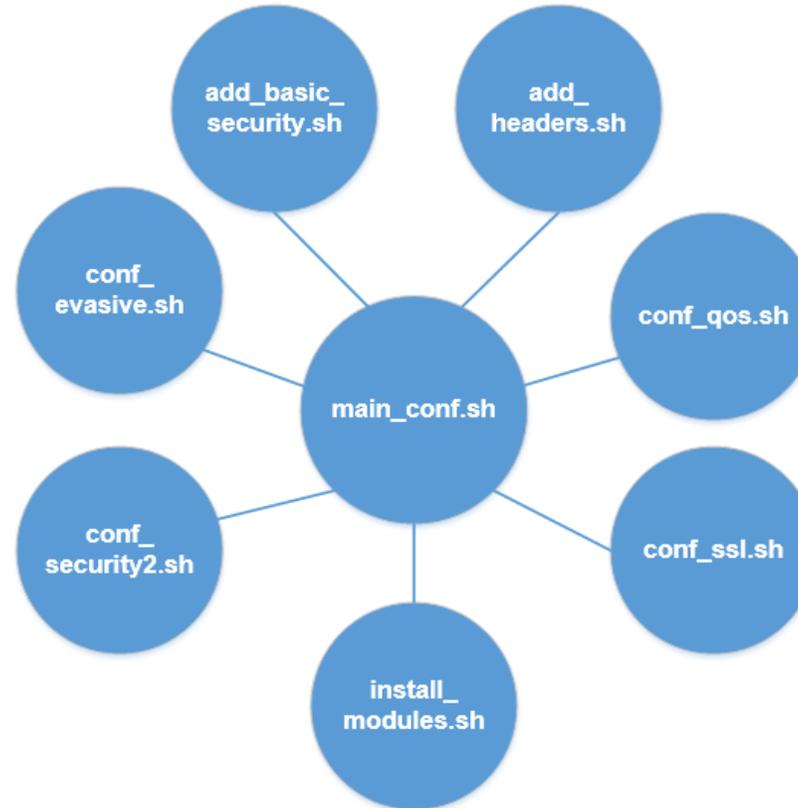
Разработка:

- Операционная система Ubuntu.
- Скриптовый язык bash.
- Текстовый редактор gedit.

Тестирование:

- Операционная система Kali Linux.
- Инструмент для моделирования атак «отказ в обслуживании» slowhttptest.
- Онлайн сканер HTTP заголовков securityheaders.io.

Архитектура разработанного продукта



Интерфейс разработанного продукта

```
kp@ubuntu:~/Desktop/SecurityConfiguration$ sudo sh main_conf.sh
Module headers already enabled

[Заголовки успешно добавлены!]

[Вывод информации о версиях ПО отключен!]

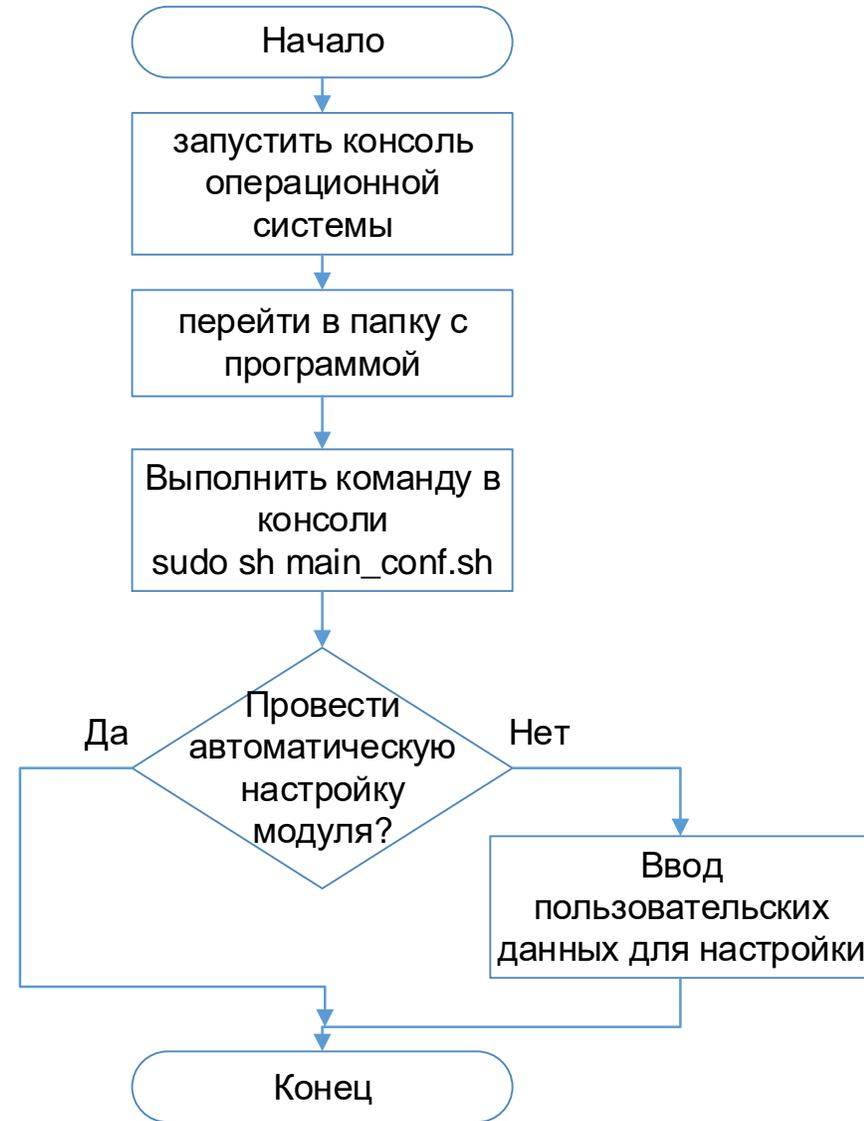
Module evasive already enabled
Провести автоматическую настройку mod_evasive? (y/n) y
Ввели «у», начинается автоматическая настройка...

[Конфигурация mod_evasive завершена!]

Module qos already enabled
Провести автоматическую настройку mod_qos? (y/n) y
Ввели «у», начинается автоматическая настройка...

[Конфигурация mod_qos завершена!]
```

Схема действий пользователя при автоматической настройке модуля



Тестирование продукта (1)

```
Sun May 29 13:14:36 2016:
slowhttptest version 1.6
- https://code.google.com/p/slowhttptest/ -
test type: SLOW BODY
number of connections: 1000
URL: http://93.95.186.228/
verb: FAKEVERB
Content-Length header value: 8192
follow up data max size: 22
interval between follow up data: 110 seconds
connections per seconds: 200
probe connection timeout: 3 seconds
test duration: 240 seconds
using proxy: no proxy

Sun May 29 13:14:36 2016:
slow HTTP test status on 15th second:

initializing: 0
pending: 598
connected: 402
error: 0
closed: 0
service available: NO
```

Рисунок 1 – Результат моделирования атаки «SLOW BODY» без проведения настройки модулей

```
slowhttptest version 1.6
- https://code.google.com/p/slowhttptest/ -
test type: SLOW READ
number of connections: 1000
URL: http://93.95.186.228/
verb: GET
receive window range: 10 - 20
pipeline factor: 1
read rate from receive buffer: 32 bytes / 5 sec
connections per seconds: 1000
probe connection timeout: 5 seconds
test duration: 350 seconds
using proxy: no proxy

Sun Jun 5 13:27:34 2016:
slow HTTP test status on 15th second:

initializing: 0
pending: 666
connected: 334
error: 0
closed: 0
service available: NO
```

Рисунок 2 – Результат моделирования атаки «SLOW READ» без проведения настройки модулей

Тестирование продукта (2)

```
number of connections:      1000
URL:                        http://93.95.186.228/
verb:                       FAKEVERB
Content-Length header value: 8192
follow up data max size:    22
interval between follow up data: 110 seconds
connections per seconds:    200
probe connection timeout:   3 seconds
test duration:              240 seconds
using proxy:                no proxy

Sun May 29 13:02:11 2016:
slow HTTP test status on 5th second:

initializing:      0
pending:           2
connected:         12
error:             0
closed:            844
service available: YES
Sun May 29 13:02:13 2016:
Test ended on 7th second
Exit status: No open connections left
root@kali:~#
```

Рисунок 1 – Результат моделирования атаки «SLOW BODY» после проведения настройки модулей

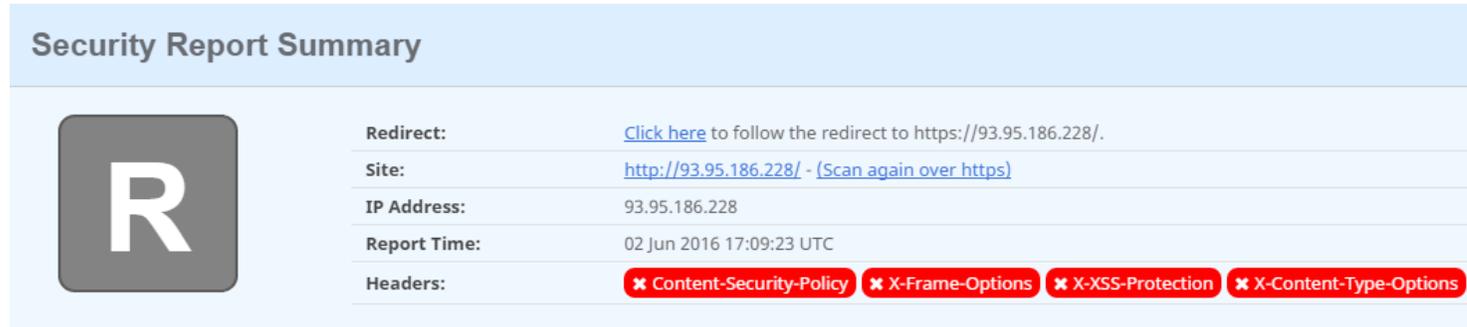
```
slowhttptest version 1.6
- https://code.google.com/p/slowhttptest/ -
test type:              SLOW READ
number of connections:  1000
URL:                    http://93.95.186.228/
verb:                   GET
receive window range:  10 - 20
pipeline factor:        1
read rate from receive buffer: 32 bytes / 5 sec
connections per seconds: 1000
probe connection timeout: 5 seconds
test duration:          350 seconds
using proxy:            no proxy

Sun Jun  5 13:26:41 2016:
slow HTTP test status on 40th second:

initializing:      0
pending:           0
connected:         294
error:             0
closed:            706
service available: YES
```

Рисунок 2 – Результат моделирования атаки «SLOW READ» после проведения настройки модулей

Тестирование продукта (3)

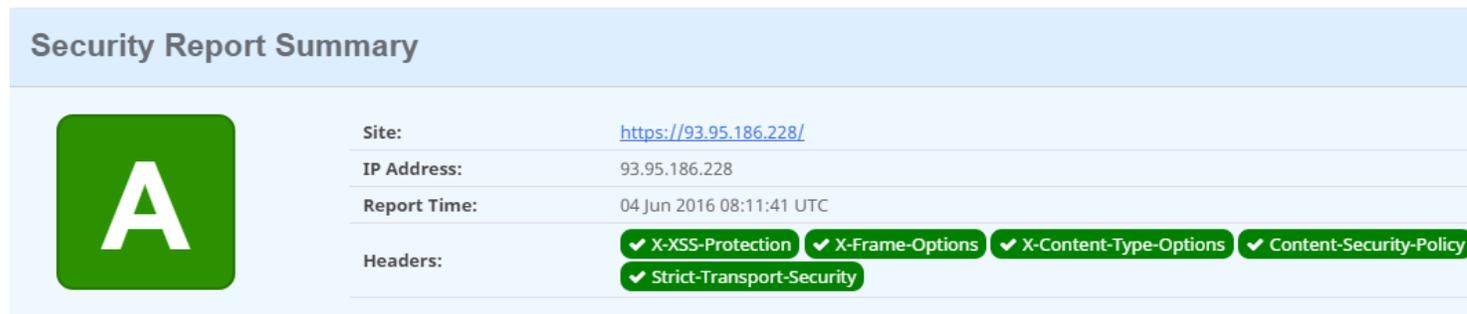


Security Report Summary

R

Redirect:	Click here to follow the redirect to https://93.95.186.228/ .
Site:	http://93.95.186.228/ - (Scan again over https)
IP Address:	93.95.186.228
Report Time:	02 Jun 2016 17:09:23 UTC
Headers:	✗ Content-Security-Policy ✗ X-Frame-Options ✗ X-XSS-Protection ✗ X-Content-Type-Options

Рисунок 3 – Отчет сканера securityheaders.io без проведения настроек



Security Report Summary

A

Site:	https://93.95.186.228/
IP Address:	93.95.186.228
Report Time:	04 Jun 2016 08:11:41 UTC
Headers:	✓ X-XSS-Protection ✓ X-Frame-Options ✓ X-Content-Type-Options ✓ Content-Security-Policy ✓ Strict-Transport-Security

Рисунок 4 – Отчет сканера securityheaders.io после проведения настроек

Тестирование продукта (4)

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <title>Main page</title>
  <link rel="stylesheet" href="style.css">
  <script>
    alert('HACKED');
  </script>
</head>
```

Рисунок 5 – Инлайновый JavaScript-код выполняемый на странице

```
✘ Refused to execute inline script because it violates 93.95.186.228/:7
the following Content Security Policy directive: "default-src 'self'".
Either the 'unsafe-inline' keyword, a hash ('sha256-
NB3gaBNAmqqNEpNiWX3dDtfJT2y/LTNZmKb7dJ4qde8='), or a nonce ('nonce-...')
is required to enable inline execution. Note also that 'script-src' was
not explicitly set, so 'default-src' is used as a fallback.
```

Рисунок 6 – Сообщение об ошибке выполнения инлайнового JavaScript-кода

Экономическая часть

Таблица 3 – Статьи калькуляции на разработку

Статья калькуляции	Затраты, грн
Материалы	4880,43
Основная заработная плата	7613,59
Дополнительная заработная плата	1522,72
Фонд заработной платы	9136,31
Начисления на заработную плату (единый и социальный взнос)	2009,99
Амортизационные отчисления	966
Затраты на электроэнергию	900
Себестоимость разработки	17892,73

Выводы

1. В ходе статистического анализа использования веб-серверов установлено, что наиболее распространенными и популярными на информационном рынке являются веб-сервера Apache, IIS и nginx. Проведенная сравнительная характеристика данных веб-серверов показывает преимущества веб-сервера Apache.
2. Для автоматической настройки функций безопасности веб-сервера Apache разработан bash-скрипт. В программном продукте использовалась модульная архитектура, которая позволяет настраивать как все модули одновременно, так и выборочно.
3. В ходе тестирования настроенных функций безопасности веб-сервера Apache с использованием моделирования сценариев атак «отказ в обслуживании» и XSS-атак веб-сервер успешно отразил их. Сканирование онлайн сканером HTTP заголовков securityheaders.io показало, что веб-сервер использует дополнительные HTTP заголовки безопасности. После автоматической настройки функций безопасности веб-сервера Apache подключение пользователя к сайту, размещенному на веб-сервере, выполнялось по безопасному протоколу с набором надежных шифров.
4. Себестоимость программного продукта по расчетам статей калькуляции составил 17892,73 грн.

Спасибо за внимание