Джефф Форристал, Крис Брумс, Дрю Симонис, Брайн Бегнолл,
Майкл Дайновиц, Джей Д. Дайсон, Джо Дьюлэй, Майкл Кросс,
Эдгар Даниелян, Дэвид Г. Скабру

Защита от хакеров Web-приложений



The Only Way to Stop a Hacker is to Think Like One

Jeff Forristal
Chris Broomes
Drew Simonis
Brian Bagnall
Michael Dinowitz
Jay D. Dyson
Joe Dulay
Michael Cross
Edgar Danielyan
David G. Scarbrough

SYNGRESS®



Серия «Информационная безопасность»

Единственный способ остановить хакера – это думать, как он

Перевод с английского
Виталия Зорина

Джефф Форристал
Крис Брумс
Дрю Симонис
Брайн Бегнолл
Майкл Дайновиц
Джей Д. Дайсон
Джо Дьюлэй
Майкл Кросс
Эдгар Даниелян
Дэвид Г. Скабру

Академия АйТи
ДМК
издательство
Москва

УДК 004.056
ББК 32.973.202
Ф79

Ф79 **Форристал Д. и др.**
Защита от хакеров Web-приложений / Джефф Форристал, Крис Брумс, Дрю Симонис, Брайн Бегнолл, Майкл Дайновиц, Джей Д. Дайсон, Джо Дьюлэй, Майкл Кросс, Эдгар Даниелян, Дэвид Г. Скабру ; Пер. с англ. В. Зорина – М. : Компания АйТи ; ДМК Пресс. – 496 с. : ил. – (Серия «Информационная безопасность»).

ISBN 5-98453-006-6 (АйТи) – **ISBN 5-94074-258-0** (ДМК Пресс)

В этом издании представлены инструментальные средства для защиты Web-приложений и подробно рассмотрены базовые подходы для обеспечения безопасного управления сайтами, разработки безопасного кода, планирования защиты. Книга научит читателя «думать, как хакер» и благодаря этому лучше защищать информационные ресурсы сайта и конфиденциальные данные, обеспечивая их целостность. Внимание читателей акцентируется на проблеме безопасности от начальной до конечной стадии разработки приложений.

Подробно рассматриваются такие важные темы обеспечения успешной защиты Web-приложений от атак, как разработка плана сетевой безопасности, плана безопасности приложений и защиты рабочих мест, тестирование, изменение и совершенствование прикладных методов и средств, а также методы осуществления угроз безопасности. В книге представлен обзор сайтов и форумов для разработчиков, публикующих перечень всех известных текущих угроз и проводящих консультации по проблемам безопасности.

УДК 004.056
ББК 32.973.202

Original English language edition published by Syngress Publishing, Inc. Copyright © 2001 by Syngress Publishing, Inc. All rights reserved.

Все права защищены. Любая часть этой книги не может быть воспроизведена в какой бы то ни было форме и какими бы то ни было средствами без письменного разрешения владельца авторских прав.

Материал, изложенный в данной книге, многократно проверен. Но, поскольку вероятность технических ошибок все равно остается, издательство не может гарантировать абсолютную точность и правильность приводимых сведений. В связи с этим издательство не несет ответственности за возможный ущерб любого вида, связанный с применением содержащихся здесь сведений.

Все торговые знаки, упомянутые в настоящем издании, зарегистрированы. Случайное неправильное использование или пропуск торгового знака или названия его законного владельца не должно рассматриваться как нарушение прав собственности.

ISBN 1-928994-31-8 (англ.)
ISBN 5-98453-006-6 (АйТи)
ISBN 5-94074-258-0 (ДМК Пресс)

Copyright © 2001 by Syngress Publishing, Inc.
© Перевод на русский язык. Компания АйТи
© Оформление. ДМК Пресс
© Издание. Тетру

Содержание

Предисловие 19

Глава 1. Методология хакинга 21

Введение 22
- Терминология 23

Краткая история хакинга 23
- Хакинг телефонных систем 24
- Компьютерный хакинг 25

Мотивы хакера 28
- Сравнение этичного и злонамеренного хакинга 29
- Сотрудничество со специалистами по безопасности 30

Наиболее распространенные типы атак 31
- DoS/DDoS 31
- Вирусный хакинг 33
- Хищение 40

Распознавание угроз безопасности Web-приложений 44
- Скрытая манипуляция 44
- Искажение параметра 45
- Создание перекрестного сценария 45
- Переполнение буфера 45
- Заражение cookie-файла 46

Изучение хакерских методов для защиты от взломов 47

Резюме 50

Конспекты 50

Часто задаваемые вопросы 53

Как апплеты злоумышленника переносят вредоносный код:

Мобильный код для приложений в виде Java-апплетов, JavaScript и ActiveX-компонентов – это мощное орудие для передачи данных по Сети. Но это также и мощное орудие для распространения вредоносного кода. Вредоносные апплеты не саморазмножаются и не предназначены, например, для искажения данных, подобно вирусам. Но они часто становятся средством специфической атаки, направленной на хищение данных или приведение системы к краху.

Творческий подход к программированию:

- Осознайте возможность постороннего вмешательства в ваш код, ожидайте непредсказуемого!
- Сделайте свой код более лаконичным; часть кода, отвечающая за функциональность, должна быть минимальной.
- Проверка, проверка, проверка! Не надо полагаться только на себя и тем более замалчивать свои ошибки.

Глава 2. Как избежать «перемалывания кода» 55

Введение 56

Что означает «перемалыватель кода» 57

Следование правилам 60

Кодирование – творческий процесс 61

Разрешение на идею 63

Обеспечение безопасности «перемалывателем кода» 66

Кодирование в вакууме 67

Что предпочесть – функциональность или безопасность Web-приложений 69

Но мой код функционален! 73

Резюме 80

Конспекты 81

Часто задаваемые вопросы 82

Глава 3. Мобильный код – ваш враг 83

Введение 84

Осмысление рисков, связанных с мобильным кодом 85

Атаки на браузер 85

Атаки на почтового клиента 86

Вредоносные сценарии, или макросы 86

Идентификация основных форм мобильного кода 87

Макроязыки: Visual Basic for Applications 88

JavaScript 94

VBScript 98

Java-апплеты 101

ActiveX 105

Почтовые вложения и загруженные исполняемые файлы 109

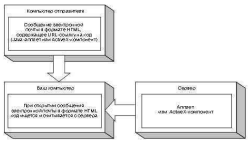
Защита от атак, использующих мобильный код 113

Приложения, обеспечивающие безопасность 113

Web-инструменты 118

Резюме 120

Как работает мобильный код в Java-апплетах и ActiveX-компонентах:



Мобильный код, содержащийся на Web-сервере

Конспекты 121
Часто задаваемые вопросы 122

Глава 4. Уязвимые CGI-сценарии 123

Введение 124
Что такое CGI-сценарий и для чего он применяется 124
Типовые использования CGI-сценариев 127
Когда следует обращаться к CGI 131
Проблемы хостинга CGI-сценария 132
Взлом слабых CGI-сценариев 133
Как создавать надежные CGI-сценарии 135
Команда индексного поиска 138
CGI-упаковщики 139
Языки для создания CGI-сценариев 143
Командный процессор UNIX 144
Perl 145
C/C ++ 145
Visual Basic 146
Преимущества использования CGI-сценариев 146
Правила создания безопасных CGI-сценариев 147
Хранение CGI-сценариев 151
Резюме 154
Конспекты 154
Часто задаваемые вопросы 157

Глава 5. Методы и инструменты хакинга 159

Введение 160
Цели хакера 161
Минимизация количества предупреждений 162
Расширение доступа 163
Убытки, убытки, убытки 166
Поменяться ролями 168
Пять стадий взлома 169
Создание схемы атаки 170
Составление плана реализации атаки 172

Инструментальные средства... Остерегайтесь ввода данных пользователем:

CGI-сценарии и программы обычно применяются, когда код разрешает пользовательский ввод, но часто данные, которые предлагает пользователь, не проверяются. Контроль информации, введенной пользователем, значительно сокращает шансы хакера на взлом вашей системы с помощью CGI-сценария.

Ознакомьтесь досконально с технологиями хакеров:

Вопрос: Что я должен делать, обнаружив «черный ход» в своем коде?
Ответ: Сначала нужно определить, действительно ли это «черный ход». Иногда некоторые части кода не имеют никаких аутентификационных данных и могут выполнять значимые операции, но тем не менее перед

их вызовом обеспечивается должная аутентификация. Если при тщательном анализе выявилось, что это «черный ход», пригласите сотрудника отдела безопасности, знающего данный язык программирования, и проведите аудит кода. Если сотрудник решит, что это «черный ход», то следует выяснить, был ли он создан преднамеренно или случайно.

Определение точки входа 173
Длительный и продолжающийся доступ 174
Атака 175
Социотехника 177
Критичная информация 177
Атака «черного хода» 184
Закодированный пароль «черного хода» 184
Эксплуатация слабостей кода или среды программирования 186
Продаваемые инструментальные средства 187
Hex-редакторы 187
Отладчики 189
Обратные ассемблеры 190
Резюме 193
Конспекты 193
Часто задаваемые вопросы 196

Как проводить эффективную трассировку выполнения программ:

- Трассировка выполнения программы от начала до конца занимает слишком много времени.
- Вам удастся сэкономить время, если вместо этого вы займетесь непосредственно разрешением известных проблем.
- Такой подход оставляет в стороне логику обработки приложений.

Глава 6. Аудит и обратная проверка кода 199
Введение 200
Эффективное трассирование программы 200
Аудит и проверка выбранных языков программирования 203
Проверка языка Java 204
Проверка языка JSP 204
Проверка языка ASP 204
Проверка языка SSI 205
Проверка языка Python 205
Проверка языка TCL 205
Проверка языка Perl 206
Проверка языка PHP 206
Проверка языка C/C++ 206
Проверка языка ColdFusion 207
Поиск уязвимостей 207
Поиск переполнения буфера 208
Получение данных от пользователя 208
Контроль пользовательского вывода 212

Контроль доступа/взаимодействия в файловой системе 216
Контроль внешней программы и выполнения кода 219
Проверка языка SQL/запросов к базе данных 221
Контроль организации сети и коммуникационных потоков 224
Обобщение полученной информации 225
Резюме 227
Конспекты 227
Часто задаваемые вопросы 228

Рассмотрите все аспекты модели безопасности Java:

- Загрузчики классов.
- Проверка байт-кода.
- Менеджеры безопасности.
- Электронные цифровые подписи.
- Аутентификация с применением сертификатов.
- Подпись JAR-файлов.
- Шифрование.

Глава 7. Безопасность Java-кода 229

Введение 230
Краткий обзор архитектуры безопасности Java 231
Модель безопасности Java 232
Sandbox 234
Как Java обеспечивает безопасность 238
Загрузчики классов 239
Верификатор байт-кода 242
Защищенные Java-домены 247
Потенциальные слабости Java 256
DoS-атака и атаки снижения эффективности служб 256
Сторонние троянские атаки 259
Кодирование функциональных и безопасных Java-апплетов 260
Дайджесты сообщения 261
Цифровые подписи 264
Аутентификация 270
Безопасность JAR с применением ЭЦП 278
Шифрование 281
Рекомендации Sun Microsystems по безопасности Java 288
Резюме 291
Конспекты 292
Часто задаваемые вопросы 293

Ущерб и защита... Отладка XSL:

Взаимодействие таблицы с XML-документом представляет собой довольно сложный процесс, и к сожалению, ошибки таблицы чаще всего являются криптографическими. Компания Microsoft предлагает XSL-отладчик, основанный на HTML, который предназначен для выполнения XSL. Вы сможете просмотреть и исходный код для его улучшения. Отладчик XSL доступен на сайте: http://msdn.microsoft.com/downloads/samples/internet/xml/sxl_debugger/default.asp.

Глава 8. Безопасность XML 295

Введение 296

Определение XML 296

Логическая структура 298

Элементы 299

XML- и XSL/DTD-документы 302

Использование XSL-шаблонов 302

Использование XSL-образцов 303

DTD 305

Создание Web-приложений с помощью XML 308

Риски, связанные с XML 311

Обеспечение конфиденциальности 312

Безопасность XML 313

XML-шифрование 314

Электронная цифровая подпись XML 319

Резюме 322

Конспекты 322

Часто задаваемые вопросы 324

Разрешайте выполнение ActiveX-компонента, опираясь на информацию в окне предупреждения:



Глава 9. Создание безопасных ActiveX-компонентов для работы в Internet 325

Введение 326

Угрозы, связанные с ActiveX 327

Как избежать типовых уязвимостей ActiveX 329

Уменьшение влияния уязвимостей ActiveX 332

Методология разработки безопасных ActiveX-компонентов 335

Настройка безопасности объектов 336

Безопасность ActiveX-компонентов 337

Подписание компонента 337

Применение Microsoft Authenticode 339

Маркировка компонента 341

Резюме 347

Конспекты 347

Часто задаваемые вопросы 349