

ВЗЛОМ

ПРИЕМЫ, ТРЮКИ И СЕКРЕТЫ ХАКЕРОВ

ВЕРСИЯ 2.0

Библиотека журнала

The logo for the journal 'Hacker' features a stylized 'H' symbol on the left, composed of three horizontal bars. To its right, the word 'HACKER' is written in a bold, white, sans-serif font.

ВЗЛОМ

ПРИЕМЫ, ТРЮКИ И СЕКРЕТЫ ХАКЕРОВ

ВЕРСИЯ 2.0

Санкт-Петербург
«БХВ-Петербург»

2022

УДК 004
ББК 32.973
В40

В40 Взлом. Приемы, трюки и секреты хакеров. Версия 2.0. — СПб.:
БХВ-Петербург, 2022. — 272 с.: ил. — (Библиотека журнала «Хакер»)
ISBN 978-5-9775-1227-5

В сборнике избранных статей из журнала «Хакер» описана технология инъекта шелл-кода в память KeePass с обходом антивирусов, атака ShadowCoerce на Active Directory, разобраны проблемы heap allocation и эксплуатация хипа уязвимого SOAP-сервера на Linux. Рассказывается о способах взлома протекторов Themida, Obsidium, .NET Reactor, Java-приложений с помощью dirtyJOE, программ fat binary для macOS с поддержкой нескольких архитектур. Даны примеры обхода Raw Security и написания DDoS-утилиты для Windows, взлома компьютерной игры и написания для нее трейнера на языке C++. Описаны приемы тестирования протоколов динамической маршрутизации OSPF и EIGRP, а также протокола DTP. Подробно рассмотрена уязвимость Log4Shell и приведены примеры ее эксплуатации.

Для читателей, интересующихся информационной безопасностью

УДК 004
ББК 32.973

Группа подготовки издания:

Руководитель проекта	<i>Павел Шалин</i>
Зав. редакцией	<i>Людмила Гауль</i>
Редактор	<i>Ярослава Платонова</i>
Компьютерная верстка	<i>Натальи Смирновой</i>
Дизайн обложки	<i>Карины Соловьевой</i>

Подписано в печать 02.06.22.
Формат 70×100^{1/16}. Печать офсетная. Усл. печ. л. 21,93.
Тираж 1200 экз. Заказ № 4220.
"БХВ-Петербург", 191036, Санкт-Петербург, Гончарная ул., 20.
Отпечатано с готового оригинал-макета
ООО "Принт-М", 142300, М.О., г. Чехов, ул. Полиграфистов, д. 1

ISBN 978-5-9775-1227-5

© ИП Югай А.О., 2022
© Оформление. ООО "БХВ-Петербург", ООО "БХВ", 2022

Содержание

Предисловие	7
Вызов мастеру ключей. Инжектим шелл-код в память KeePass, обойдя антивирус	10
Предыстория	10
Потушить AV	11
Получить сессию C2	13
Перепать инструмент	14
Классическая инъекция шелл-кода	14
Введение в D/Invoke	20
DynamicAPIInvoke без D/Invoke	21
DynamicAPIInvoke с помощью D/Invoke	27
Зачем системные вызовы?	33
GetSyscallStub с помощью D/Invoke	35
Модификация KeeThief	42
Подготовка	42
Апгрейд функции ReadProcessMemory	43
Время для теста!	46
Выводы	47
ShadowCoerce. Как работает новая атака на Active Directory	48
PetitPotam и PrinterBug	48
Что такое VSS	49
Стенд	49
Как работает ShadowCoerce	50
Эксплуатация	53
Выводы	56
Круче кучи! Разбираем в подробностях проблемы heap allocation	58
Основы GDB	58
Структура чанков	59
Арена	60
Флаги	61
Bins	61

Тестовая программа	63
Практика	63
Fast bin Dup	69
Что еще почитать про кучу	73
WinAFL на практике. Учимся работать фаззером и искать дыры в софте	74
Требования к функции	75
Компиляция WinAFL	75
Поиск подходящей цели для фаззинга	76
Поиск функции для фаззинга внутри программы	77
Аргументы WinAFL, подводные камни	84
Прокачка WinAFL — добавляем словарь	85
Особенности WinAFL	86
Побочные эффекты	86
Дебаг-режим	86
Эмуляция работы WinAFL	86
Стабильность	87
Набор входных файлов	87
Отучаем программу ругаться	87
Неядерный реактор. Взламываем протектор .NET Reactor	88
Фемида дремлет. Как работает обход защиты Themida	95
Сны Фемиды. Ломаем виртуальную машину Themida	102
Грязный Джо. Взламываем Java-приложения с помощью dirtyJOE	110
Obsidium fatality. Обходим триальную защиту популярного протектора	122
В итоге	129
Липосакция для fat binary. Ломаем программу для macOS с поддержкой нескольких архитектур	130
Немного теории	130
Intel	132
ARM	134
Патчим плагин	136
Разборки на куче. Эксплуатируем хип уязвимого SOAP-сервера на Linux	138
Реверс-инжиниринг	139
handleCommand	139
parseArray	144
executeCommand	148
deleteNote	150
editNote	151

newNote	152
show	154
Итоги реверса	154
Анализируем примитивы	154
UAF (show после delete)	154
Heap overflow	155
Неочевидный UAF и tcachebins	156
Собираем эксплоит	159
Запускаем эксплоит	160
Выводы	160
Routing nightmare. Как пентестить протоколы динамической маршрутизации OSPF и EIGRP	164
Проблематика, импакт и вооружение	164
Протокол OSPF	164
Протокол EIGRP	166
Импакт	167
Вооружение с FRRouting	168
Настройка FRRouting	168
Виртуальная лаборатория	169
Инъекция маршрутов и перехват трафика в домене OSPF	171
Инъекция маршрутов и переполнение таблицы маршрутизации в домене EIGRP	173
Меры предотвращения атак на домены маршрутизации	176
Выводы	177
Разруливаем DTP. Как взломать протокол DTP и совершить побег в другую сеть VLAN	178
Как это работает	178
Уязвимость	180
Виртуальная лаборатория	181
Кастомная эксплуатация уязвимости ++без использования++ Yersinia	182
Эксплуатация	185
Побег в другую сеть VLAN	187
Защита	189
Вывод	189
DDoS с усилением. Обходим Raw Security и пишем DDoS-утилиту для Windows	190
Ищем уязвимые серверы	193
Разработка	194
Функция выбора интерфейса, из которого будут поступать пакеты	195
Функции формирования UDP-пакета	196
Формирование пакета	200
Отправка пакета	200
Заключение	201

Чит своими руками. Вскрываем компьютерную игру и пишем трейнер на C++	202
Выбор игры	202
Поиск значений	202
Что такое статический адрес	205
Поиск показателей здоровья	206
Поиск статического адреса для индикатора здоровья	210
Поиск значения числа патронов	213
Поиск статического адреса для ammo	213
Проверка полученного статического адреса	218
Проверка для HP	218
Проверка для ammo	219
Как будет выглядеть наш указатель в C++	220
Написание трейнера	220
Injector	221
DLL	222
Модуль обратных вызовов	229
Модуль работы с памятью	229
Проверка работоспособности	232
Выводы	232
Log4HELL! Разбираем Log4Shell во всех подробностях	233
Log4Shell	233
Патчи для патчей	234
Майнеры, DDoS и вымогатели	235
Защита	237
Списки уязвимых	237
Как работает уязвимость	238
Как нашли уязвимость	238
Стенд	240
build.gradle	240
src/main/java/logger/Test.java	240
build.gradle	241
Детали уязвимости	242
RCE через Log4j	250
Эксплуатация Log4j в Spring Boot RCE на Java версии выше 8u19	251
Не RCE единым	253
Манипуляции с пейлоадом и обходы WAF	256
Патчи и их обходы	259
Выводы	264
«Хакер»: безопасность, разработка, DevOps	265
Предметный указатель	269