SYNGRESS®

4 FREE BOOKLETS
YOUR SOLUTIONS MEMBERSHIP

Buffer Overflow Attacks

DETECT, EXPLOIT, PREVENT

Will the Code You Write Today Headline Tomorrow's BugTraq Mail List?

- Includes Numbered-by-Line Exploit Code Examples that Illustrate the Differences Between Stack Overflows, Heap Corruption, and Format String Bugs
- Provides Case Studies for Most Major Platforms and Environments, Including Windows, FreeBSD, FrontPage, and Linux
- Avoid Worm or Custom Exploits by Analyzing your Source Code to Detect Buffer Overflow Vulnerabilities

James C. Foster
Vitaly Osipov
Nish Bhalla
Niels Heinen

FOREWORD BY DAVE AITEL
FOUNDER AND CEO
IMMUNITY, INC.

Register for Free Membership to

solutions@syngress.com

Over the last few years, Syngress has published many best-selling and critically acclaimed books, including Tom Shinder's *Configuring ISA Server 2000*, Brian Caswell and Jay Beale's *Snort 2.0 Intrusion Detection*, and Angela Orebaugh and Gilbert Ramirez's *Ethereal Packet Sniffing*. One of the reasons for the success of these books has been our unique **solutions@syngress.com** program. Through this site, we've been able to provide readers a real time extension to the printed book.

As a registered owner of this book, you will qualify for free access to our members-only solutions@syngress.com program. Once you have registered, you will enjoy several benefits, including:

- Four downloadable e-booklets on topics related to the book. Each booklet is approximately 20-30 pages in Adobe PDF format. They have been selected by our editors from other best-selling Syngress books as providing topic coverage that is directly related to the coverage in this book.
- A comprehensive FAQ page that consolidates all of the key points of this book into an easy to search web page, providing you with the concise, easy to access data you need to perform your job.
- A "From the Author" Forum that allows the authors of this book to post timely updates links to related sites, or additional topic coverage that may have been requested by readers.

Just visit us at **www.syngress.com/solutions** and follow the simple registration process. You will need to have this book with you when you register.

Thank you for giving us the opportunity to serve your needs. And be sure to let us know if there is anything else we can do to make your job easier.

SYNGRESS®

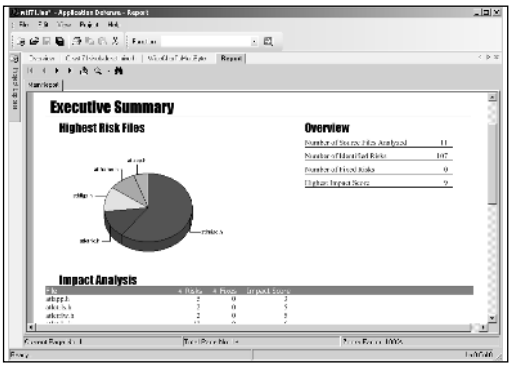
Application Defense
www.applicationdefense.com

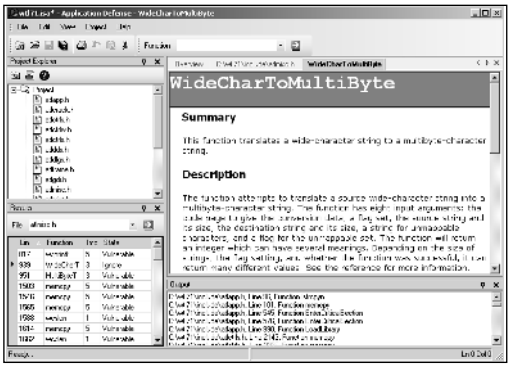
Application Defense Specials

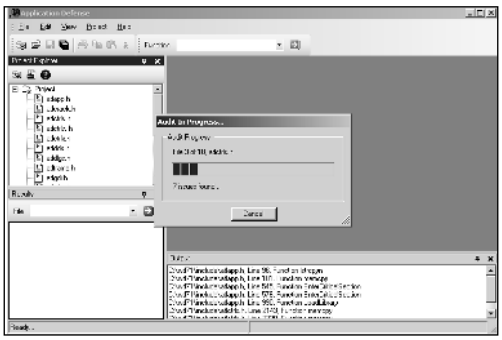
- Free Software with Purchase of Application Security Services Program
- $1,000 Enterprise Language Special Until February 2005 with Proof of Purchase of Ultimate DeskRef.

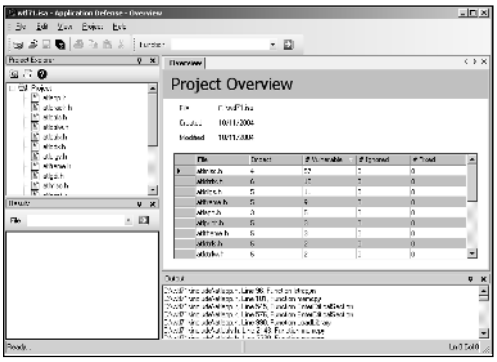
Business Benefits

- Application Defense Developer Edition, strives to educate individual developers on proper secure programming techniques during the development cycle, thereby saving thousands in post-development consulting
- Developmental education approach on secure development strengthens your business at the core, its people
- Executive-level reporting allows your development team to visually depict trending improvements, vulnerability remediation, and high-risk segments of code
- Distributed Software Architecture permits development teams to review their code centrally by a QA or Auditing team or individually by the developers
- Industry-best multi-language support permits organizations to manage all their software development needs with one application









Application Defense Technology Features:

- Industry leading analysis engine can parse and examine entire software code base in under a minute
- Executive, technical, and trending reports allow information to be displayed for all audiences
- Flexible XML output allows easy integration with other enterprise applications
- Unique IDE allows you to update results in real-time or in batches to code base – no need to recreate code in multiple locations!
- Custom developer code is analyzed by proprietary artificial intelligence engine
- Project file storage allows developers to save analysis results for later review or to save for continued analysis
- Real-time bug tracking system
- Interactive software interface allows developers to make security decisions during analysis
- Able to input Visual Studio Project files
- Customizable reports allow you to specify company name, application, auditor, and more…

SYNGRESS®

Buffer Overflow Attacks

DETECT, EXPLOIT, PREVENT

James C. Foster
Vitaly Osipov
Nish Bhalla
Niels Heinen

FOREWORD BY DAVE AITEL
FOUNDER AND CEO
IMMUNITY, INC.

KEY	SERIAL NUMBER
001	HJIRTCV764
002	PO9873D5FG
003	829KM8NJH2
004	HJBC43288N
005	CVPLQ6WQ23
006	VBP965T5T5
007	HJJJ863WD3E
008	2987GVTWMK
009	629MP5SDJT
010	IMWQ295T6T

PUBLISHED BY
Syngress Publishing, Inc.
800 Hingham Street
Rockland, MA 02370

Buffer Overflow Attacks: Detect, Exploit, Prevent

Printed in the United States of America
1 2 3 4 5 6 7 8 9 0
ISBN: **1-932266-67-4**

Publisher: Andrew Williams
Acquisitions Editor: Jaime Quigley
Technical Editor: James C. Foster
Cover Designer: Michael Kavish
Page Layout and Art: Patricia Lupien
Copy Editor: Mike McGee
Indexer: Richard Carlson

Acknowledgments

Syngress would like to acknowledge the following people for their kindness and support in making this book possible.

Syngress books are now distributed in the United States and Canada by O'Reilly Media, Inc. The enthusiasm and work ethic at O'Reilly is incredible and we would like to thank everyone there for their time and efforts to bring Syngress books to market: Tim O'Reilly, Laura Baldwin, Mark Brokering, Mike Leonard, Donna Selenko, Bonnie Sheehan, Cindy Davis, Grant Kikkert, Opol Matsutaro, Steve Hazelwood, Mark Wilson, Rick Brown, Leslie Becker, Jill Lothrop, Tim Hinton, Kyle Hart, Sara Winge, C. J. Rayhill, Peter Pardo, Leslie Crandell, Valerie Dow, Regina Aggio, Pascal Honscher, Preston Paull, Susan Thompson, Bruce Stewart, Laura Schmier, Sue Willing, Mark Jacobsen, Betsy Waliszewski, Dawn Mann, Kathryn Barrett, John Chodacki, Rob Bullington, and Aileen Berg.

The incredibly hard working team at Elsevier Science, including Jonathan Bunkell, Ian Seager, Duncan Enright, David Burton, Rosanna Ramacciotti, Robert Fairbrother, Miguel Sanchez, Klaus Beran, Emma Wyatt, Rosie Moss, Chris Hossack, Mark Hunt, and Krista Leppiko, for making certain that our vision remains worldwide in scope.

David Buckland, Marie Chieng, Lucy Chong, Leslie Lim, Audrey Gan, Pang Ai Hua, and Joseph Chan of STP Distributors for the enthusiasm with which they receive our books.

Kwon Sung June at Acorn Publishing for his support.

David Scott, Tricia Wilden, Marilla Burgess, Annette Scott, Andrew Swaffer, Stephen O'Donoghue, Bec Lowe, and Mark Langley of Woodslane for distributing our books throughout Australia, New Zealand, Papua New Guinea, Fiji Tonga, Solomon Islands, and the Cook Islands.

Winston Lim of Global Publishing for his help and support with distribution of Syngress books in the Philippines.

For the men and woman who proudly
"serve in silence,"
dedicating their lives to
Mission, Workmate, and Country.

Lead Author

James C. Foster, Fellow, is the Deputy Director of Global Security Solution Development for Computer Sciences Corporation. Foster is responsible for directing and managing the vision, technology, and operational design of all security services within CSC. Prior to joining CSC, Foster was the Director of Research and Development for Foundstone Inc., and was responsible for all aspects of product, consulting, and corporate R&D initiatives. Prior to joining Foundstone, Foster was a Senior Advisor and Research Scientist with Guardent Inc., and an editor at Information Security Magazine, subsequent to working as an Information Security and Research Specialist for the Department of Defense. Foster's core competencies include high-tech management, international software development and expansion, web-based application security, cryptography, protocol analysis, and search algorithm technology. Foster has conducted numerous code reviews for commercial OS components, Win32 application assessments, and reviews of commercial and government cryptography implementations.

Foster is a seasoned speaker and has presented throughout North America at conferences, technology forums, security summits, and research symposiums, including the Microsoft Security Summit, BlackHat, MIT Wireless Research Forum, SANS, MilCon, TechGov, InfoSec World 2001, and the Thomson Security Conference. He frequently comments on pertinent security issues and has been sited in USA Today, Information Security Magazine, Baseline, Computerworld, Secure Computing, and the MIT Technologist. Foster holds degrees in Business Administration, Software Engineering, and Management of Information Systems. He has attended the Yale School of Business, Harvard University, and the University of Maryland. He is currently a Fellow at the University of Pennsylvania's Wharton School of Business.

Foster has written many commercial and educational papers. He has also contributed to several books, including: *Snort 2.0, Snort 2.1 2nd Edition, Hacking Exposed 4th Ed and 5th Edition, Special Ops Security, Anti-Hacker Toolkit 2nd Ed, Advanced Intrusion Detection, Hacking the Code, Anti-Spam Toolkit, Programmer's Ultimate Security DeskRef, Google for Penetration Testers, Buffer Overflow Attacks,* and *Sockets/Porting/and Shellcode*.

Contributing Authors

Vitaly Osipov (CISSP, CISA) is currently managing intrusion detection systems for a Big 5 global investment bank in Sydney, Australia. He previously worked as a security specialist for several European companies in Dublin, Prague and Moscow. Vitaly has co-authored books on firewalls, IDS, and security including *Special Ops: Host and Network Security for Microsoft, UNIX and Oracle* (ISBN: 1-931836-69-8) and *Snort 2.0: Intrusion Detection* (ISBN: 1-931836-74-4). Vitaly's background includes a long history of designing and implementing information security systems for financial institutions, ISPs, telecoms, and consultancies. He is currently studying for his second postgraduate degree in mathematics.

Niels Heinen is a security researcher at a European security firm. Niels has researched exploitation techniques and specializes in writing position independent assembly code used for changing program execution flows. While the main focus of his research is Intel Systems, he's also experienced with MIPS, HPPA and PIC processors. Niels, enjoys writing his own polymorphic exploits, wardrive scanners and OS fingerprint tools. His day-to-day job involves in-depth analysis of security products.

Nishchal Bhalla is a specialist in product testing, code reviews and web application testing. He is the lead consultant at Security Compass, providing consulting services for major software corporations & Fortune 500 companies. He's a contributing author to *Windows XP Professional Security and Hack Notes*. Prior to joining Security Compass, Nish worked for Foundstone, TD Waterhouse, Axa Group and Lucent. He holds a master's degree in parallel processing from Sheffield University, is a post-graduate in finance from Strathclyde University, and received a bachelor's degree in commerce from Bangalore University.

Additional Area Experts

Marshall Beddoe is a Research Scientist at McAfee, and conducts extensive research in passive network mapping, remote promiscuous detection, OS fingerprinting, FreeBSD internals, and new exploitation techniques. Marshall has spoken at such security conferences as Black Hat, Defcon, and Toorcon.

Tony Bettini leads the McAfee Foundstone R&D team and has worked for other security firms, including Foundstone, Guardent, and Bindview. He specializes in Windows security and vulnerability detection, as well as programs in Assembly, C, and various others. Tony has identified new vulnerabilities in PGP, ISS Scanner, Microsoft Windows XP, and Winamp.

Author's Acknowledgements

Most importantly, I'd like to thank my family for continuously believing in me and my ambitious goals. Mom, Dad, Steve – to you all I am grateful.

I'd like to thank everyone who helped contribute to this book, especially Vitaly (you rock!), Nish, Niels, Marshall, Tony, Dave Aitel, Johnny Long, Conrad Smith, and last but certainly not least, Stuart McClure. You guys' talent will no doubt ensure the lasting success of this book.

Sincere thanks go out to the true professionals at the software security vendors who helped make Chapter 9 a success, specifically Fortify Software's Brian Chess Ph.D, Chris Prevost, and Steve Garrity; Ounce Labs' Robert Gottlieb, John Peyton, Ellen Sinett, and Chris McClean; and Secure Software's John Viega, Dale R. Gardner, and Joel Greenberg.

An additional thank you goes out to Computer Sciences Corporation for allowing this publication to take place. Reg – you are still the man! Additional well-deserved thanks go out to Chris, Jason, Ron, Jen, and Mary.

Last but certainly not least, I'd like to thank the Syngress Publishing team. Jaime, thanks for putting up with "Foster" content and making sure this book stayed on schedule; you're an outstanding editor. Andrew – your vision is what spawned this book so thank you and kudos! Syngress continues to be my publishing company of choice.

-James Foster
December, 2004

Contents

Foreword

I can recall with crystal clarity three times relating to buffer overflows. I remember the first moments of when I used a buffer overflow in the wild and it worked (imapd for Linux). I remember the first buffer overflow that I found and exploited all by myself (a Linux local). And I remember the first time I got into a host by writing a buffer overflow.

When most people read Aleph1's seminal paper on buffer overflows "Buffer overflows for fun and profit," they think largely on the "profit." Someone skilled in the art of writing buffer overflows can make $90-120K USD a year as a consultant for any of the large companies in the market.

Getting that skill, on the other hand, has one major roadblock: most people think that they can "learn" it. Many knacks in the IT world can be learned. Once you realize them, they're yours to keep. But writing a buffer overflow is not like that. It's more like weight lifting, or judo. You can learn the basics from a book, or perhaps a short class, but the landscape you work in changes constantly. Microsoft is adding new protections to their code every day, as hackers are constantly finding new ways to make exploits more reliable, and finding ways to develop new types of bugs. Three months after you stop writing buffer overflows, your skills are ancient history. The hard part about writing buffer overflows is staring an endless treadmill of work in the face.

The tools you once used to use to write overflows are changing, too. It used to be something you could do alone, with nothing but a cracked copy of Softice or GDB. These days, Immunity invests heavily into infrastructure in order to write even a simple buffer overflow. For example, we have a proprietary debugger all our own, which allows us to query and script running programs. We have special purpose compilers which allow us to create and tune our shellcode to be exactly what the vulnerability needs. We purchase or create reverse engineering tools specialized for various problems. We have complete mySQL and SSL libraries written entirely in Python. A moderately complex exploit will involve our entire team working in unison on various parts of the problem.

And of course, afterwards, there's the inevitable paper on each difficult exploit, which must be written to share the knowledge gained with the difficult exploits within the team. It's no longer an individual sport.

The most powerful of buffer overflows are never going to be in worms. Custom exploits are what gets you in, and what keeps you in. If you see a truly top notch hacker going after someone, he'll have a complete mirror of the target hosts' environments, all for one purpose: To create a buffer overflow he only plans on using once.

There are levels in the game. James Foster will take you through basic training, giving you the broad view and skills you need to figure out what you want to specialize in. From there, you can walk away into management, confident that although you're not on the front line, you at least know how the fundamentals and can thereby make

informed decisions. Or you can dedicate yourself to the work, commit to the code, and develop the craft. If this is your choice, then I have a few maxims for you:

- Never be afraid. Microsoft has an entire army of marketing people telling you how hard it is to find and write buffer overflows against their newest software. My strategy is to fantasize about what I'm going to do with the exploit once it's working – it keeps me going. Writing exploits is going to require mastery of a lot of difficult and boring technology. HP-UX uses wacky quadrant memory access. Irix has inane caches. And learning the thousands of assembly languages this requires is no small feat. If you think you can do it, you can.
- Don't take yourself too seriously. Realize that no matter how good you get, some fifteen year-old in Sweden is dedicating twenty hours a day to be better than you. It's not a competition, and you'll burn out quick if you play the game as such.
- Find some friends. Like weight lifting or judo, this is not a skill you're going to keep up with if you do it all yourself. And you need people to tell you where you're weak.

Regardless of your goals, use this book as a worksheet, rather than a novel. Don't read it without a computer next to you. A book on buffer overflows can't deliver that hacker high with knowledge alone. Work along with the chapters and you'll find that you can't stop fiddling with your exploit when it's not working right. You'll give up food, sleep, or money to make it that much cleaner.

My belief is this: In the end, an exploit is a complex statement of truth. What you're saying is "This is possible." And saying it truthfully makes it beautiful.

Someday, I hope to appreciate your code as art.

—Dave Aitel
Founder, CEO
Immunity, Inc.

Part 1

Expanding on Buffer Overflows