

Immunity and Cyber Defense Institute present a custom class:

ADVANCED EXPLOIT DEVELOPMENT: WINDOWS HEAP OVERFLOWS

Schedule:

February 17, 2009 to February 20, 2009

Location:

Tokyo JAPAN

Course Fee:

JPY498,850(tax inclusive)

Max Capacity:

20 attendees

Course Style:

Exercise Driven

Language: English & Japanese

Course Instructor

Nicolas Waisman:

Nicolas Waisman is a Senior Security Researcher and Regional Manager at Immunity, Inc, which he joined in 2004. Mr. Waisman is one of the driving forces behind the CANVAS attack framework. His security research is concentrated on win32 vulnerability development, specifically reliable heap exploitation techniques. Mr. Waisman has spoken on and taught these techniques to government and private sector researchers in Japan, Singapore, Europe, North and South America. Mr. Waisman is also the Manager and main point of contact for Immunity South America, where he is leading Immunity's newest project, the vuln-dev oriented Immunity Debugger.

COURSE AGENDA

Nowadays Information Technology Security has two aspects. While most corporations tend to stress the defensive one, only a few initiate themselves to its offensive aspect, conferring them an unquestionable advantage. Taking control of computers remotely by relying on stack overflows is almost impossible nowadays due to Windows protection mechanisms. The modern security professional needs to improve their capabilities by adding the latest technique in the exploitation field: heap overflow exploitation. Immunity, Inc., a market leader in exploit solutions and the only provider of heap exploitation training, will share its experience in this domain during a four day course. Immunity, Inc. will provide an introduction of the last cutting-edge techniques and technology used in the development of offensive tools using heap exploitation technology. Attendees are expected to obtain an understanding of the following:

- Terminology of offensive security
- Understanding of stakes
- Fundamentals of the Windows dynamic memory allocator (heap)
- Basic knowledge of internal structures
- Basic knowledge of the main algorithms
- Effective use of CANVAS for exploit development
- Concepts and basics of heap crash analysis
- Concepts and basics of heap exploitation
- Heap exploitation techniques: Write4, Coalescing and Write8
- Effective use of Immunity Debugger for heap exploit creation
- Introduction to the concept of "heap layout"
- Basics of reverse engineering
- Advanced reversing engineering for finding memory leaks
- Heap shellcodes
- Advanced exploitation techniques: multiple write4 and lookaside
- New protections introduced in Windows 2003: safe unlinking
- Concept and basics of exploiting Windows 2003 Heaps
- Introduction to Vista heap and protection mechanisms
- Exploiting the Vista heap

The attendees will test and apply all those techniques in a lab environment, effectively compromising virtual machines under the guidance of Immunity, Inc. researchers.

Who Should Attend the Class

- Security Engineers
- Security Professionals
- Network Engineers
- Military
- Law Enforcement

Prerequisites

Knowledge

Each attendee is expected to have basic knowledge in programming, preferably Python and assembly, networking, basic Information Technology security, and stack overflows.

Hardware

Each attendee will be given a laptop running a recent version of Linux or Windows. Each laptop will have VMware installed, or an equivalent virtualization software, as well as three guest OSes installed:

- a Windows 2000 Workstation SP4
- a Windows 2003 SP1
- a Windows Vista

Course Syllabus

Day 1: Basics

Introduction: Immunity, Inc. instructors will introduce themselves as well as the company. Attendees will introduce themselves, and underline an aspect of the class they are particularly interested in.

Knowing the basics: Immunity, Inc. instructors will go over the various terms that will be used during the class as well as the concepts involved. The state of the art of exploitation will be presented, through various real-life examples. We will introduce Python briefly, and review x86 assembly basics.

CANVAS framework: We will introduce the CANVAS framework. We will present VisualSploit, as visual tool for exploit creation, the CANVAS structure and functionalities. We will explain how a basic CANVAS exploit is organized.

Immunity Debugger: An introduction to Windows debugging will be given with Immunity Debugger - a debugger designed for exploit development. We will also cover relevant Immunity Debugger extended features.

Windows Heap Introduction: An introduction to how the Windows 2000 heap works, including the internal structure and the main algorithms. Attendees will build on theory with a hands-on exercise.

Windows Heap Exploitation: An introduction to the common techniques used for the process of heap exploitation. Attendees will attack a specially built network service and make it crash. They will debug the target, understand the heap corruption and try to obtain a memory write.

Day 2: Simple Exploit Development

Windows Heap Exploitation: We will go over most of the problems that can appear when exploiting heap overflows, understanding all the alternatives to obtain a memory write. Those include forward and backward coalescation and lookaside unlinking.

Heap Layout Crafting: Introduction to the techniques used to craft the heap layout, including reverse engineering techniques used for finding soft and hard memory leaks. Attendees will use given examples to apply this new concept.

Advanced Heap Exploitation: Attendees will learn by example the different techniques to obtain shellcode execution when exploiting a Windows service bug. Techniques such as write8 and lookaside overwrite will be explained.

Exploit reliability: Reliability is a common issue in publicly available exploits. Some time will be spent explaining how one can make an exploit work against a larger variety of targets, including different versions of Windows and different localizations. The case study of MS06-040 will be analyzed to present a very specific but reliable way of writing a portable exploit.

Heap Overflows and Function Pointers: Various techniques will be explained for identifying function pointers that help transform a memory overwrite into shellcode execution. Attendees will use their recently gained debugging knowledge to discover functions pointers.

Shellcoding: Specially crafted shellcode will be explained and used by the attendees to obtain remote access to a target. Concepts such as heap injection, forkloading and SeDebugPrivilege permissions will be explained.

Day 3: Advanced Exploit Development: Windows 2003

Protection measures: Microsoft has implemented some protections in Windows 2003 versions in order to reduce the exploitability of bugs. We will discuss mechanisms such as non-executable pages, heap cookie, safe unlink, DEEP and how to overcome them.

Windows 2003 bypassing: Cutting edge techniques will be explained to understand and bypass the last security mechanisms. Techniques such as lookaside overwrite will be introduced. The case study will present a generic way to exploit heap overflows reliably on this operating system.

Exploiting DEP: Immunity, Inc. will discuss the problem of DEP protection mechanisms and how to overcome them. The concept of "Ret into the win32 API" will be explained and attendees will apply it to the case of study.

Day 4: The Vista Heap

Basic Vista Heap Concepts: Microsoft has rewritten the whole heap implementation, taking it to a new protection level and making exploitation very difficult. We will discuss non-public details regarding how the new Vista heap algorithm works. Attendees will work on practical examples with the debugger to understand the concepts.

Windows Vista Exploitation: Public and non-public techniques will be presented for exploiting the Vista heap. Attendees are expected to apply the techniques learned during the four days to bypass Vista protections and exploit a specially built network service.

To attend:

mail: sales [at] cyberdefense.jp

phone: +81-3-5209-4335 (call Mr.SAKAI or Mr.IINUMA)