RSA Conference 2017

Learning Labs

Hands-On Exploit Development for Beginners

Post-Conference Summary

Sam Bowne

Devin Duffy-Halseth

Dylan Smith

Table of Contents

Introduction

Thanks for attending!

We are grateful and honored that you chose to spend some of your valuable time attending our Learning Lab, *Hands-On Exploit Development for Beginners*, at RSA Conference 2017. We hope you found it engaging and educational.

This summary is designed to recap the projects you did, and the lessons they provide.

Sam Bowne, Devin Duffy-Halseth, and Dylan Smith

Command Injection Projects

Ping Form

Intended Functionality

An HTML form allows the user to enter a domain name. A PHP script then builds a Linux shell command using this input to ping the remote host, verifying Internet connectivity.

Vulnerability

By adding special characters such as ; or &&, an attacker can execute arbitrary commands on the host after the ping command. This grants the attacker Remote Code Execution with the privileges of the Web server (www-data).

Solution

Using PHP to construct a shell command is a sloppy programming practice, since the command must be parsed by the shell interpreter. Mixing languages like this should be avoided. If it must be performed, the developer has an obligation to consider the vulnerabilities of both environments and filter out all possible attack strings.

Buffer Overflow

Intended Functionality

An HTML form allows the user to enter a name, which is then echoed back. A second variable in the code contains a harmless shell command that is executed.

Vulnerability

By entering a long name, the second string variable is overwritten and can be used to execute arbitrary commands.

Solution

The primary defense is to take care not to allow the user to enter more characters than the name buffer can store. Reviewing source code should detect this sort of error. However, conventional buffer overflow defenses such as canaries, Data Execution Prevention, and Address Layout Space Randomization will not protect code like this, because the overflow does not fill the entire stack frame.

ImageMagick

Intended Functionality

An HTML form allows the user to upload an image, from which a thumbnail version is constructed.

Vulnerability

The ImageMagick library is an old version, with a known command injection vulnerability. By uploading a crafted image file containing text, commands within the text are executed on the server as shell commands. The root cause is that the ImageMagic code uses text from the file to build a command using "curl" and then executes it as a shell command.

Solution

Like the Ping Form, building a shell command with user input is a risky practice. Developers who choose to do so must thoroughly filter the user input to remove special characters, or, better, allow only known good characters.

SQL Injection Projects

Intended Functionality

An HTML form allows the user to enter a user name and search a database for that user. A PHP script builds a SQL query using that name.

Vulnerability

By adding an apostrophe followed by SQL commands such as SELECT and UNION, an attacker can execute additional SQL commands on the host. This can be used to read or write files on the server, and to upload a PHP file which allows the attacker to execute arbitrary shell commands.

Solution

A weak defense is to filter special characters out of the name variable. A much stronger defense is to use parameterized queries which treat the user input as a data object, which cannot be misinterpreted by a parser as executable code.

Binary Exploitation Projects

For More Advanced Participants

Topics Covered

The projects above all used Code Injection, the top web application vulnerability according to OWASP. The remaining projects guide participant through the more complex process of exploiting code with memory corruption vulnerabilities, using a debugger to control the processor at the binary level.

The specific projects included in this workshop demonstrated thee techniques:

Adding a Trojan to an EXE file with Immunity

Very Simple Heap Overflow

Linux Buffer Overflow (32-bit) without shellcode, and with shellcode

Linux Buffer Overflow (64-bit)

Further Resources

All the projects are available at this URL, along with an entire Exploit Development course with slide decks and video lectures:

https://samsclass.info/124/RSA2017-ExploitDev.shtml

The classic reference for binary exploitation is *The Shellcoder's Handbook: Discovering and Exploiting Security Holes*, by Chris Anley, John Heasman, Felix Lindner, and Gerardo Richarte.

RSAConference2017
San Francisco | February 13 – 17 | Moscone Center

SESSION ID: LAB1-W12

Hands-On Exploit Development for Beginners

Sam Bowne
Instructor
City College San Francisco
@sambowne

Devin Duffy
Student
City College San Francisco

Dylan James Smith
Student
City College San Francisco
@heydylanjames