

Offensive Security

Advanced Web Attacks and Exploitation

v. 1.0



Mati Aharoni
Devon Kearns



Course Overview

The days of porous network perimeters are fading fast as services become more resilient and harder to exploit. In order to penetrate today's modern networks, a new approach is required. In order to gain that initial critical foothold in a network, penetration testers must be fluent in the art of exploiting front-facing web applications. Offensive Security's Advanced Web Attacks and Exploitation will take you far beyond the simple basics of SQL injection and bring you deep into the realm of web application penetration testing.

From mind-bending XSS attacks, to exploiting race conditions, to advanced SQL injection attacks, Advanced Web Attacks and Exploitation will broaden your knowledge of web application hacking and help you identify and circumvent various protection mechanisms in use on the web today.

Course Description

Advanced Web Attacks and Exploitation is NOT an entry-level course. The pace of learning is fast and furious - students are expected to have a solid understanding of how to perform basic web application attacks, at a minimum. This class is aimed at penetration testers and security auditors who need to take their web application penetration testing skills to a new level.

It is assumed that the student already has a medium understanding of the underlying protocols and technologies involved in testing web applications such as the HTTP protocol, SSL communications, and the usage of various browser plugins and proxies. A basic familiarity with web based programming languages such as php, javascript and mysql will also prove helpful.



Course Outline

1. Atmail Mail Server Appliance Case Study – CVE-2012-2593

1.1 Getting Started

1.2 Web Related Attack Vectors

1.2.1 Impact of XSS Attacks

1.2.2 Types of XSS Attacks

1.2.3 XSS Vulnerability Discovery

1.3 Attack Implementation

1.3.1 Exercise: Atmail - Document Dookie

1.4 Stealing Cookies and Hijacking Authenticated Sessions

1.4.1 Exercise: Atmail – All your Email is Belong to Us

1.5 Cross Site Request Forgery 101

1.5.1 Types of CSRF Attacks

1.5.2 CSRF Vulnerability Discovery

1.6 Better Email Snooping Through CSRF

1.6.1 Exercise: Atmail – All Your Email is Forwarded to Us

1.7 Research, Research, Research

1.7.1 Exercise: Atmail - uid=3000(atmail) groups=3000(atmail)

1.8 From XSS to Server Compromise

1.8.1 The Atmail Attack Plan

1.9 Further Reading

2. X-Cart Shopping Cart Case Study – CVE-2012-2570

2.1 Getting Started

2.1.1 XSS Filter Evasion

2.1.2 Exercise: Revenge of alert(“XSS”)

2.2 Getting the Lay of the Land

2.3 Building the XSS Payload

2.4 Exploiting the XSS Vulnerability

2.4.2 Exercise: XSS Filter Bypassing

2.5 Further Reading

3. SolarWinds Orion Case Study - CVE-2012-2577

3.1 Getting Started

3.2 Web Related Attack Vectors

3.3 View State Stuff

3.4 Attack Implementation

3.4.1 Exercise: Alert(“SNMP Rules, Always”)

3.5 SolarWinds Orion XSS: Now What?

3.5.1 Exercise: More Than Meets the Eye

3.6 Trying to Add a User

3.6.1 Exercise: Hopeless Quest

3.7 Trying Harder



- 3.7.1 Exercise - I Can Haz Admin?
- 3.8 Backdooring the Login Page
 - 3.8.1 Exercise: Backdoor Galore
 - 3.8.2 Extra Mile Exercise
- 3.9 Further Reading
- 4. DELL SonicWall Scrutinizer Case Study - CVE-2012-XXXX
 - 4.1 Getting Started
 - 4.2 Attack Implementation
 - 4.3 SQL Injection 101
 - 4.3.1 Types of SQL Injection Attacks
 - 4.4 Enumerating the Database
 - 4.4.1 Exercise: What Do We Have Here?
 - 4.5 Getting Code Execution
 - 4.5.1 Exercise: Run Forest, Run!
 - 4.6 Further Reading
- 5. SolarWinds Storage Manager 5.10 - CVE-2012-2576
 - 5.1 Getting Started
 - 5.2 Attack Implementation
 - 5.2.1 Exercise: Right in Front of Your Eyes
 - 5.3 Further Reading
- 6. WhatsUp Gold 15.02 Case Study - CVE-2012-2589
 - 6.1 Getting Started
 - 6.2 Web Related Attack Vectors
 - 6.3 Attack Implementation
 - 6.3.1 Exercise: Alert("SNMP Rules, Again")
 - 6.4 WhatsUp Gold, Round 2 - SQL Injection
 - 6.4.1 Exercise: Find Me If You Can
 - 6.5 Proving SQL Injection
 - 6.5.1 Demonstrating the Comma Issue
 - 6.5.2 Exercise: The Database Does Not Exist
 - 6.6 Bypassing the Character Restrictions
 - 6.6.1 Exercise: Welcome to the Database, Hax
 - 6.7 Getting "Arbitrary" Code Execution
 - 6.7.1 Exercise: Wherefore Art Thou, Calc?
 - 6.8 Chaining the Vulnerabilities
 - 6.8.1 Exercise: Return of the Calc
 - 6.9 Improving our Payload
 - 6.9.1 EXEC xp_cmdshell 'debug<123.hex';--
 - 6.9.2 Exercise: Make Us Proud
 - 6.10 Further Reading
- 7. Symantec Web Gateway Blind SQLi - CVE-2012-2574
 - 7.1 Getting Started



- 7.2 Blind Pre-Authentication SQL Injection**
 - 7.2.1 Exercise: Yes or No?
- 7.3 Timing-Based Blind SQL Injection**
 - 7.3.1 Exercise: Three Blind Bytes
- 7.4 Blind Extraction of the Admin Hash**
 - 7.4.1 Exercise: See How They Run
- 7.5 Select into OUTFILE Reloaded**
 - 7.5.1 Exercise: Select Nothing into Outfile
- 7.6 Abusing MySQL Delimiters**
 - 7.6.1 Exercise: Select Shell into Hacker
- 7.7 Getting Code Execution**
- 7.8 Backdooring Symantec Gateway Server with MySQL Triggers**
 - 7.8.1 Exercise: Show us What you Got
- 7.9 Further Reading**
- 8. AlienVault OSSIM – CVE-2012-2594, CVE-2012-2599**
 - 8.1 Getting Started**
 - 8.2 Vulnerability Analysis and Attack Plan**
 - 8.3 Reflected Cross Site Scripting**
 - 8.3.1 Exercise: You Know the Drill
 - 8.4 Blind SQL Injection**
 - 8.4.1 Exercise: Query Me This
 - 8.5 Extracting Data From the Database**
 - 8.5.1 Exercise: What Me Query?
 - 8.6 Bypassing Filters**
 - 8.6.1 Exercise: One By One, They Will Fall
 - 8.7 Extracting the Admin Hash**
 - 8.7.1 Exercise: Take a Break
 - 8.8 Reading Local Files**
 - 8.8.1 Exercise: There is No Spoon
 - 8.9 Further Reading**
- 9. Symantec Web Gateway 5.0.2 Case Study – CVE-2012-0297**
 - Getting Started**
 - 9.1 Web Related Attack Vectors**
 - 9.2 Local File Inclusion 101**
 - 9.3 Getting Code Execution**
 - 9.3.1 Exercise: Flogging a Dead Horse
 - 9.4 Getting an (apache) Reverse Shell**
 - 9.4.1 Exercise: Null the boy
 - 9.5 Getting a (Root) Reverse Shell**
 - 9.5.1 Exercise: Seriously?
 - 9.6 Further Reading**
- 10. PHPNuke CMS Case Study – CVE – 2010-XXXXX**



- 10.1 Getting Started**
- 10.2 Code Analysis**
 - 10.2.1 Exercise: But I *AM* So Cool!
- 10.3 The Root of the Problem**
 - 10.3.1 Exercise: So Now What?
- 10.4 Attack Implementation**
 - 10.4.1 Stealing the Remote Database
 - 10.4.2 Using the Password Hash to Login
 - 10.4.3 Exercise: My Preshussssss...cookie
- 10.5 Recreating an Authentication Token**
- 10.6 Getting Code Execution**
- 10.7 Getting a Shell**
 - 10.7.1 Exercise: The Whole Ball of Wax
- 10.8 Further Reading**
- 11. Symantec Web Gateway 5.0.3.18 RCE – CVE-2012-2953**
 - 11.1 Getting Started**
 - 11.2 Vulnerability Analysis**
 - 11.2.1 Exercise: Get Me Code Exec
 - 11.3 Testing the Vulnerability**
 - 11.3.1 Exercise: Gimme Shell!
 - 11.4 Further Reading**
- 12. FreePBX Elastix Remote Code Execution – CVE-2012-XXXX**
 - 12.1 Getting Started**
 - 12.2 Vulnerability Analysis**
 - 12.3 Testing the Vulnerability**
 - 12.4 Code Execution**
 - 12.4.1 Exercise: Touch me if you can
 - 12.5 Reverse Shell**
 - 12.5.1 Escalation to the root User
 - 12.5.2 Exercise: Bow down, for I am root