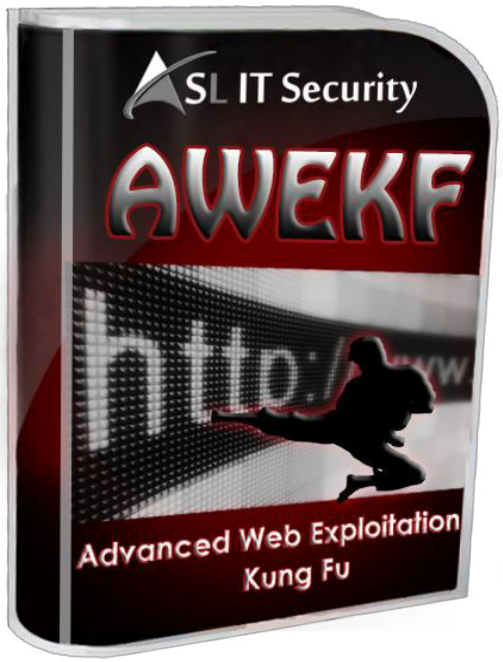
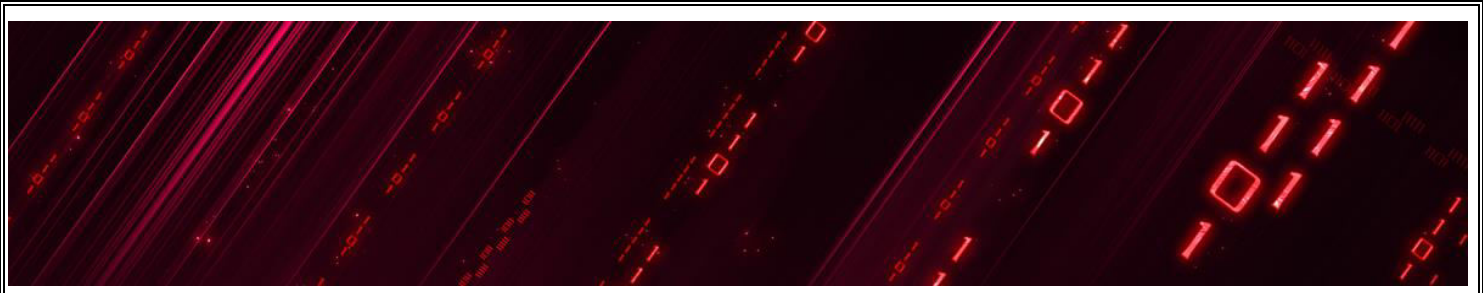
ASL IT Security

Advanced Web Exploitation Kung Fu



V2.0

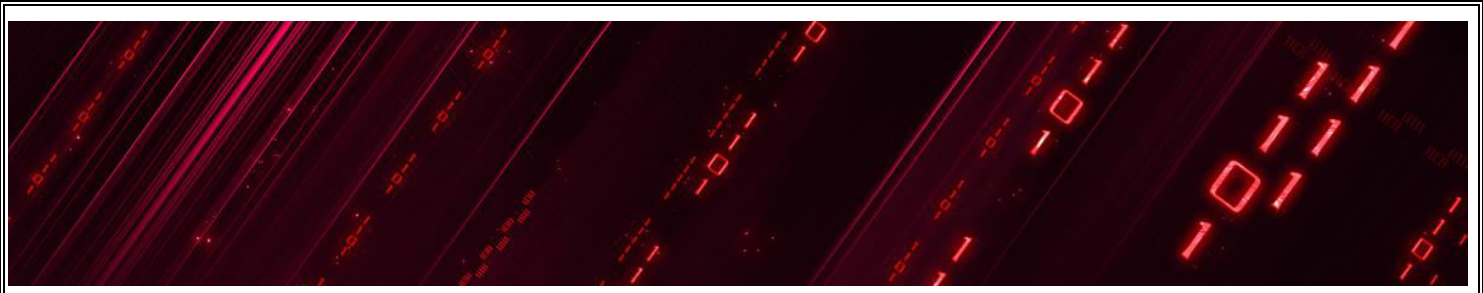


Overview: There is a lot more in modern day web exploitation than the good old alert("xss") and union select. Take your exploitation skills to next level by learning serialization attacks, bypassing hard WAF's, creating stealthiest backdoors in the applications you compromise and chaining vulnerabilities. Find your 0days and write your exploits. Complete training will be hands on based on the challenges faced in real life exploitation.

Course Description: Advanced Web Exploitation Kung Fu is a fast paced training for penetration testers from intermediate level to a pro. It's a complete hands on training where students will learn the skills to chain various exploits with limited impact to enhance the impact of the final attack, detect tricky vulnerabilities which they have missed in past and exploit them. After the training student will be able craft advanced payloads according to requirement in different situations and automate the exploitation by writing custom scripts.

Creating 0 day exploits and bypassing hard filters will not be a dark secret skill after the completion of this training.

- How to detect a vulnerability
- Different methods of exploiting a particular vulnerability
- Evading filters
- Chaining an exploit with others
- Writing custom exploitation scripts
- Stealth Backdoors
- Stealth advanced payloads
- Mitigation methods
- And much more…..



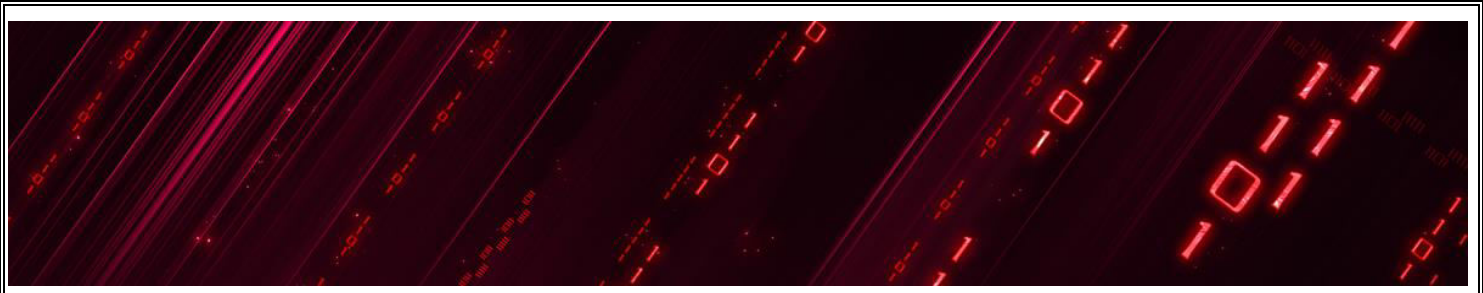
Trainers:

Abhishek Sahni has seven years of experience in web application penetration testing. He had reported security vulnerabilities to yahoo and AOL. Abhishek had conducted successful security trainings on various topics for government agencies in India and abroad. He had performed many web application penetration tests and faced really challenging scenarios and found methods to overcome them.

Marco Genovese is an expert penetration tester and programmer. As he had performed several penetration tests for clients on complex web applications he has a good knowledge of developing custom reliable tools and exploits according to the requirement during pentests. In free time Marco enjoys coding complex, realistic and interesting web hacking labs for training the ASL IT Security's pentesting team.

Who should attend this training: This training is designed for penetration testers and web application security professionals with intermediate to advanced skills and wants to take their skills to next level. The training will include a quick overview to traditional exploitation method and will focus in deep on advanced techniques (both hands on). Students will learn to find new vulnerabilities and write 0days.

Student's prerequisites: Knowledge of HTTP protocol and web applications is must for this training. Knowledge of some scripting language to write custom tools and exploits will be a plus but not necessary.



Module 1. Chaining hard filtered SQL injections with other exploits

1.1 Introduction to the scenario (ASL Hack Labs Forums)

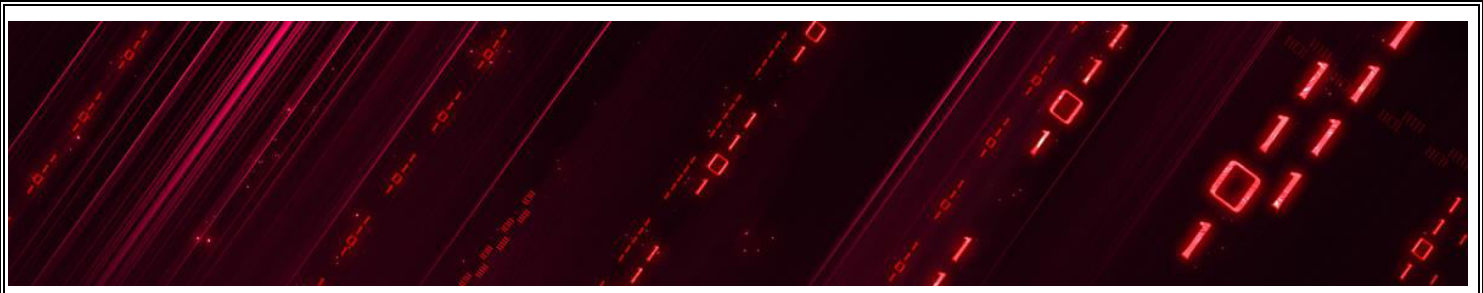
1.2 SQL Injections
1.2.1 Types of sql injection
1.2.2 Detection of SQL injection

1.3 Exploitation
1.3.1 Found injection point, what to do now???
1.3.2 Demo: Extracting the database
1.3.3 Various ways of filter evasion
1.3.4 Demo: filter evasion
1.3.5 No permission to extract db admin ☹
1.3.6 Lets read the local files (LOAD_FILE)
1.3.7 Demo: Can read files but nothing useful ☹
1.3.8 Shell it with into_outfile
1.3.9 Demo: No writable directory. What I do now????

1.4 Chaining SQLI with other exploits
1.4.1 Session Hijacking and Javascript injection
1.4.2 Demo: Javascript injection using sql injection
1.4.3 Demo: Can I haz admin cookies???

1.5 Malicious file Uploads
1.5.1 Introduction to malicious file upload
1.5.2 Bypassing image upload filters
1.5.3 Demo: there is hax0r in my image

1.6 SQL injection prevention



Module 2. See my XSS foo….

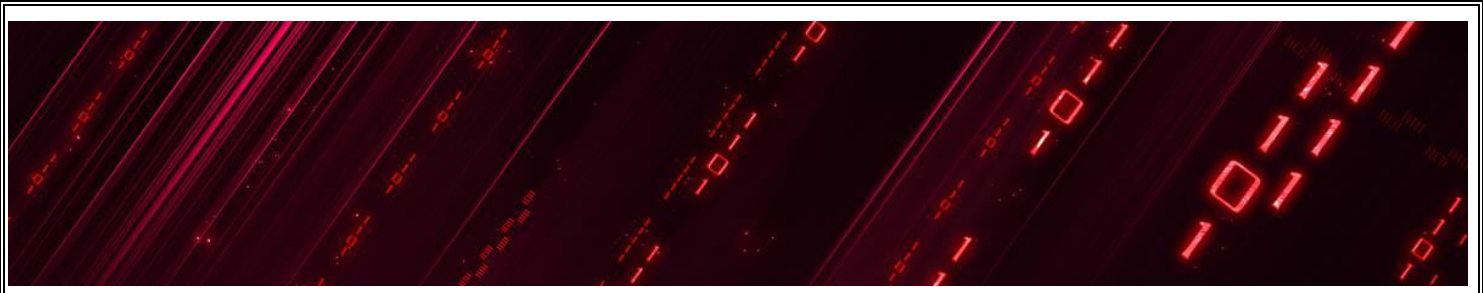
2.1 Introduction to the scenario (ASL Hackme Labs CMS)

2.2 Cross Site Scripting
2.2.1 Types of Cross Site Scripting
2.2.2 New HTML5 attack vectors
2.2.3 Detection of XSS
2.3.4 Demo: Interesting input vector
2.3.5 Bypassing various xss filters
2.3.5 Demo: Filters again

2.3 Exploiting XSS
2.3.1 Headers for XSS prevention
2.3.2 Demo: Hijack admin cookie
2.3.4 HTTP Only Cookie
2.3.5 Demo: What now?? Keylog the admin with XSS. URL Hiding
2.3.6 Demo: Show me your face admin

2.5 More fun with xss. Let me count the ways
2.5.1 Stealing cookies.
2.5.2 Port Scanning
2.5.1 Demo: Show me your screen.
2.5.2 Demo: Sniff Sniff. I smell traffic

2.6 Preventing Cross Site Scripting attacks



Module 3. File Inclusions

3.1 Introduction to scenario (ASL Hackme Labs Image gallery)

3.2 File inclusion vulnerabilities
3.2.1 Types of file inclusions vulnerabilities
3.2.2 Different php wrappers
3.2.3 Detecting file inclusion vulnerabilities

3.3 Various attack methods
3.3.1 Demo: Abuse the wrappers
3.3.2 Demo: Executing code with LFI

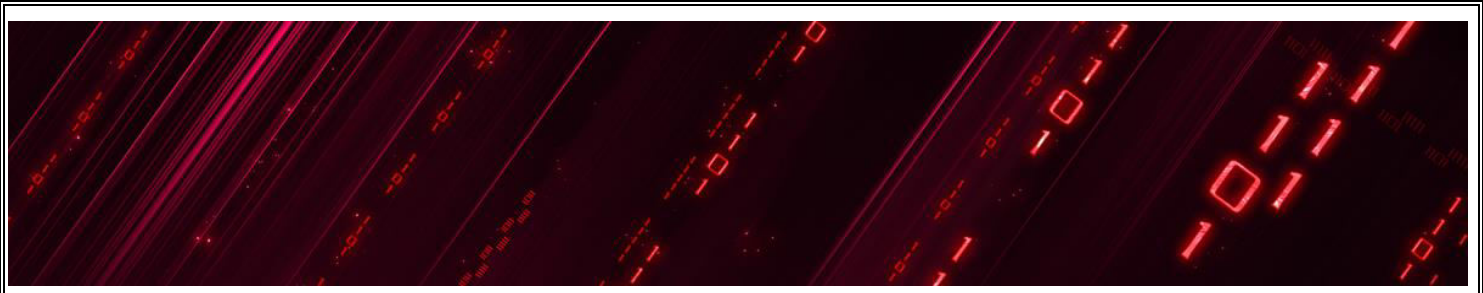
3.4 File inclusion prevention

Module 4. Exploiting SQLI in hard to exploit filters

4.1 Introduction to the scenario

4.2 Different methods to evade the filters
4.2.1 Blind SQL injection
4.2.2 Boolean blind
4.2.3 Time based blind
4.2.3 Demo: bypassing filters and injecting
4.2.4 Demo: Automate the injection
4.2.5 Uploading shell using blind sql injection
4.2.4 Demo: Shell the server with into_outfile

4.3 Abusing mysql triggers
4.3.1 Demo: Escalating privileges to admin



Module 5: Attacking XML and HTML5

5.1 Introduction to the scenario (ASL Hackme Labs webshop)

5.2 Various XML attacks
5.2.1 Detection of XML vulnerabilities
5.2.2 Various filter evasions
5.2.3 Impact of XML attacks

5.3 XML External Entity injection
5.3.1 Ways to exploit
5.3.2 Attack implementation
5.3.2 Demo: Let me read all the files

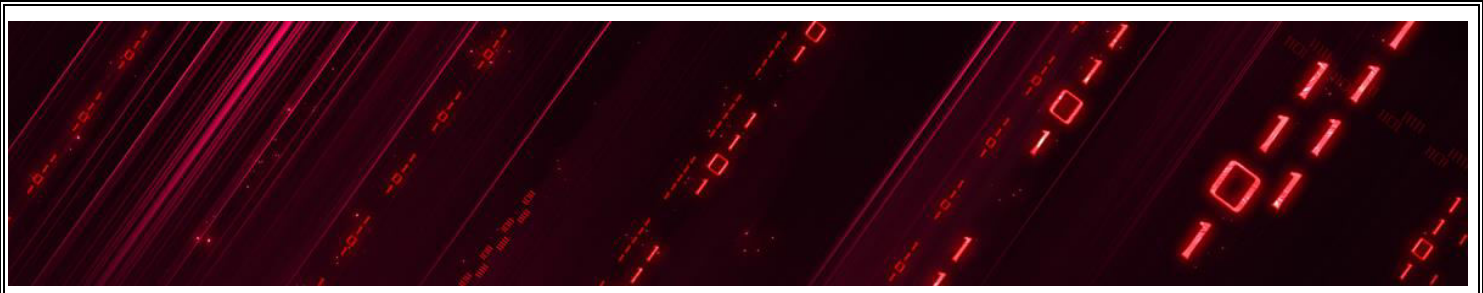
5.4 Introduction to HTML5 local storage
5.4.1 Demo: Admin claims he don't store CC details on sever. I see what you did there.
5.4.2 Getting local storage data
5.4.3 Demo: alert(localstorage.getitem);
5.4.4 Stealing local storage data
5.4.5 Demo: Steal the user's data

5.5 Prevention of XML and HTML5 attacks

Module 6. Improper Session managements

6.1 Introduction to the scenario

6.2 Various Session related vulnerabilities
6.2.1 Sensitive data in cookies
6.2.2 Demo: Analyzing the cookies



6.2.3 Weak Session identifiers
6.2.4 Demo: Brute them sessions. I found u admin.

6.3 Session management security checklist

Module 7. Chaining CSRF and HPP

7.1 Introduction to the scenario

7.2 Introduction to HPP and CSRF
7.2.1 Types of CSRF
7.2.2 CSRF detection
7.2.3 CSRF prevention mechanisms

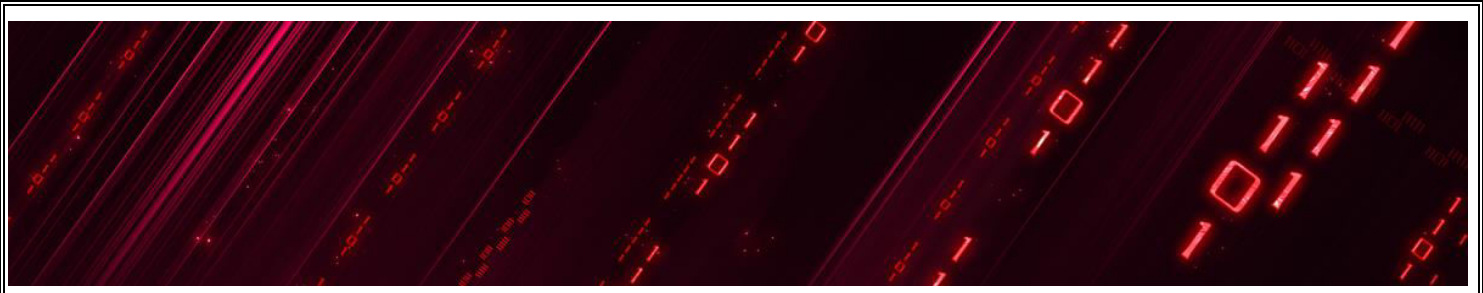
7.3 Chaining CSRF and HPP
7.3.1 Demo: Mail me the reset link
7.3.2 Demo: Reset admin password

Module 8. Command Execution

8.1 Introduction to scenario

8.2 Command Injection Flaws
8.2.1 Detection of command injection
8.2.2 Impact of command injection flaws
8.2.3 Demo: Write shell to server
8.2.4 Backdooring existing php files
8.2.5 Demo: Can you see me???

8.3 Countermeasures for Command Injection



Module 9. Exploiting PHP object injection

9.1 Introduction to scenario

9.2 PHP object injection vulnerabilities
9.2.1 serialize() and unserialize()
9.2.2 Detection if object injection
9.2.3 Real world examples
9.2.4 Demo: From read file to code execution

9.3 Prevention mechanisms