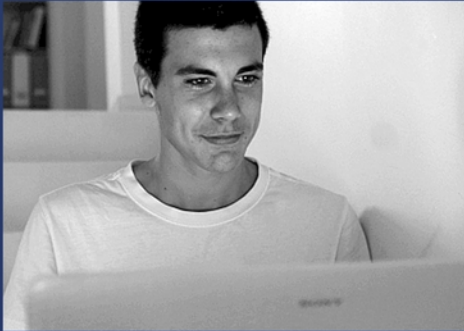


Hacker Highschool

SECURITY AWARENESS FOR TEENS



LESSON 16 CRACKS AND EXPLOITS

DRAFT



HACKING IS LEARNING
www.hackerhighschool.org

ISECOM

Creative Commons 3.3 Attribution-Non-Commercial-NoDerivs ISECOM

WWW.ISECOM.ORG - WWW.OSSTMM.ORG - WWW.HACKERHIGHSCHOOL.ORG - WWW.BADPEOPLEPROJECT.ORG - WWW.OSSTMMTRAINING.ORG



WARNING

The Hacker Highschool Project is a learning tool and as with any learning tool there are dangers. Some lessons, if abused, may result in physical injury. Some additional dangers may also exist where there is not enough research on possible effects of emanations from particular technologies. Students using these lessons should be supervised yet encouraged to learn, try, and do. However ISECOM cannot accept responsibility for how any information herein is abused.

The following lessons and workbooks are open and publicly available under the following terms and conditions of ISECOM:

All works in the Hacker Highschool Project are provided for non-commercial use with elementary school students, junior high school students, and high school students whether in a public institution, private institution, or a part of home-schooling. These materials may not be reproduced for sale in any form. The provision of any class, course, training, or camp with these materials for which a fee is charged is expressly forbidden without a license, including college classes, university classes, trade-school classes, summer or computer camps, and similar. To purchase a license, visit the LICENSE section of the HHS web page at <http://www.hackerhighschool.org/licensing.html>.

The Hacker Highschool Project Project is an open community effort and if you find value in this project, we ask that you support us through the purchase of a license, a donation, or sponsorship.



Table of Contents

Introduction.....	5
Vulnerabilities and Exploits.....	6
What are Vulnerabilities?.....	6
2.1.2 What is an Exploit?.....	6
2.2 Some of the most common Web Security Vulnerabilities (categorized).....	7
2.3 The most common web vulnerabilities and How to exploit them?.....	8
1) XSS.....	8
3.1 Exploit Resources.....	11



Contributors

Marta Barceló, ISECOM

Pete Herzog, ISECOM

Bob Monroe, ISECOM

Aneesh Dogra

ISECOM



Introduction

If you are a web user its most likely you must have seen Websites, Projects, Organizations getting hacked, ever wondered how these Evil People hack stuff, and How people like you can protect it? If yes, this chapter is for you.

The focus of this chapter will purely deal with Software Vulnerabilities, and We'll also look on the Top 10 PHP vulnerabilities which are being hacked and exploited everyday, At the end of this chapter we'll have some basic exercises based on what you'll learn throughout this chapter.

So, lets grab a cup of coffee and get wired up.



Cracks and Exploits

What are Vulnerabilities?

A Vulnerability in its most basic sense is a flaw or a weakness which can be exploited in one or more ways. So, Vulnerability = System Flaw + Attackers Access to Flaw + Attackers Capability to exploit the flaw. A flaw which cannot be exploited or which has no value to the organization, doesn't effects its missions, goals, operations and working is not considered a vulnerability.

For example consider the following scenario :-

Aneesh found an SQL Server's username/password but fortunately the server doesn't allow logging in from an external machine. Now, This is a vulnerability if and only if Aneesh can devise a way to login or do something to their SQL Server which can harm the organization's assets.

An asset doesn't strictly mean a data breach, it can be anything that can has value to the organization, its business operations and their continuity, including information resources that support the organization's mission.

2.1.2 What is an Exploit?

An exploit is a piece of software, code or sometimes even a chunk of data which takes an advantage of a bug or vulnerability in order to cause unintended behavior.

These can be classified into following main categories :-

- 1) Remote Exploits
- 2) Local Exploits
- 3) Web Exploits
- 4) Denial Of Service Exploits

A Remote Exploit is used to exploit vulnerabilities in victims machine to gain remote access.

A Local Exploit is used to exploit vulnerabilities in a machine to mainly gain access to higher privileges.

Note: Most of the times Remote and Local exploits are used in couple to gain access to root in a victim machine.

A Web Exploit is used to exploit vulnerabilities in a web software.



DoS or Denial-of-Service are the types of exploits which are used to make the resources of the victim machine or a network unavailable to its unintended users.

2.2 Some of the most common Web Security Vulnerabilities (categorized)

1) Injection

These type of vulnerabilities exist when an attacker is able to modify a command or a query to a Server, or interpreter. Example: SQL Injection, Cross Side Scripting etc.

2) Broken Authentication

Attackers can use flaws in the authentication or session management mechanisms to impersonate users. These vulnerabilities are very dangerous and can be lethal to victim's privacy.

3) CSRF (Cross-Side-Request Forgery)

In this type of attack, a dummy that is a forged HTTP request is used to trick victims into submitting them. As an example consider a webpage with a simple feedback form which is coded in way to send HTTP requests to Paypal's server to issue a \$100 USD transaction in the name of the company, now if you are logged into your Paypal account you would have just lost your \$100 USD.

4) Security Misconfiguration

These type of vulnerabilities include Default Password, Default Configurations, using plaintext password storage instead of hashed one etc. Example most of the Web Content Management Systems have their admin panel situated at "site/admin/" which can act as a starting point for an attacker. A worse case would be default passwords, most of routers use "admin:admin" or "router:router" as their default credentials which can be easy guessed by an attacker and lead him to an easy way to get admin rights.

5) Open Access

These type of vulnerabilities are rare but they do exist, such type of vulnerabilities allow an attacker to access privileged resources. If this security hole exists an attacker can simply edit a url and access content which was meant for authenticated users, for example, consider a Company website which provides access to sensitive material to its staff, now if the website has this security hole the attacker could get access to these resources without even having to log in.



6) Lack of validation

These type of vulnerabilities exists if the developer din't cared about validation of data which is received as input, like in case of a login form which is getting data from an SQL server and using user's input to query the database, in this case the code should filter out the data sent by the user so as it doesn't contains any SQL commands or else it can lead to an Injection attack.

2.3 The most common web vulnerabilities and How to exploit them?

1) XSS

XSS aka Cross-Side-Scripting is a type of vulnerability which allows the attacker to inject HTML code into a webpage. These are of 2 types persistent and non-persistent.

a) Non-Persistent

These are most commonly occurring type of XSS, the input injected by the attacker is

immediately used to render a page (without even sanitizing the input params). Now, the attacker than inject any code and it will be used to render the page.

b) Persistent

These are the most dangerous form of XSS, it mainly occurs when the HTML code injected by the attacker is actually saved by the server and displayed on normal pages.

Vulnerable PHP Code (save it as 'vul.php') :-

```
<html>
```

```
</head>
```

```
<title>Vulnerable to XSS</title>
```

```
</head>
```

```
<body>
```

```
<h1>I am vulnerable to XSS please click on the links below to Exploit me.</h1>
```

```
<ul>
```

```
<li><a href="vul.php?param=<script type='javascript'>alert('XSSSED!')</script>">Javascript  
Alert Box</a>
```

```
<li><a href="vul.php?param=<h1>Yo Homie!</h1>">Header Injection</a>
```




```
<li><a href="vul.php?param=<img src='http://thednetworks.com/wp-content/uploads/2012/01/website-hacked-steps-to-folow-to-recover.jpeg'>">Image Injection</a>
```

```
</ul>
```

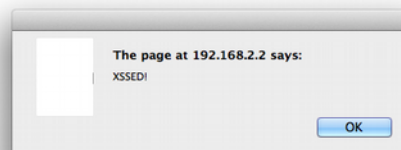
```
<?php
    if (isset($_GET['param'])) {
        echo $_GET['param'];
    }
?>
</body>
```

Javascript Injection:-

Note: This will fail on most of the browsers. Browsers prevent detect JavaScript injection to ensure security.

I am vulnerable to XSS please click on the links below to Exploit me.

- [Javascript Alert Box](#)
- [Header Injection](#)
- [Image Injection](#)



Header Injection:-

I am vulnerable to XSS please click on the links below to Exploit me.

- [Javascript Alert Box](#)
- [Header Injection](#)
- [Image Injection](#)

Yo Homie!

Image Injection:-

I am vulnerable to XSS please click on the links below to Exploit me

- [Javascript Alert Box](#)
- [Header Injection](#)
- [Image Injection](#)



Note: The image is not mine, I got it from (<http://thednetworks.com/>)

For examples of XSS bugs in real world web applications and websites visit <http://www.xssed.com/>

2) SQL Injection

Another common injection vulnerability, This type of vulnerability is actually quite dangerous and has a lot of impact on a website if exploited, SQL injection vulnerabilities occurs when an attacker is able to inject SQL code into a web application. Injection SQL code could be very lethal indeed, it can allow an attacker to change database contents, Dump the whole database, steal username/passwords etc. Many online tutorials are written on how to exploit SQL injection, and you can check them out in the Further reading section.

3) Arbitrary File Upload vulnerabilities

These type of vulnerabilities allow an attacker to upload any type of file to a web server. These files can include PHP backdoors and other malicious content. These vulnerabilities are not so common now but have a great impact on the website/webapp if exploited, An attacker can get access to all files/folders and do mostly anything if he manages to upload a PHP backdoor to the server. Yes, there are ways to secure your server using appropriate permissions and sanitizing the file types before uploading to the server.

4) Client-Side Validation


These vulnerabilities exist when a webapp relies only on Client-Side validation to sanitize inputs. These include Javascript Validation etc. Exploiting such applications is very easy and most of the times disabling your browser to handle javascript can bring down their Validator and then try the vulnerabilities we discussed earlier.



3.1 Exploit Resources

There are a ton of websites to get news about the latest exploits released, I have listed some of the best one's here.

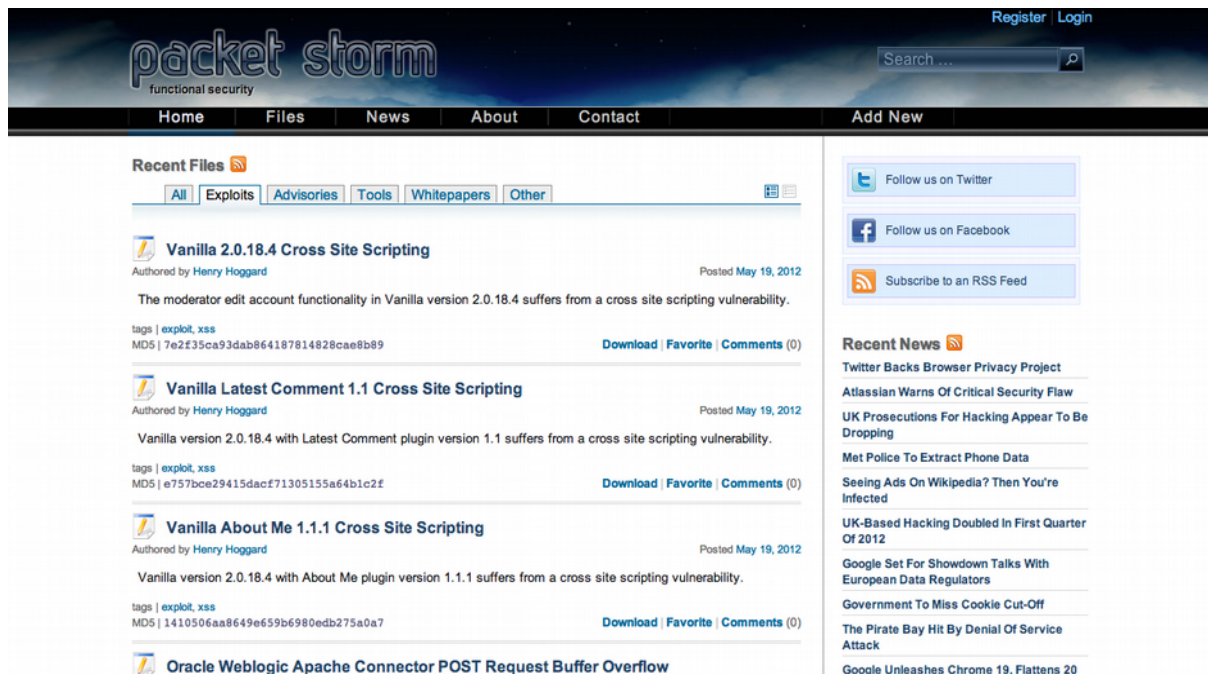
1) Exploit-db.com



The screenshot shows the homepage of the Exploit Database (EDB). The header features the site's logo, navigation links (HOME, BLOG, GHDB, ABOUT, REMOTE, LOCAL, WEB, DOS, SHELLCODE, PAPERS, SEARCH, SUBMIT), and social media icons. A banner for 'The Exploit Database' describes it as an ultimate archive of exploits and vulnerable software. To the right, there's a 'GOOGLE HACKING-DATABASE' section with a list of recent exploits. Below this, a 'Remote Exploits' table is displayed.

Date	D	A	V	Description	Plat.	Author
2012-05-19	✓	-	✓	Active Collab "chat module" <=> 2.3.8 Remote PHP Code Injection Exploit	221	php metasploit
2012-05-19	✓	-	✓	Squiggle 1.7 SVG Browser Java Code Execution	305	multiple metasploit
2012-05-19	✓	-	✓	Oracle Weblogic Apache Connector POST Request Buffer Overflow	381	windows metasploit
2012-02-17	✓	-	✓	HP VSA Remote Command Execution Exploit	560	hardware Nicolas Gregoire
2012-05-13	✓	-	✓	Firefox 8/9 AttributeChildRemoved() Use-After-Free	2820	windows metasploit
2012-05-12	✓	-	✓	Distinct TFTP 3.01 Writable Directory Traversal Execution	1089	windows metasploit

2) Packetstormsecurity.org



The screenshot shows the Packet Storm Security website. The header includes the site logo, navigation links (Home, Files, News, About, Contact, Add New), and a search bar. The main content area is titled "Recent Files" and lists several security advisories, including "Vanilla 2.0.18.4 Cross Site Scripting" and "Vanilla Latest Comment 1.1 Cross Site Scripting". Each entry includes the author (Henry Hoggard), a brief description of the vulnerability, and links to download, favorite, or comment. A sidebar on the right contains social media links (Twitter, Facebook, RSS) and a "Recent News" section with headlines like "Twitter Backs Browser Privacy Project" and "Atlassian Warns Of Critical Security Flaw".

3) 1337day.com aka Inj3ct0r



The screenshot shows the Inj3ct0r website, which is a repository for exploits. The header features a navigation bar with links like [home], [contents], [platforms], [shellcode], [search], [submit], [links], [root], [team], [rss], [style], [mirror], and [db: 18312]. The main content area is titled "Inj3ct0r" and includes a search bar and a list of exploits. The exploits are categorized into "Inj3ct0r", "remote exploits", and "local exploits". Each entry includes a date, description, type, hits, risk, and author. For example, under "Inj3ct0r", there are entries for "Inj3ct0r wishes you a Happy Milw0rm 1337day!!!" and "Emergency message to all Inj3ct0r users". Under "remote exploits", there are entries for "Active Collab 'dust module' <= 3.3.8 Remote PHP Code Injection Exploit" and "Spiggie 1.7 SVG Browser Java Code Execution". Under "local exploits", there are entries for "Veritasov 2.27 Local Privilege Escalation Exploit" and "SkinCrafter ActiveX Control version 3.0 Buffer Overflow".

Today's teens are in a world with major communication and productivity channels open to them and they don't have the knowledge to defend themselves against the fraud, identity theft, privacy leaks and other attacks made against them just for using the Internet. This is the reason for Hacker Highschool.

The Hacker Highschool project is the development of security and privacy awareness learning materials for junior high and high school students.

Hacker Highschool is a set of lessons and a practical means of making hackers. Beyond just providing cybersecurity awareness and critical Internet skills, we need to teach the young people of today how to be resourceful, creative, and logical, traits synonymous with hackers. The program contains free security and privacy awareness teaching materials and back-end support for teachers of accredited junior high, high schools, and home schooling. There are multiple workbooks available in multiple languages. These are lessons that challenge teens to be as resourceful as hackers, including safe Internet use, web privacy, researching on the internet, avoiding viruses and Trojans, legalities and ethics, and more.

The HHS program is developed by ISECOM, a non-profit, open-source research group focused on security awareness and professional security development and accreditation.