PortSwigger

LOGIN

Academy home

ntexts

Back to all topics

What is XSS?

How does XSS work?

Impact of an attack

Proof of concept

Testing

Reflected XSS

Stored XSS

DOM-based XSS

XSS contexts

Between HTML tags

In HTML tag attributes

JavaScript

Client-side template injection

Exploiting XSS vulnerabilities

Dangling markup injection

Content security policy (CSP)

contexts

XSS, a key task is to identify the XSS

where attacker-controllable data appears.
cessing that is being performed on that

select one or more candidate XSS
ffective.

SS cheat sheet to help testing web
ter by events and tags and see which
he cheat sheet also contains AngularJS
sections to help with XSS research.

tags

SIGN UP LOGIN

When the XSS context is text between HTML tags, you need to introduce some new HTML tags designed to trigger execution of JavaScript.

Some useful ways of executing JavaScript are:

```
<script>alert(document.domain)</script>
<img src=1 onerror=alert(1)>
```

Back to all topics
What is XSS?
How does XSS work?
Impact of an attack
Proof of concept
Testing
Reflected XSS
Stored XSS
DOM-based XSS
XSS contexts
Between HTML tags
In HTML tag attributes
JavaScript
Client-side template injection
Exploiting XSS vulnerabilities
Dangling markup injection
Content security policy (CSP)

TML context with nothing encoded →

L context with nothing encoded →

TML context with most tags and

TML context with all tags blocked →

vent handlers and `href` attributes

ome SVG markup allowed →

tributes

When the XSS context is into an HTML tag attribute value, you might sometimes be able to terminate the attribute value, close the tag, and introduce a new one. For example:

```
"><script>alert(document.domain)</script>
```

More commonly in this situation, angle brackets are blocked or encoded, so in which it appears. Provided you can normally introduce a new attribute that an event handler. For example:

```
(document.domain) x="
```

`cus` event that will execute JavaScript and also adds the `autofocus` attribute utomatically without any user interaction. air the following markup.

tribute with angle brackets HTML-

ype of HTML tag attribute that itself can can execute JavaScript without needing xample, if the XSS context is into the can use the `javascript` pseudo-e:

```
(document.domain)">
```

Back to all topics
What is XSS?
How does XSS work?
Impact of an attack
Proof of concept
Testing
Reflected XSS
Stored XSS
DOM-based XSS
XSS contexts
Between HTML tags
In HTML tag attributes
JavaScript
Client-side template injection
Exploiting XSS vulnerabilities
Dangling markup injection
Content security policy (CSP)

You might encounter websites that encode angle brackets but still allow you to inject attributes. Sometimes, these injections are possible even within tags that don't usually fire events automatically, such as a canonical tag. You can exploit this behavior using access keys and user interaction on Chrome. Access keys allow you to provide keyboard shortcuts that reference a specific element. The
lefine a letter that, when pressed in
ry across different platforms), will cause
experiment with access keys and exploit a
hidden input fields using a technique

onical link tag →

g JavaScript within the response, a wide
ferent techniques necessary to perform a

t

imply close the script tag that is enclosing
some new HTML tags that will trigger
if the XSS context is as follows:

```
<script>
...
var input = 'controllable data here';
...
</script>
```

then you can use the following payload to break out of the existing JavaScript

```
or=alert(document.domain)>
```

vser first performs HTML parsing to
locks of script, and only later performs
execute the embedded scripts. The
pt broken, with an unterminated string
osequent script being parsed and

JavaScript string with single quote
ed →

string

ide a quoted string literal, it is often
execute JavaScript directly. It is essential
context, because any syntax errors there
uting.

a string literal are:

```
'
/
```

Back to all topics
What is XSS?
How does XSS work?
Impact of an attack
Proof of concept
Testing
Reflected XSS
Stored XSS
DOM-based XSS
XSS contexts
Between HTML tags
In HTML tag attributes
JavaScript
Client-side template injection
Exploiting XSS vulnerabilities
Dangling markup injection
Content security policy (CSP)

LAB

APPRENTICE

Reflected XSS into a JavaScript string with angle brackets HTML encoded →

Some applications attempt to prevent input from breaking out of the JavaScript string by escaping any single quote characters with a backslash. A backslash parser that the character should be ial character such as a string terminator. In the mistake of failing to escape the s that an attacker can use their own backslash that is added by the application.

/

/ /

ad:

/ /

) / /

e second backslash is interpreted r. This means that the quote is now so the attack succeeds.

JavaScript string with angle brackets

Some websites make XSS more difficult by restricting which characters you are allowed to use. This can be on the website level or by deploying a WAF that prevents your requests from ever reaching the website. In these situations, you need to experiment with other ways of calling functions which bypass these security measures. One way of doing this is to use the `throw` statement with
u to pass arguments to a function without
e assigns the `alert()` function to the
ow statement passes the `1` to the
t). The end result is that the `alert()`
ent.

technique to call functions without

that filters certain characters. You'll have
cribed above in order to solve it.

vaScript URL with some characters

g

g JavaScript within a quoted tag attribute,
e to make use of HTML-encoding to work

HTML tags and attributes within a
ing of tag attribute values before they are

characters that are needed for a successful XSS exploit, you can often bypass the input validation by HTML-encoding those characters.

For example, if the XSS context is as follows:

```
<a href="#" onclick="... var input='controllable data he
```

and the application blocks or escapes single quote characters, you can use the avaScript string and execute your own

```
ain)-&apos;
```

ntity representing an apostrophe or ML-decodes the value of the `onclick` preted, the entities are decoded as s, and so the attack succeeds.

lick event with angle brackets and encoded and single quotes and

erals

literals that allow embedded JavaScript ons are evaluated and are normally t. Template literals are encapsulated in marks, and embedded expressions are

rint a welcome message that includes the

When the XSS context is into a JavaScript template literal, there is no need to terminate the literal. Instead, you simply need to use the `${...}` syntax to embed a JavaScript expression that will be executed when the literal is processed. For example, if the XSS context is as follows:

```
<script>
...
data here`;
```

d to execute JavaScript without

template literal with angle brackets, , backslash and backticks Unicode-

template injection

late framework, such as AngularJS, to embed user input into these templates in e able to inject their own malicious XSS attack.

Back to all topics
What is XSS?
How does XSS work?
Impact of an attack
Proof of concept
Testing
Reflected XSS
Stored XSS
DOM-based XSS
XSS contexts
Between HTML tags
In HTML tag attributes
JavaScript
Client-side template injection
Exploiting XSS vulnerabilities
Dangling markup injection
Content security policy (CSP)

Burp Suite

Web vulnerability scanner
Burp Suite Editions
Release Notes

Vulnerabilities

Cross-site scripting (XSS)
SQL injection
Cross-site request forgery
XML external entity injection
Directory traversal
Server-side request forgery

Back to all topics
What is XSS?
How does XSS work?
Impact of an attack
Proof of concept
Testing
Reflected XSS
Stored XSS
DOM-based XSS
XSS contexts
Between HTML tags
In HTML tag attributes
JavaScript
Client-side template injection
Exploiting XSS vulnerabilities
Dangling markup injection
Content security policy (CSP)

vigger

Follow us

er Ltd.

SIGN UP LOGIN