

[登录](#)

LinkedIn.com中存储型xss(嵌入未经验证的Open Graph)

[niexinming](#) / 2019-01-09 08:24:00 / 浏览数 2697 [技术文章](#) [翻译文章](#) [顶\(0\)](#) [踩\(0\)](#)

翻译自: <https://medium.com/@jonathanbouman/persistent-xss-unvalidated-open-graph-embed-at-linkedin-com-db6188acedd9>

翻译: 聂心明

你想参加私有众测? 我很乐意邀请你, 请联系我Jonathan@Protozoan.nl

背景


在我的上一篇文章中, 我们已经学到[存储型xss](#)中的特殊类型。这次攻击允许我们通过操作oEmbed功能去注入HTML和JavaScript。oEmbed是一种开放的格式, 它允许从其他网站插入内容。大多数富文本平台支持标准的oEmbed。比如, 你会很容易的在你[Wordpress博客](#)的文章中通过粘贴一段Vimeo的视频链接来添加一段视频。Wordpress将把链接转换成HTML。这段HTML加载视频播放器, 然后播放指定的视频。

如果我粘贴[我的推特](#)的链接到这篇文章中, 那么Medium就会创建一段带我照片的简介。这就是oEmbed的功能。在我们知道这些之前, 它们有一个很严重的[漏洞](#), 允许用户

Jonathan Bouman (@JonathanBouman) | Twitter

The latest Tweets from Jonathan Bouman (@JonathanBouman). Medical Doctor (GP in training), Web Developer, Security...



www.twitter.com



Jonathan Bouman (@JonathanBouman)

Beveiligd | https://twitter.com/JonathanBouman

Home Notifications Messages Search Twitter Tweet



Console Sources Elements Network Performance Memory Application Security Audits GraphQL SnappySnippet

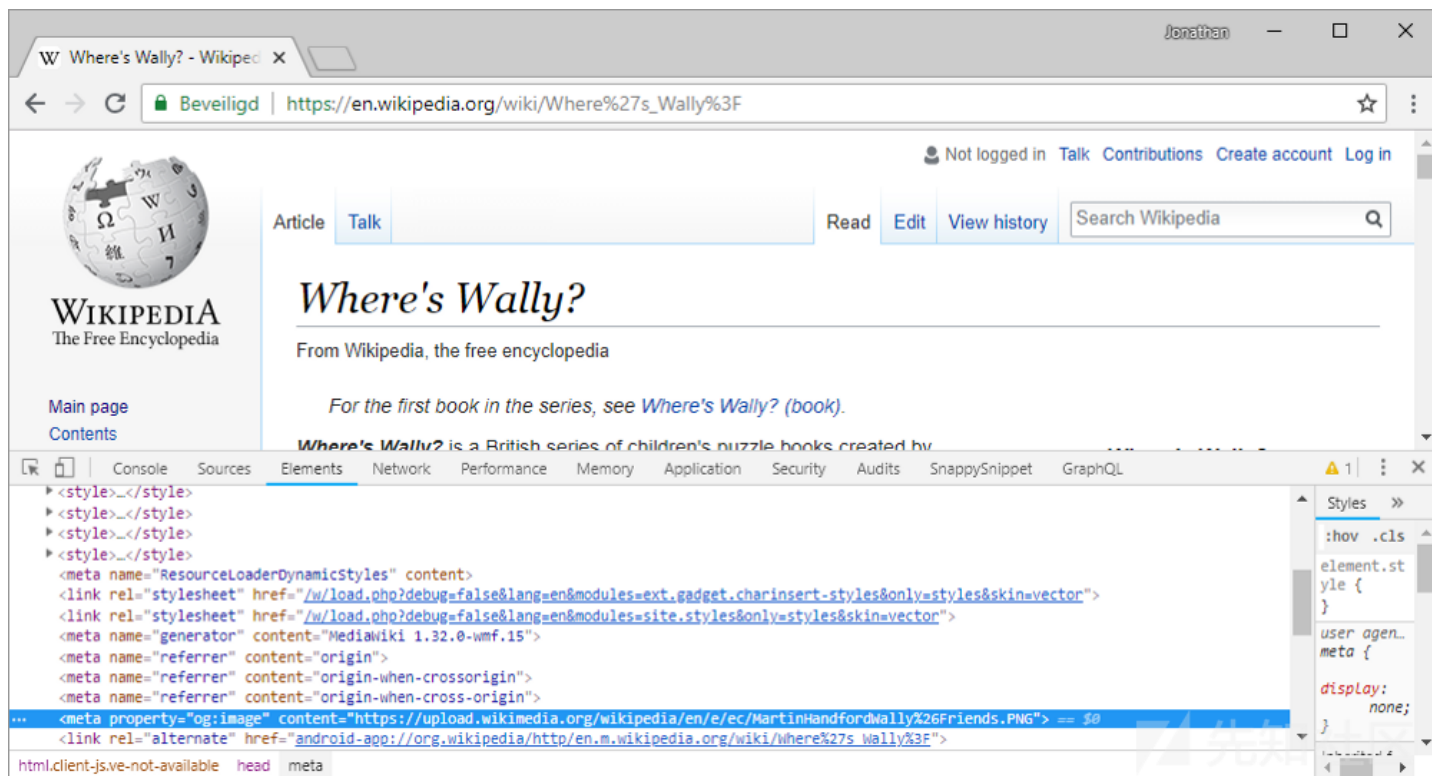
```
<link rel="preload" href="https://abs.twimg.com/k/rn/1.03865_0r0f11c.cn.0088w/4..js" as="script">
<title>Jonathan Bouman (@JonathanBouman) | Twitter</title>
<meta name="msapplication-TileImage" content="//abs.twimg.com/favicons/win8-tile-144.png">
<meta name="msapplication-TileColor" content="#00aced">
<link rel="mask-icon" sizes="any" href="https://abs.twimg.com/a/1523075192/icons/favicon.svg" color="#1da1f2">
<link rel="shortcut icon" href="//abs.twimg.com/favicons/favicon.ico" type="image/x-icon">
<link rel="apple-touch-icon" href="https://abs.twimg.com/icons/apple-touch-icon-192x192.png" sizes="192x192">
<link rel="manifest" href="/manifest.json">
<meta name="swift-page-name" id="swift-page-name" content="me">
<meta name="swift-page-section" id="swift-section-name" content="profile">
<link rel="canonical" href="https://twitter.com/jonathanbouman">
...
<link rel="alternate" type="application/json+oembed" href="https://publish.twitter.com/oembed?url=https://twitter.com/jonathanbouman" title="Jonathan Bouman (@JonathanBouman) | Twitter">
<link rel="search" type="application/opensearchdescription+xml" href="/opensearch.xml" title="Twitter">
```

html head link

Styles

```
:hov .cls
element.st
yle {
}
user agen
link {
display:
none;
```

另一种方式就是通过 [Open Graph 协议](#)来嵌入一个假的富文本。一个网站在页面上添加Open Graph标签, 目的是指定你插入的内容类型。



在用户嵌入内容之前，大多数平台会检查指定的oEmbed和Open

Graph标签。网站会遵守[特殊的规则](#)。然后网站会检查所有的标签，以决定是否嵌入其中或者以怎样的方式嵌入其中。

还有一个有利条件是，独自不需要离开博客去查看富文本内容（比如：视频，图片，表演）。还可以插入像Vimeo和Youtube等视频平台的链接，也可以提高他们的播放量。

大多数平台允许你嵌入一些其他的内容，但是它们会有域名的白名单。看一下 [Wordpress的做法](#)。你想把你的HTML注入到平台中，这可能有点难度。

但是如果白名单的实现出现了错误，那么我们就可以把我们的恶意代码注入到我们的目标平台上去，那么这样的错误是什么呢？

我们已经证明了[oEmbed中的漏洞](#)，现在我们看看能不能利用这个技巧来操纵Open Graph标签呢？让我们试试。

选取目标

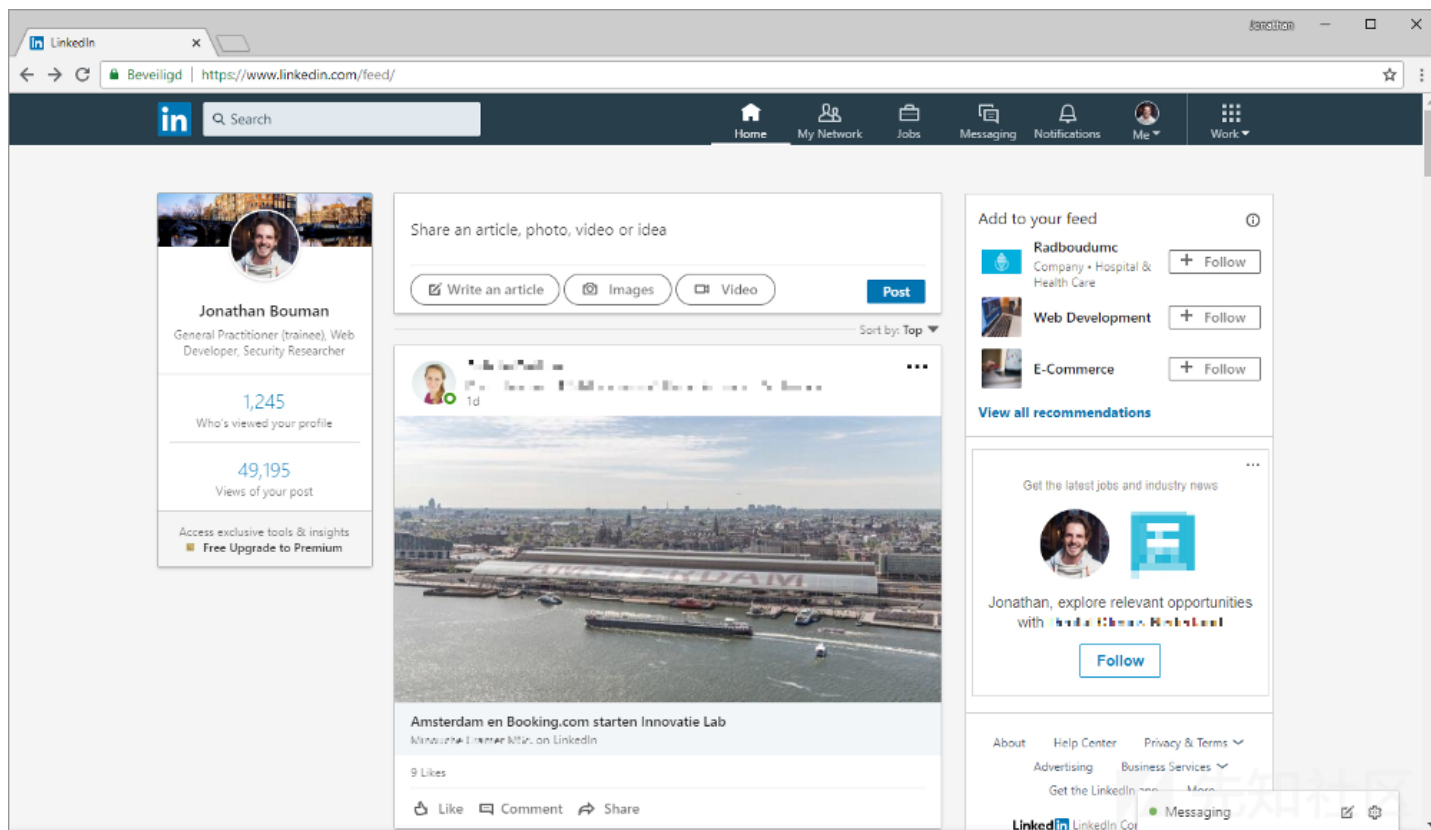
今天我们应该选谁好呢？你说LinkedIn？好主意。当我想联系我关注的研究员时，LinkedIn是我最喜欢的去的地方。而且上面有大量的CEO/CTO/CTTO/CNYANCATO。所

这里我插一句嘴，LinkedIn是一个有一个非常负责的漏洞[披露平台](#)，它们甚至也有私有[众测项目](#)

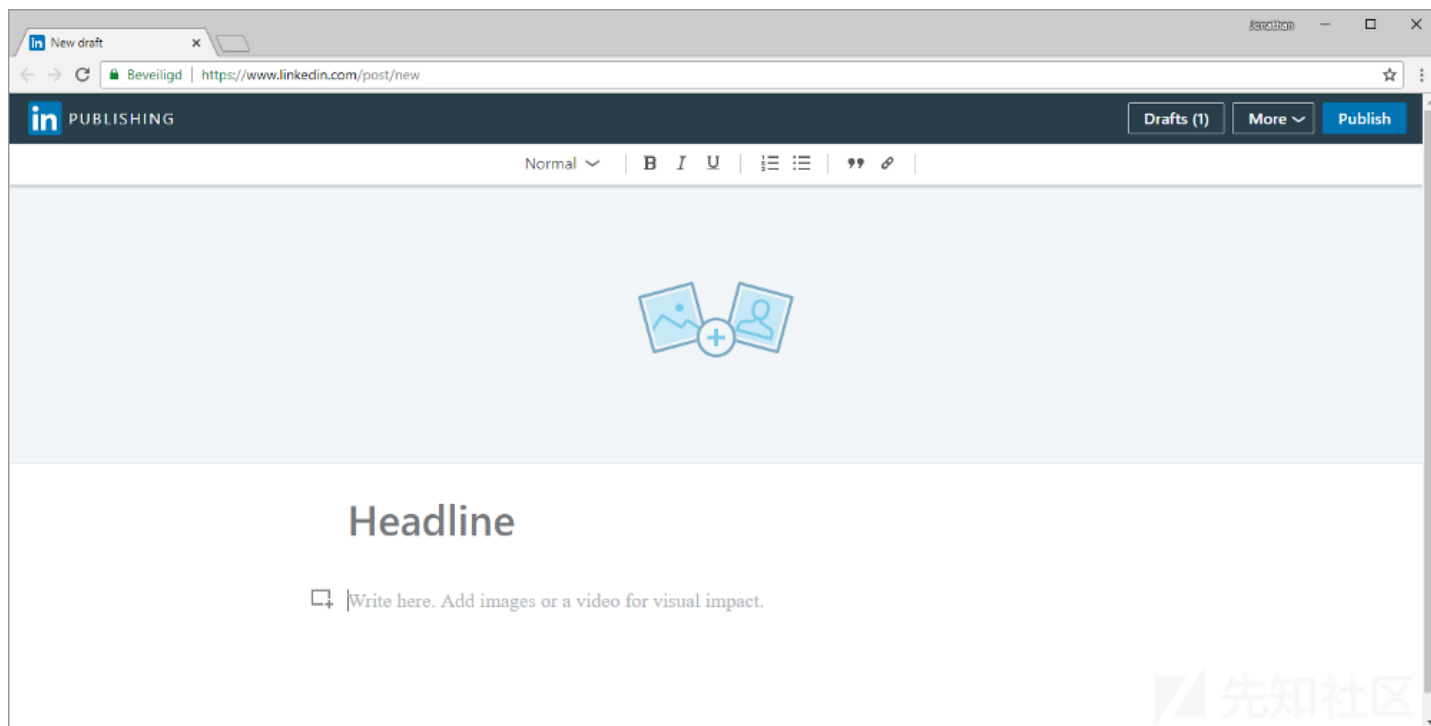
让我们给他们报告一个漏洞，然后争取参加私有众测的机会吧

识别目标

博客，这就是我想要寻找的地方，这里允许我们插入一些额外的内容。对[LinkedIn 文章](#)说hello。通过访问你的动态然后点击 Write an Article 这个按钮，你就会打开这部分功能。



我们访问到了一个看起来比较干净的编辑器，这里允许我们去写我们的第一篇文章，这个文章由标题和内容组成。我们移动鼠标到下面的小图标中，这里有一些可以添加图片

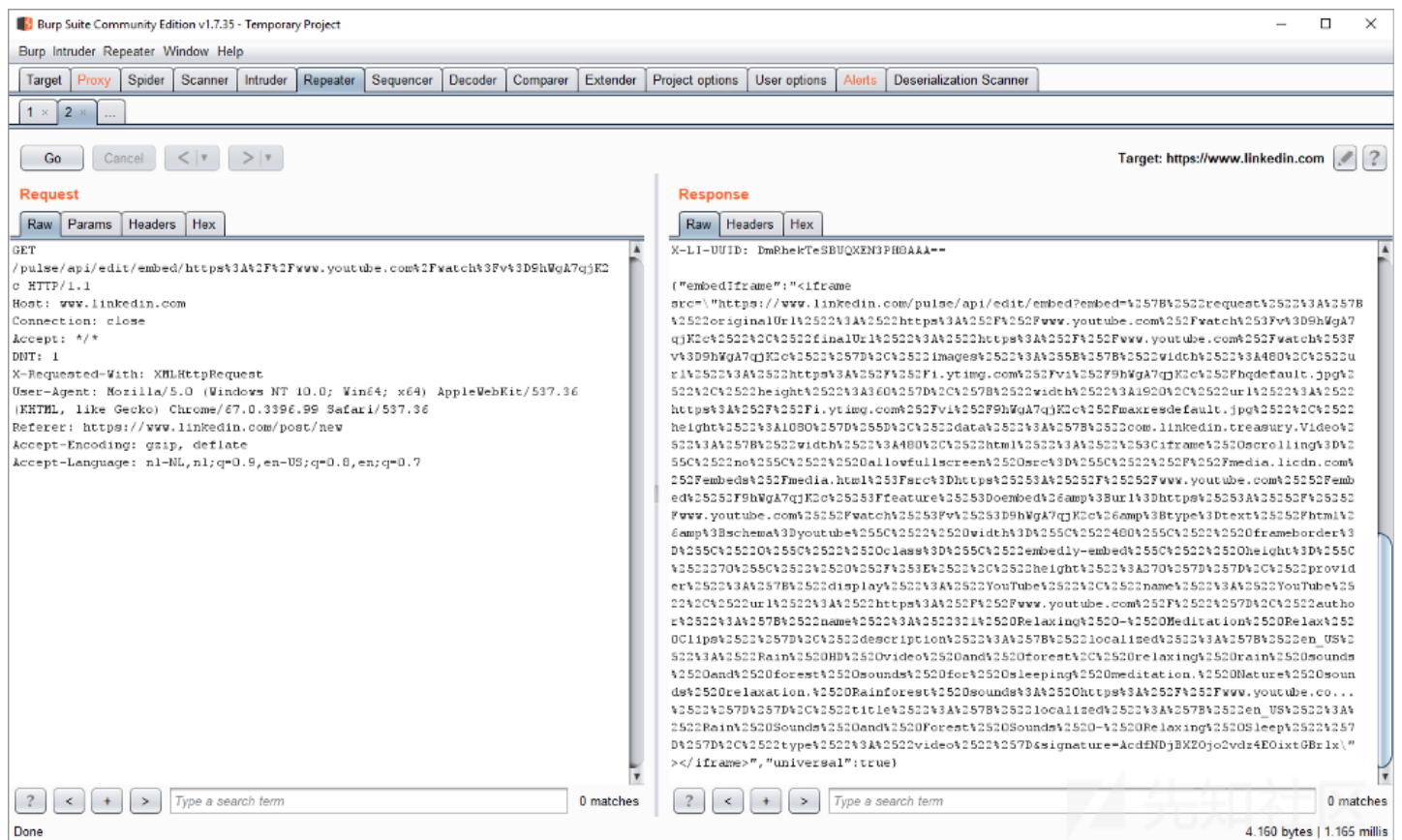


点击这个小图标，我们点开添加链接的按钮，然后在我们的博客中添加富文本内容。

Headline

插入请求

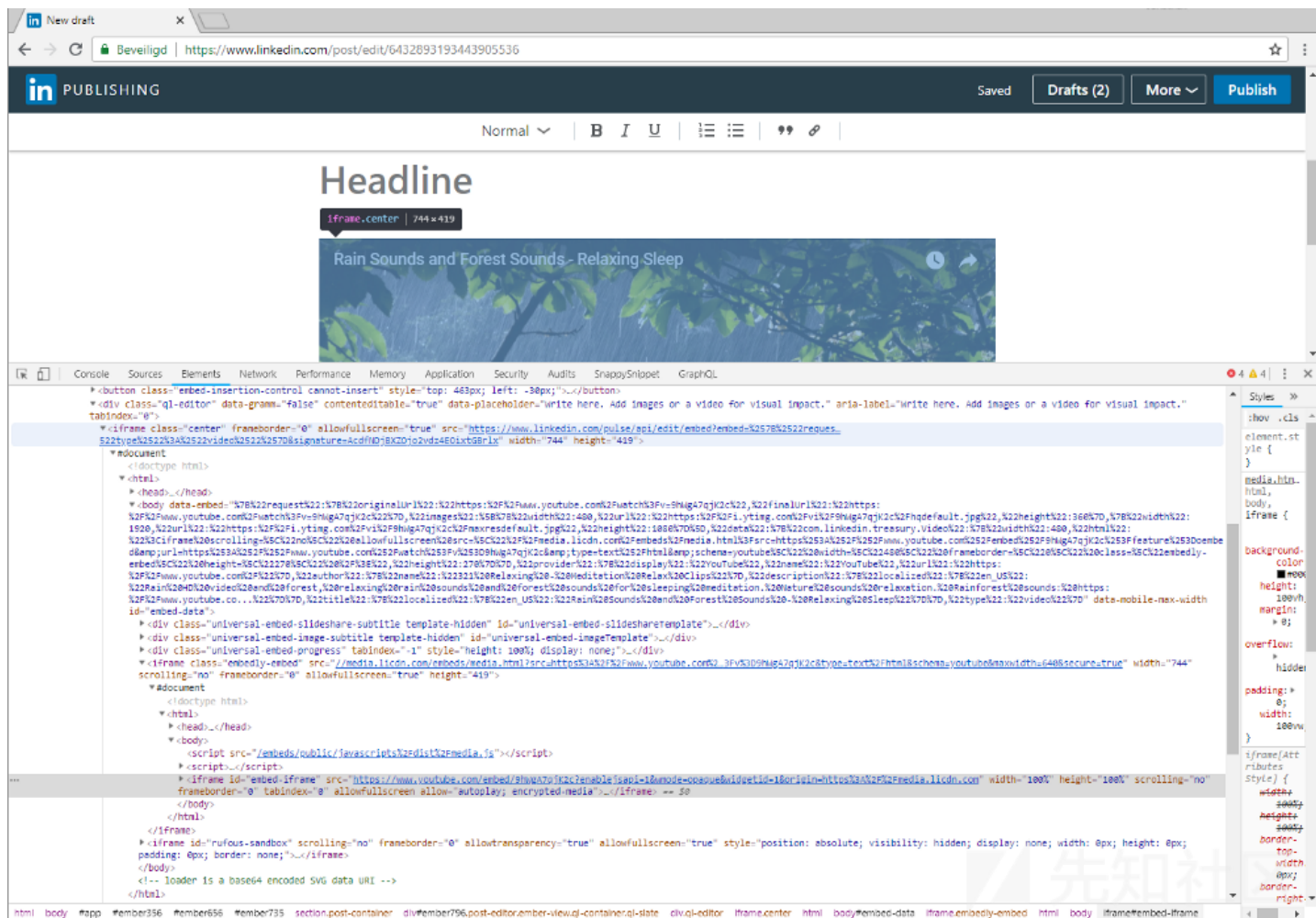
是时候打开 [Burp Suite](#) 然后观察我们的数据传输，当我们填入链接的时候到底发生了什么？



我们看到LinkedIn把我们的url转换成了HTML代码。返回报文被url编码了，下面是解码的结果：

```
{\"embedIframe\":<iframe src=\\\"https://www.linkedin.com/pulse/api/edit/embed?embed={\"request\":{\"originalUrl\":\"https://www.youtube.com/watch?v=3DShWgA7qjK2c\"}}\\\">
```

编辑器把它解析成框架，然后放入我们的文章中，结果被放入了三个框架之中，它们中每一个之后都有一个视频播放器。我们继续，然后就看到我们嵌入的YouTube视频 [relaxing rain sounds](#)

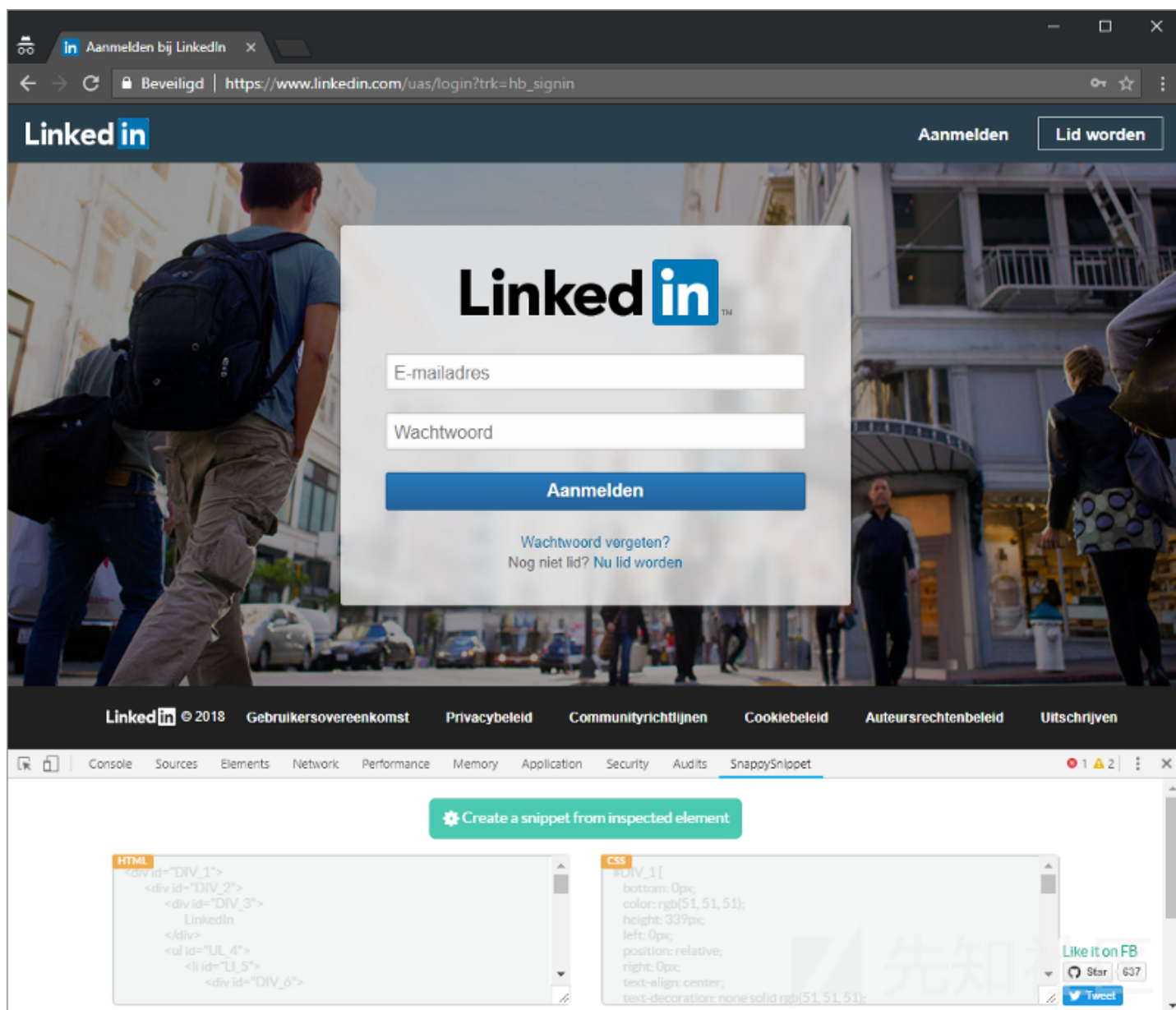


这意味着我们能够注入我们自己的恶意HTML，这个html不是LinkedIn域下的。我们可以创建独立的框架。但是我们不能得到LinkedIn的cookie，我们也不能操作框架外的内容。如果我们能注入一个假的LinkedIn登录页面，然后偷走访客的密码，这样的话，影响也是蛮大的。在LinkedIn中嵌入内容，LinkedIn不会暗示这是一个嵌入的东西，就只是

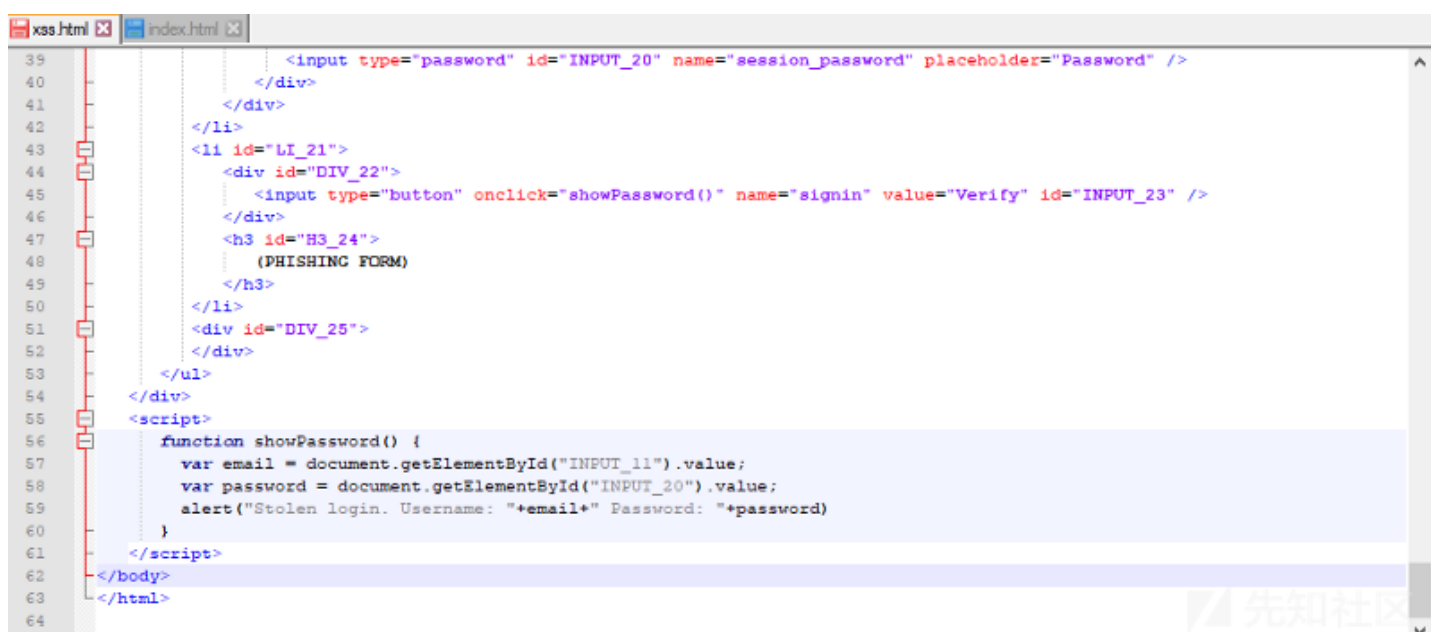
完美的钓鱼页面

首先我们要设计一个钓鱼页面来，因为我希望能让其他人很快的了解到这个漏洞的危害程度到底有多深。弹出一个JavaScript的警告框是不够的。

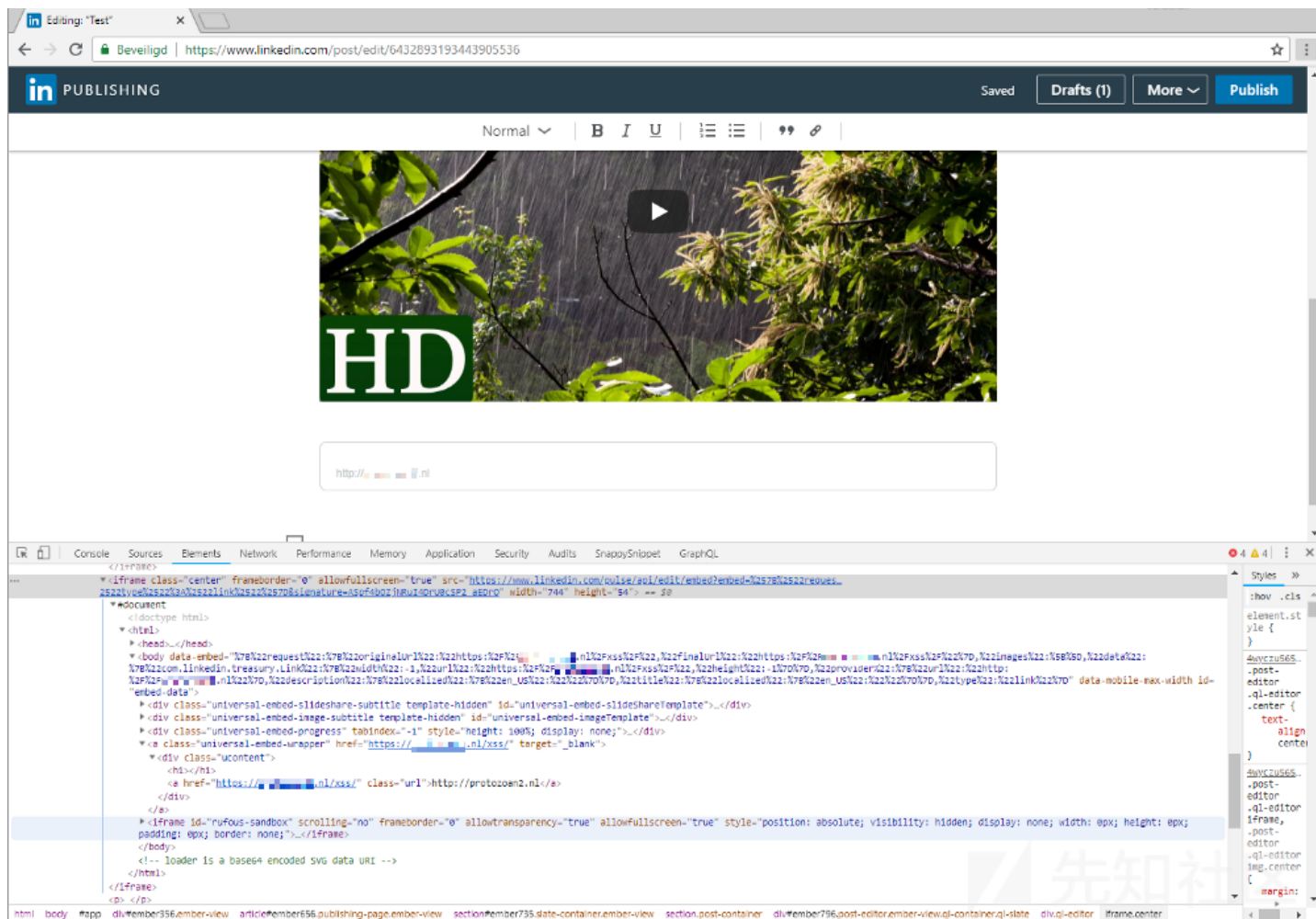
想要复制一个网站的页面元素，可以用SnappySnippet。这是一个Chrome的插件，它可以让你选择网站的元素，并且把它的HTML和css文本都复制下来。



我们把SnappySnippet中的代码贴到新的HTML文件中去，然后把它转换的小一点，并在结尾添加一些JavaScript代码，以捕捉到用户输入的邮箱和密码。如果你输入用户



我把假的登录页面上传到我自己的服务器中，然后把它嵌入到LinkedIn的文章里面，如你所见，我们没有用到oEmbed或者Open Graph标签



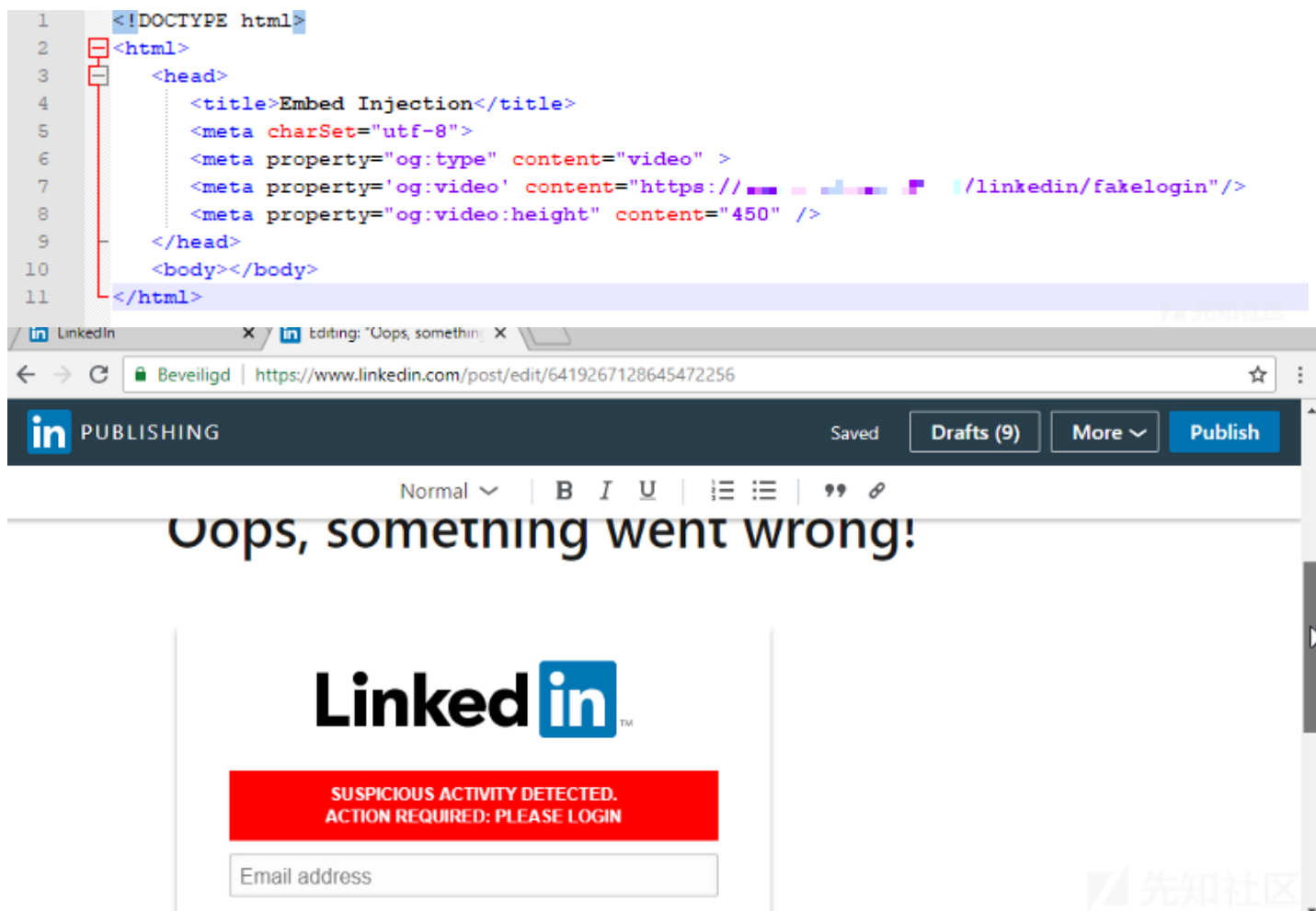
有一点失败的地方就是我们插入的页面周围有一圈灰色边框，看上去就像被嵌在里面的一样。

利用Open Graph

如果你仔细观察 [Open Graph](#) 协议，你会发现有个标签名是 `og:video`。它告诉浏览器视频播放器该如何被嵌入其中。



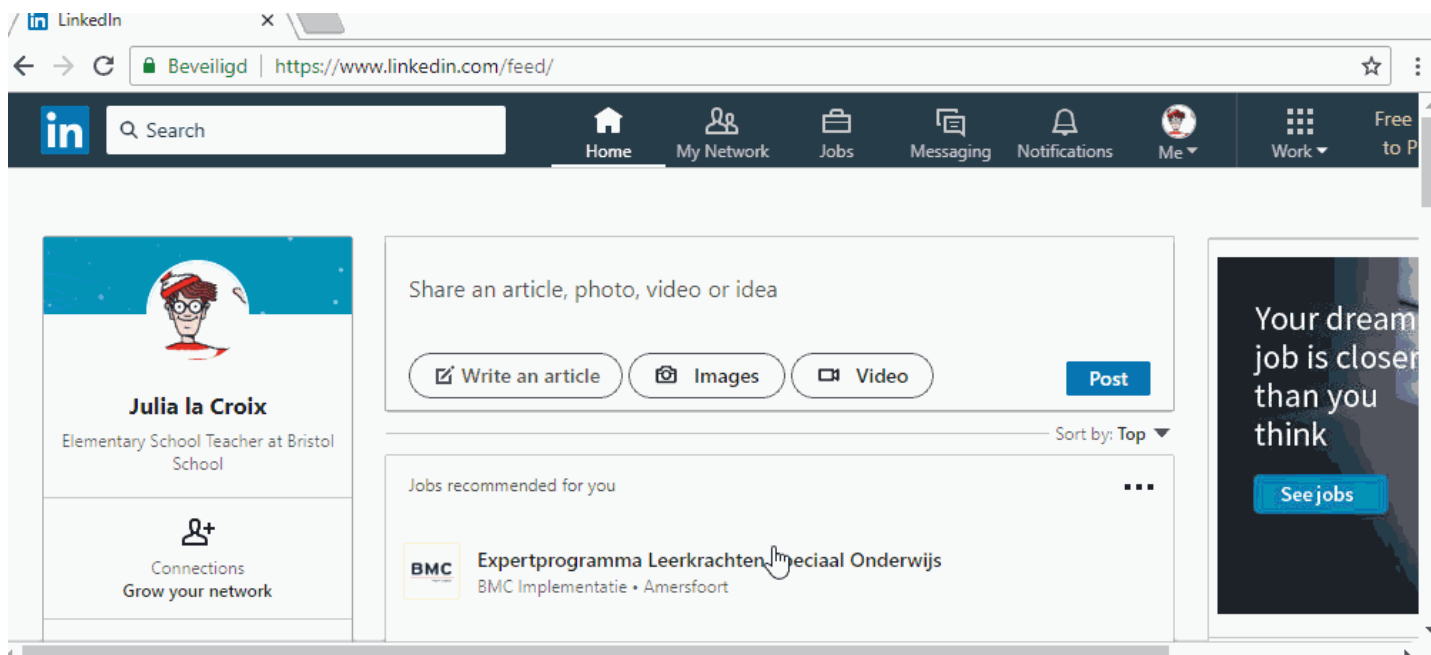
如果我们用 `og:video` 标签加载我们的钓鱼页面而不是视频播放器，那么会发生什么？



欧耶，我们把钓鱼页面成功的嵌入到了LinkedIn的文章中。让我们把文章标题改成'Oops, something went wrong!', 然后继续

现在我们可以通过精心构造一篇文章，然后把链接发给其他人，就可以盗取其他用户的用户名和密码了。或者，我们可以把它公开在LinkedIn上，看看会发生什么。

现在所有的事情都已经完成，是时候写一篇报告交给LinkedIn的安全团队了



结尾

我在LinkedIn.com 中创建了一个完美的钓鱼页面。我们在LinkedIn的文章中在Open Graph Video标签中注入了我们的钓鱼页面。这也是一种存储型xss攻击

攻击的影响

- 完美的钓鱼页面
- 在用户输入他们的凭证之后，我可以把页面自动重定向到另一个页面，而不会引起怀疑（通过使用top.location.href）
- 用beef攻击访问者
- 会造成[点击劫持攻击](#)

我还忘了哪些呢？请给我留言

避免这次攻击的解决方案

- 不要让用户来控制Open Graph标签，使用白名单来控制嵌入的内容
- 不要用框架

赏金

没有

点击收藏 | 0 关注 | 1

[上一篇：CRLF Injection In...](#) [下一篇：CRLF Injection In...](#)

1. 0 条回复
- 动动手指，沙发就是你的了！

[登录](#) 后跟帖

先知社区

[现在登录](#)

热门节点

[技术文章](#)

[社区小黑板](#)

目录

[RSS](#) [关于社区](#) [友情链接](#) [社区小黑板](#)