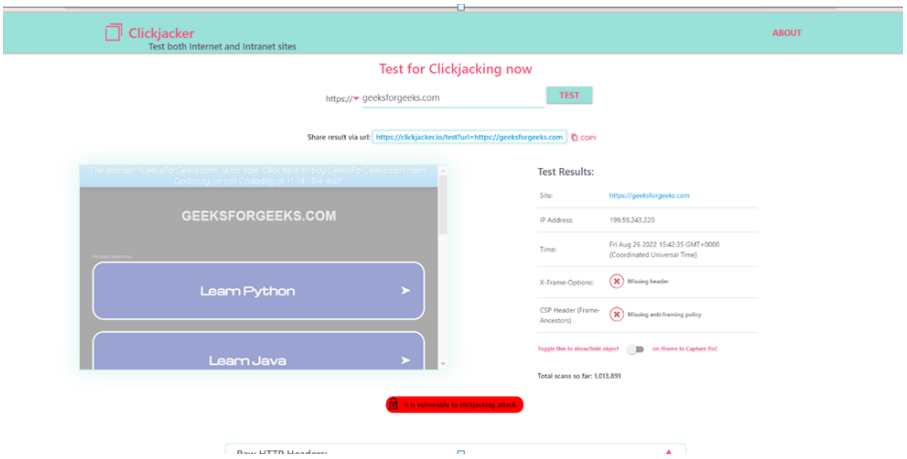
CYBER SECURITY

CLICK JACKING ON WEBSITES:

1. **geeksforgeeks.com**:



- Navigate to click jacker.io website and search for geeksforgeeks.com website
- click on test and it gives the result whether it is vulnerable to click jacking or not.

DESCRIPTION/EXPLANATION:

Clickjacking is a portmanteau of two words 'click' and 'hijacking'. It refers to hijacking user's click for malicious intent. In it, an attacker embeds the vulnerable site in an transparent iframe in attacker's own website and overlays it with objects such as button using CSS skills. This tricks users to perform unintended actions on vulnerable website, thinking they are doing those on attacker's website. Clickjacking, also known as a "UI redress attack".

COPY

IMPACT:

Users are tricked into performing all sorts of unintended actions are such as typing in the password, clicking on 'Delete my account' button, liking a post, deleting a post, commenting on a blog. In other words all the actions that a normal user can do on a legitimate website can be done using clickjacking.

COPY

STEPS TO REPRODUCE:

1. Go to this URL: https://clickjacker.io/test?url=<target site url here>
2. Observe that the website is getting embeded in an Iframe.
3. Observe that the headers x-frame-options and content-security-policy frame ancestors are missing.

COPY

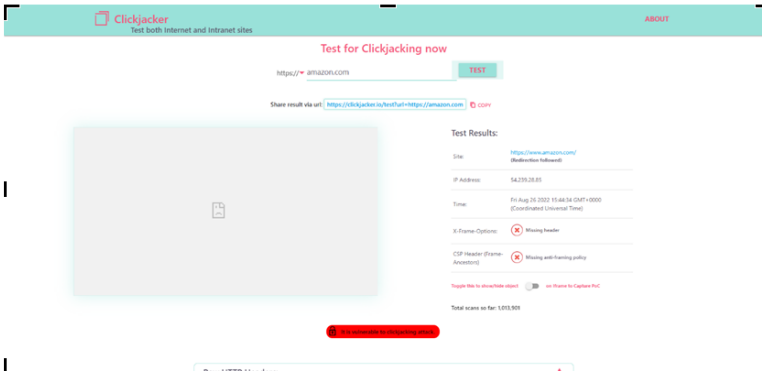
MITIGATION:

In order to fix the issue, we must know the underlying reason that is causing the issue. Clickjacking is caused due to allowing permission to a third party website to embed the vulnerabe site using Iframe. Disallowing this can be done by setting HTTP headers that direct browser to not allow the target website to be iframed. This can be done by configuring server on the following two response headers: X-Frame-Options Content-Security-Policy. Implement anyone of the below basaed on your business requirements:

1. Content-Security-Policy: frame-ancestors 'none' : Set this if you want to disallow every domain from embedding your site in an Iframe.
2. Content-Security-Policy: frame-ancestors 'self' : Set this if you want to disallow every domain from embedding your site in an Iframe and allow only your domain (i.e. the site itself) to embed itself in Iframe.
3. Content-Security-Policy: frame-ancestors uri : Set this if you want to allow a specifc uri to embed your site in an Iframe and disallow all the others.

COPY

2. Amazon.com:



- Navigate to click jacker.io website and search for amazon.com website
- click on test and it gives the result whether it is vulnerable to click jacking or not.

DESCRIPTION/EXPLANATION:

Clickjacking is a portmanteau of two words 'click' and 'hijacking'. It refers to hijacking user's click for malicious intent. In it, an attacker embeds the vulnerable site in an transparent iframe in attacker's own website and overlays it with objects such as button using CSS skills. This tricks users to perform unintended actions on vulnerable website, thinking they are doing those on attacker's website. Clickjacking, also known as a "UI redress attack".

COPY

IMPACT:

Users are tricked into performing all sorts of unintended actions are such as typing in the password, clicking on 'Delete my account' button, liking a post, deleting a post, commenting on a blog. In other words all the actions that a normal user can do on a legitimate website can be done using clickjacking.

COPY

STEPS TO REPRODUCE:

1. Go to this URL: https://clickjacker.io/test?url=<target site url here>
2. Observe that the website is getting embeded in an Iframe.
3. Observe that the headers x-frame-options and content-security-policy frame ancestors are missing.

COPY

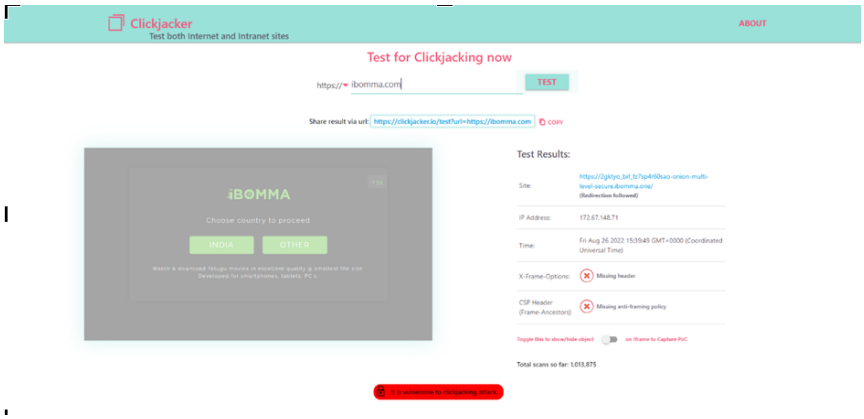
MITIGATION:

In order to fix the issue, we must know the underlying reason that is causing the issue. Clickjacking is caused due to allowing permission to a third party website to embed the vulnerabe site using Iframe. Disallowing this can be done by setting HTTP headers that direct browser to not allow the target website to be iframed. This can be done by configuring server on the following two response headers: X-Frame-Options Content-Security-Policy. Implement anyone of the below basaed on your business requirements:

1. Content-Security-Policy: frame-ancestors 'none' : Set this if you want to disallow every domain from embedding your site in an Iframe.
2. Content-Security-Policy: frame-ancestors 'self' : Set this if you want to disallow every domain from embedding your site in an Iframe and allow only your domain (i.e. the site itself) to embed itself in Iframe.
3. Content-Security-Policy: frame-ancestors uri : Set this if you want to allow a specifc uri to embed your site in an Iframe and disallow all the others.

COPY

3. **ibomma.com**:



- Navigate to click jacker.io website and search for ibomma.com website
- click on test and it gives the result whether it is vulnerable to click jacking or not

DESCRIPTION/EXPLANATION:

Clickjacking is a portmanteau of two words 'click' and 'hijacking'. It refers to hijacking user's click for malicious intent. In it, an attacker embeds the vulnerable site in an transparent iframe in attacker's own website and overlays it with objects such as button using CSS skills. This tricks users to perform unintended actions on vulnerable website, thinking they are doing those on attacker's website. Clickjacking, also known as a "UI redress attack".

COPY

IMPACT:

Users are tricked into performing all sorts of unintended actions are such as typing in the password, clicking on 'Delete my account' button, liking a post, deleting a post, commenting on a blog. In other words all the actions that a normal user can do on a legitimate website can be done using clickjacking.

COPY

STEPS TO REPRODUCE:

1. Go to this URL: https://clickjacker.io/test?url=<target site url here>
2. Observe that the website is getting embeded in an Iframe.
3. Observe that the headers x-frame-options and content-security-policy frame ancestors are missing.

COPY

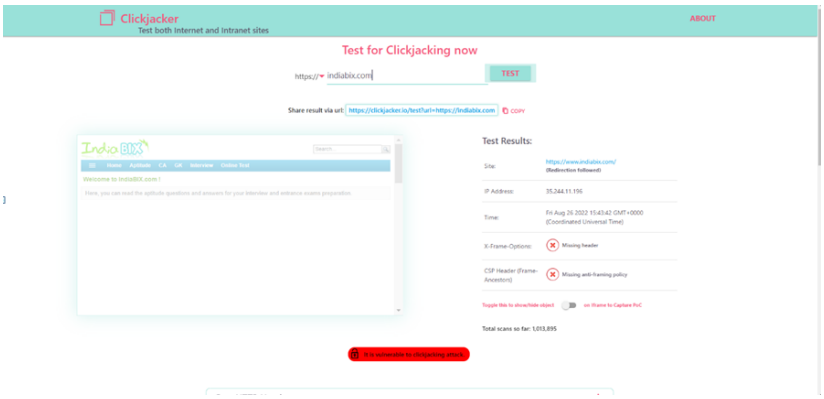
MITIGATION:

In order to fix the issue, we must know the underlying reason that is causing the issue. Clickjacking is caused due to allowing permission to a third party website to embed the vulnerabe site using Iframe. Disallowing this can be done by setting HTTP headers that direct browser to not allow the target website to be iframed. This can be done by configuring server on the following two response headers: X-Frame-Options Content-Security-Policy. Implement anyone of the below basaed on your business requirements:

1. Content-Security-Policy: frame-ancestors 'none' : Set this if you want to disallow every domain from embedding your site in an Iframe.
2. Content-Security-Policy: frame-ancestors 'self' : Set this if you want to disallow every domain from embedding your site in an Iframe and allow only your domain (i.e. the site itself) to embed itself in Iframe.
3. Content-Security-Policy: frame-ancestors uri : Set this if you want to allow a specifc uri to embed your site in an Iframe and disallow all the others.

COPY

4. Indiabix.com:



- Navigate to click jacker.io website and search for indiabix.com website
- click on test and it gives the result whether it is vulnerable to click jacking or not

DESCRIPTION/EXPLANATION:

Clickjacking is a portmanteau of two words 'click' and 'hijacking'. It refers to hijacking user's click for malicious intent. In it, an attacker embeds the vulnerable site in an transparent iframe in attacker's own website and overlays it with objects such as button using CSS skills. This tricks users to perform unintended actions on vulnerable website, thinking they are doing those on attacker's website. Clickjacking, also known as a "UI redress attack".

COPY

IMPACT:

Users are tricked into performing all sorts of unintended actions are such as typing in the password, clicking on 'Delete my account' button, liking a post, deleting a post, commenting on a blog. In other words all the actions that a normal user can do on a legitimate website can be done using clickjacking.

COPY

STEPS TO REPRODUCE:

1. Go to this URL: https://clickjacker.io/test?url=<target site url here>
2. Observe that the website is getting embeded in an Iframe.
3. Observe that the headers x-frame-options and content-security-policy frame ancestors are missing.

COPY

OR

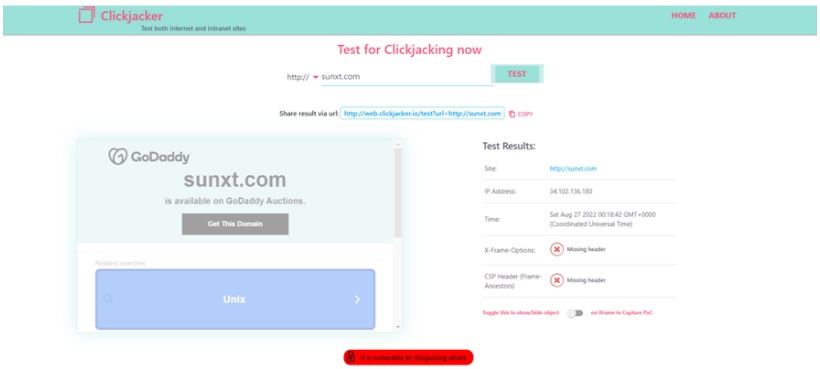
MITIGATION:

In order to fix the issue, we must know the underlying reason that is causing the issue. Clickjacking is caused due to allowing permission to a third party website to embed the vulnerabe site using Iframe. Disallowing this can be done by setting HTTP headers that direct browser to not allow the target website to be iframed. This can be done by configuring server on the following two response headers: X-Frame-Options Content-Security-Policy. Implement anyone of the below basaed on your business requirements:

1. Content-Security-Policy: frame-ancestors 'none' : Set this if you want to disallow every domain from embedding your site in an Iframe.
2. Content-Security-Policy: frame-ancestors 'self' : Set this if you want to disallow every domain from embedding your site in an Iframe and allow only your domain (i.e. the site itself) to embed itself in Iframe.
3. Content-Security-Policy: frame-ancestors uri : Set this if you want to allow a specifc uri to embed your site in an Iframe and disallow all the others.

COPY

5. sunxt.com:



- Navigate to click jacker.io website and search for sunxt.com website
- click on test and it gives the result whether it is vulnerable to click jacking or not.

DESCRIPTION/EXPLANATION:

Clickjacking is a portmanteau of two words 'click' and 'hijacking'. It refers to hijacking user's click for malicious intent. In it, an attacker embeds the vulnerable site in an transparent iframe in attacker's own website and overlays it with objects such as button using CSS skills. This tricks users to perform unintended actions on vulnerable website, thinking they are doing those on attacker's website. Clickjacking, also known as a "UI redress attack".

COPY

IMPACT:

Users are tricked into performing all sorts of unintended actions are such as typing in the password, clicking on 'Delete my account' button, liking a post, deleting a post, commenting on a blog. In other words all the actions that a normal user can do on a legitimate website can be done using clickjacking.

COPY

STEPS TO REPRODUCE:

1. Go to this URL: https://clickjacker.io/test?url=<target site url here>
2. Observe that the website is getting embeded in an Iframe.
3. Observe that the headers x-frame-options and content-security-policy frame ancestors are missing.

COPY

OR

MITIGATION:

In order to fix the issue, we must know the underlying reason that is causing the issue. Clickjacking is caused due to allowing permission to a third party website to embed the vulnerabe site using Iframe. Disallowing this can be done by setting HTTP headers that direct browser to not allow the target website to be iframed. This can be done by configuring server on the following two response headers: X-Frame-Options Content-Security-Policy. Implement anyone of the below basaed on your business requirements:

1. Content-Security-Policy: frame-ancestors 'none' : Set this if you want to disallow every domain from embedding your site in an Iframe.
2. Content-Security-Policy: frame-ancestors 'self' : Set this if you want to disallow every domain from embedding your site in an Iframe and allow only your domain (i.e. the site itself) to embed itself in Iframe.
3. Content-Security-Policy: frame-ancestors uri : Set this if you want to allow a specifc uri to embed your site in an Iframe and disallow all the others.

COPY