CSRF

- في الـ HTML في Tag اسمه Iframe ده بيسمحلك تضيف موقع جوا موقع تاني, ولكن التواصل بينهم ممنوع بسبب الـ SOP الي بتمنع التوصل بين 2 دومين إلا لو توافرت الـ 3 شروط الي عارفينهم (تقدر ترجع لثغرة CORS misconfigrationوهتفهم),

تعريفها هي اختصار لـ Cross Site Request Forgery ومعناها إن موقع site/بيزيف طلب لموقع site/ تاني بهدف تغيير قيمة معينة أو إنه يعمل action معين, أو بصيغة تاني وهو إن في يوزر بيجبر يوزر تاني إنه يعمل action هو مش عايز يعمله وبدون علمه زي إنه يحذف الاكونت أو يضيف رقم الـ Attacker أو يضيف اميله وهكذا.

- المعروف إنك طالما Logged in في موقع ما, فـ بيكون في cookies في المتصفح محفوظة بتتبعت كل ما تعمل ريكوست داخل الموقع ده, يعني هتزور جروب ف بيبعت الريكوست, ومعاه الكوكيز عشان يحدد انت مين, لو هتحذف اكونت بيبعت الكوكيز مع الريكوست عشان السيرفر يعرف هيحذف اكونت مين وهكذا, الي بتعمله الـ CSRF إن الـ Attacker بيبعت لينك موقع غير الموقع الي انت عامله logged in, بمجرد ما تزور الموقع التاني هو بيعمل Request للموقع الي انت عامل لوجين عليه ويعمل ريكوست لـ endpoint حذف الاميل وبالتالي المتصفح بيبعت مع الريكوست الكوكيز بتاعتك وبالتالي بيتحذف الاكونت بدون ما تعرف.

خطورتها الخطورة في معظم الأحيان بتؤدي إلى Account takeover زي إن الـ Attacker هيغير البريد الالكتروني للضحية للبريد الالكتروني بتاعه, او يغيرله الباسورد أو يضيف رقم موبايله وبالتالي يعمل reset passwordعن طريق الموبايل وياخد اميل الضحية, كمان لو الضحية كان الادمن كده المهاجم هياخد full access على الموقع.

شروط حدوث الثغرة

-1إن يكون في vulnerable action هيستغله الـ Attacker زي تغيير الباسورد وغيره.

-2إن الموقع يكون معتمد على الـ Session Cookies في تحديد مين اليوزر الفلاني.

-3إن ميكونش في بارامتر إضافي بيحتاج تخمين لإنه بيكون صعب على الـ Attacker إنه يعرف قيمة البارامتر ده زي مثلا الباسورد القديم وهكذا.

ازاي تنفذ الهجوم

- عن طريق إنك هت create CSRF Poc عن طريق الـ Burp أو يدوي عادي وتضيف فيه خاصية الـ auto submitيعني بمجرد ما الضحية يزور الصفحة الكود هيتنفذ, وبعدين كل الي عليك تبعت اللينك لليوزر سواء في اميل او عن طريق رسالة على مواقع السوشيال ويُفضل إستخدام الـ Social Engineering عشان تقنع الضحية إنه يدوس على اللينك.
- ### لو الـ Action الـمصاب بـ CSRF بيستخدم GET Method ف انت مش محتاج تعمل الخطوة الي فوق, كل الي محتاجه تضيف tag الـسورس بتاعه بيعمل الـ Action الي هو تغيير الاميل مثلا زي – img , iframeأو لو في XSS ف تقدر تخزن الـ CSRF Attack كـ payload في الـ XSS وبالتالي الخطورة بتزيد

جداً لإن اليوزر مش محتاج يعمل حاجة هو بمجرد ما يزور الصفحة المصابة في الموقع الي هو authenticatedفيه هيتنفذ الهجوم وهنا الـ CSRF بتُسمى Stored CSRF.

CSRF Vs XSS

- الـ CSRF مقتصرة على بعض الأماكن الي بيقدر يعملها اليوزر, لكن الـ XSS بتلاقيها في أماكن كتير جداً.
- الـ CSRF بتصنف إنها one-way يعني من ناحية واحدة وهو إنها بتمكنك تعمل request ولكن مش بتشوف الـ response على عكس الـ XSS الي بتعمل request وتشوف الـ response وتستخرج الداتا وتبعتها لموقع خارجي تابع ليك.
- ###الـ CSRF Tokens بتقدر تمنع الـ Reflected XSS لان كل طلب بيتطلب token جديد وبالتالي الـ Attackerصعب يخمن التوكين ولكن مش بتمنع الـ Stored XSS.
- ###لو في Reflected XSS في مكان معين ومش محمية بـ CSRF Tokens ف تقدر من خلالها تستغل أماكن محمية بـ CSRF Tokens وده عن طريق إنك تستغل الـ XSS الي مش محمية بإنها تجيب valid CSRF Tokenوبعدين تستخدمه في المكان المحمي.

CSRF tokens

- الـ tokens دي عبارة عن قيمة بتكون سرية وفريدة وغير متوقعة وغير قابلة للتخمين, بيتم إنشائها عن طريق السيرفر وبعدين بيرسلها للـ client-side بهدف استخدامها فيما بعد, ف لما يجي اليوزر يعمل action معين بيرسل معاه التوكين ده وبيتأكد السيرفر إن التوكين موجود ولا لا, ولو موجود هيكون valid ولا لا, في أفضل الحالات ف السيرفر المفروض يرفض أي invalid token أو أي ريكوست مفهوش توكين من الأساس.
- في وجود التوكينز بقى صعب على الـ Attacker إنه يعمل fully valid request يقدر من خلاله يخلي اليوزر يعمل action معين, وده بسبب إنه مش هيقدر يعرف التوكين بتاع اليوزر خالص.
- بيتم نقل التوكين من السيرفر للـ client-side عن طريق إنها بيعملها hidden input بتكون موجودة جواه, وبتتبعت مع الريكوست بعدين.
- ###في حاجة كمان من خلالها نقدر نمنع الـ CSRF Attacks وهي الـ SameSite Cookies
  - ✓ هي attribute بيتضاف لما السيرفر يـ set-cookie في الـ Response header, ف بيحط مع الكوكي الـ attributeده وبياخد قيمة حاجة من اتنين: يا strict ودي بتمنع إضافة الكوكي لاي ريكوست جاي من دومين خارجي وبالتالي ده آمنة جداً بس مشكلتها في الـ Third party وإنك بتحتاج تكتب الـ credentials من جديد وده بيصعب الخدمة للمستخدم, إنما القيمة التانية بتكون Lax وهنا بيوافق إنها الكوكي تتبعت عادي ولكن بشرطين تكون الميثود GET والـ action يكون بواسطة Top-Level navigation زي إنه يدوس علي لينك أو حاجة داخل الموقع, إنما الي بتكون بواسطة scripts ف دي مش هيبعت معاها الـكوكيز.
- في حال الموقع بيضيف token للريكوست ازاي اعملbypass ؟

-1لو التوكين static في تقدر تبعتها مع الريكوست عادي.

-2لو التوكين قصير ويمكن تخمينه ف جرب عليه brute force.

-3لو السيرفر مش بيعمل validation أصلا على التوكين, وتقدر تجرب إنك تمسح التوكين وتشوف الريكوست هيكمل عادي ولا لا.

أماكن تواجد الثغرة (سيناريوهات)

- الثغرة دايما بتبقى موجودة في الأماكن الحساسة زي:
- تغيير الباسورد والاميل
- إضافة اميل أو رقم تليفون
- شراء شيء معين من الموقع
- إضافة يوزر كأدمن في صفحة أو في جروب
- حذف أو الخروج من خدمة معينة أو جروب وهكذا.
- رفع صورة/ملف/فيديو وتجبر باقي اليوزرز إنهم يرفعوها عندهم.
- عمل لايك على كومنت وتجبر باقي اليوزرز يعملوا لايك أو كومنت.
- في سيناريو بيحصل إن لما يكون في function إنك تضيف الفيس والتويتر وكده ف بتـ create لينك لما الـ victimيدوس عليه بيروح يضيف أميلك عنده وبالتالي تقدر تعمل reset password وبقى ليك full access على اميل الـ victim.
- إرسال اميل من الموقع أو هدية لحد وكل الـ Functions دي.
- الخروج من جروب أو الدخول ف جروب أو شات أو عمل لايك لصفحة أو بوست وهكذا.
- عمل بان أو حذف لعضو في جروب أو شات أو برودكاست أو الموافقة ع طلبه.
- لما تعمل subscribe لحاجة في الموقع أو تقبل عرض ف كده بتجبر اليوزر يقبل العرض غصب عنه.
- تحميل تطبيق عن طريق SMS بكتابة رقم الفون.
- أي Function فيها حذف إضافة تعديل (زي جروب, صفحة, بيانات شخصية, صورة, فيديو, بوست, كومنت, اميل, بطاقة شخصية, بطاقة ائتمان, أي Action يقدر يعمله أكتر من يوزر وهكذا(
-

طرق تخطي الحماية (Bypass)

1-إنك تغير الريكوست من GET => POST وتشيل التوكين لإنه أحيانا بيعمل validation لما تكون الريكوست POSTبس.

2-إنك تحذف التوكين اصلا, لإنه احيانا بيعمل validation لما التوكين يكون موجود ولو مش موجود بيعدي الريكوست عادي.

3-تاخد تكوين يوزر تاني وتجرب, لإنه احيانا السيرفر بيـ create كمية كبيرة من التوكينز ولما يجيله ريكوست بتوكين معين هو بيـعمل check ان التوكين ده واحد من الي هو عملهم create ولا لا مش اكتر يعني مش مهم التوكين ده تبع مين.

4- بعض الـ CSRF Attacks بتتأكد من الـ Referer Header وتشوف لو خارج الدومين بتلغي الريكوست, واحيانا السيرفر بيعمل check لو هو موجود بس, لو مش موجود بيعدي الريكوست ف الـ Tip دي بتخليك تحذف الـ Referer Headerفي ريكوست الضحية عن طريق:

<meta name="referrer" content="never">

5- أحيانا بيكون في CSRF token في الـ cookie وكمان CSRF كـ بارامتر وهنا في 2 سيناريو:

1. إنه يكون بيعمل check إن البارامتر valid والتاني نفس الكلام بدون أي ربط ما بينهم، يعني أي 2 CSRF tokens هيتم الموضوع.

2. إنه يكون لازم الـ 2 tokens مربوطين ببعض وبالتالي لازم الـ2 يكونوا valid, related
3. إن يكون الاتنين نفس القيمة وهو بيعمل check إن الاتنين نفس القيمة وبس، وبالتالي تقدر تحط أي قيمتين من عندك المهم يكونوا نفس الـ size ونفس الـ value.

#- لو بتكتب قيمة ما والتطبيق بياخدها ويحطها فالـ cookie تقدر وقتها إنك تتلاعب بالـ cookie وتحط قيمة جديدة خالص عن طريق الـ :%0d%0aSet-Cookie وتكتب الي عايزه.

-5لو بيعمل check على الـ Referer Header ولازم يكون موجود جرب تكتب اي صبدومين قبل الدومين الاساسي أو تضيفه كـ TLD بعد الدومين الاساسي أو تخليه كـ parameter في الـ URL

http://attacker-website.com/csrf-attack?vulnerable-website.com

http://vulnerable-website.com.attacker-website.com/csrf-attack

-6لما تجرب تغيير حاجة معينة ويكون حاطط الاميل كنوع من الحماية ضد الـ CSRF انت كده محتاج تكتب اميل الضحية كل ما تيجي تنفذ الـ attack ف جرب تشيل بارامتر الاميل كله أو تشيل الـ value بس أو تكتب random value.

-7لو الـ CSRF attack بيتنفذ بـ PUT Method وجربت POST ومش بيقبلها ف جرب تضيف في آخر الـ URL ?_method=PUT <=وتكون POST هي الرئيسية وكذلك الـ Delete Method زي الـ.PUT

-8لو لقيت CSRF json حاول تغيرها لـ Regular parameters وتشوف التطبيق هيقبلها ولا لا, لو قبلها ممكن تنفذ.CSRF

-9لو بتنفذ هجوم CSRF وكان في token مش راضي لا يتحذف ولا حذف القيمة بتاعته ولا تغييرها حتى, ف جرب تعمل account تاني وتاخد الـ token بتاعته وتجرب.

-10احيانا بيكون في CSRF_token في الـ Body بتاع الريكوست و CSRF_tokenفي الـ Headers والاتنين نفس القيمة واحيانا هنا السيرفر بيتشك على إن القيمة في الاتنين واحد يعني لو كتبت واحدة عشوائية في الاتنين بيعديها.

-11ممكن تعمل Bypass للـ Referer Header عن طريق ثغرة(iframe injection (HTML Injection

-12ممكن تجرب تحط قيمة الـ.CSRF token = undefined

-13ممكن الـ CSRF token عبارة عن => array زي=[]xsrf_token

-14لو مفيش أي طريقة لإنك تـ bypass الـ CSRF token حاول تشوف ريسبونس فيها التوكين (ادخل على صفحات التعديل والـ setting وهكذا) يعني بتعمله discolse وبالتالي أنت ممكن تعمل cilickjacking وتسرق التوكين بتاع الضحية وتنفذ الـ.CSRF Attack

15-

طرق تخطي الـ JSON

-1ممكن تجرب بإنك تخلي الـ JSON Payload هو الـ Input name نفسه زي كده والتكنيك بيتم عن طريق إنك تضيف = بعد الـ JSON Paylaod في الريكوست واحيانا السيرفر بيعمل reject للريكوست لما بيلاقي.=

- مهم إنك تخلي الـencrypte="text/plain"

```
1:  <html>
2:  <form action=http://192.168.1.41:3000 method=post enctype="text/plain" >
3:  <input name='{"a":1,"b":{"c":3}}' type='hidden'>
4:  <input type=submit>
5:  </form>
6:  </html>
```

-2عن طريق إنك تضيف parameter زيادة في الـ JSON زي ده

```
1:  <html>
2:  <form action=http://192.168.1.41:3000 method=post enctype="text/plain" >
3:  <input name='{"a":1,"b":{"c":3}, "ignore_me":"' value='test"}'type='hidden'>
4:  <input type=submit>
5:  </form>
6:  </html>
```