|=| Access Control |=|

|=| تعريفها |=|

الـ Access control باختصار هو الي بيحدد بناءاً على دور اليوزر هو المفروض يوصل لايه ويقدر يعمل ايه وهو بيعتمد على المصطلحات الجاية:

لازم قبل ما نبدأ في الثغرة نعرف كام مصطلح مهمين:

Authentication: ده معناه إنه بيعرف مين الـ user ده، هو الي بيفرق بيني وبين user تاني, بالبلدي بيقولك انت مين؟

Session management: ده بيحدد كل الـ Requests الي هتتعمل بعد كده بنفس اليوزر وتحفظ إنه ده بيعمل كذا وعمل كذا وهكذا 😆

Access Control: دي بقى بتحدد اليوزر ده يقدر يعمل إيه وايه الي ممنوع يدخله او يعمله.

|=| أنواعها |=|

Vertical access control: دي بقى الي بتقسم المهمات على المستخدمين وبتحدد دور كل واحد جوا الموقع, يعني مثال ع جروب الفيس ف هي الي بتحدد إن الادمن يقدر يحذف ويضيف أعضاء وكمان يقدر يخلي في ادمنز معاه, على عكس اليوزر العادي الي ميقدرش يعمل كده وصلاحياته بتتوقف على كتابة بوست او حذفه وهكذا.

Horizontal access control: دي بقى مهمتها إنها بتفصل كل يوزر عن التاني, يعني ايه بردو ؟ يعني مثلا إنك تشوف بياناتك الحساسة في الفيس زي الاميل والبطاقة الائتمانية ورقمها والفون وكل ده بس بدون ما تشوف كل الي فوق ليوزر تاني , يعني ده بيحافظ إن كل يوزر مش هيجي جنب التاني.

Context-dependent access control: دي بتقول إن خطوات حاجة معينة بتكون 1 2 3، يعني بتحدد سير الـ functions في الموقع, يعني لو هتشتري حاجة ف انت هتضيفها للـ cart بتاعتك وتدفع وبعدين يخصم من الفيزا بتاعتك وتتم العملية، مينفعش تتجاوز خطوة في النص زي مثلا إنك متدفعش.

|=| Broken Access Control |=|

|=| تعريفها |=|

معناها إنك بتقدر تـ access resources انت مش privileged إنك تـعملها access أصلا أو إنك تقوم بدور هو مش دورك أو تعمل حاجة، وبعض أنواع الثغرة دي:

- إن يكون في إمتيازات (privileges) لحد معين زي مثلا Admin أو أي role عالي ولكن الـ privileges دي بتكون متاحة لأي حد.
- إنك تعمل Bypass access control عن طريق إنك بتعدل الـ request سواء من الـ URL، صفحة الـ HTML، أو أداة تعدل الـ API requests.
- إنك تعرض صفحة شخص أو يوزر تاني أو تعدلها من خلال إنك بتغير الـ ID بتاعك لـ ID الشخص ده (IDOR)
- إنك توصل للـ API بدون access control موجود على الـ POST, PUT, DELETE methods.
- إنك تتعامل كيوزر وأنت أصلا مش logged in أو تتعامل كـ Admin وأنت يوزر عادي، وده بيُسمى رفع الامتياز.
- الـ CROS misconfiguration بتسمح للـ API إنه يـ access عن طريق untrusted origins.
- الوصول لصفحات مش المفروض إنك توصلها وأنت يوزر عادي، أو صفحات لازم تكون login وأنت مش مسجل دخول.

|=| أنواعها |=|

Vertical Privilege Escalation: ودي بقى إنك تـ **access Functionality** هي مش دورك أصلا, زي إنك تقدر تبقى ادمن وتحذف أعضاء وتضيف براحتك.

|=| سيناريوهات |=|

- من المشهور في النوع ده هو إنك تقدر تـ access الـ Admin panel وانت المفروض مش دورك ده, التطبيق مش بيفلتر صح مين الي ليه access عليها فـ انا كيوزر عادي هكتب اللينك وادخل ع لوحة التحكم واعمل الي عايزه, لكن لو في فلترة صح ف هيمنع إن اشوف الصفحة إلا لو كنت Admin فعلا.
- أحيانا حماية الـ functions زي انك تقدر تـ access الـ Admin panel بتكون مش محمية اوي ولكن هما بيحاولوا يخلوها محمية عن طريق إنهم بيغيروا اسمها فـالـ URL يعني بدل ما يكون admin/ ف الاخر لا يكون admin-erw4d5/ وبالتالي صعب جدا على الـ Attacker إنه يخمن الـ URL ده وفي الحالة دي بتلجأ إنك تقرأ ملفات الـ JavaScript كويس او ملف الـ robots.txt.
- أحيانا بردو التطبيق بيحددلك دور بعد ما تسجل دخول ويحفظهالك في الـ Cookies، وبناءا على الـ role دي انت بتستخدم الموقع سواء كيوزر عادي او ادمن , طيب نفرض إن الموضوع عبارة عن subsequent number يعني لما تسجل يكون الـ id=10 وهكذا بالتسلسل ف لو جربت الـ id=1 هتلاقي نفسك ادمن.
- كمان ركز في البارامترز لما بتيجي تعمل action معين زي إنك تضيف اميل او تحذف حد وانت صلاحيتك اعلى منه واقل من الـ Admin او انك تضيف بوست وهكذا , ممكن يكون في بارامتر بيدل عليك كيوزر حالي ف لما تغيره الـ action هيتنفذ كـ Admin

- ممكن يكون في بارامتر اسمه admin وده بيكون ليه قمتين true / false لو لقيته وكان false ده معناه إنك بتعمل الـ action ده كيوزر عادي ف لو خليته true تبقى ادمن، وف الغالب مش هيكون البارامتر اسمه كده أو بالوضوح ده، ف حاول تلعب مع البارامترز وتبص على الـ Response.
- أحيانا وانت بتيست انك تدخل على لوحة التحكم بتاعة الادمن أو أي URL بيحمي حاجة معينة بيقولك access denied هنا انت بتلجأ إنك تستخدم الـ Header الي اسمه X-Original-URL وبتحط فيه الـ URL الي محتاج تدخل عليه ولكن أي بارامتر بيفضل ف المسار بتاعه، وده بسبب إن الـ Framework في الباك اند بيسمح بالـ Header ده وبالتالي بتقدر تعمل Bypass.
- أحيانا وأنت بتحاول تـ access URL معين يقولك access denied ف جرب تغير الـميثود سواء من GET لـ POST أو العكس.
- وانت بتيست كـ Admin وبتجرب الـ Functions الي الادمن بس يقدر يعملها, جرب تغير السيشن بتاعة الادمن بيوزر عادي وتشوف هينفع ولا لا, وممكن تغير الريكوست من POST لـ GET او العكس, يعني وانت بتحذف عضو من جروب فيس مثلا جرب تغير السيشن بيوزر تاني عادي في الجروب وتشوف هينفع ولا لا.
- نفس كلام النقطة الي فوق بس تجرب تـ access endpoint مش ليك access عليها.

Horizontal Privilege Escalation: دي بقى إنك بتقدر تـ **access Resources** بتنتمي ليوزر تاني, زي معلوماته الشخصية والحاجات دي، وده مش مسموح بالنسبالك.

- أحيانا بعض المواقع بيكون فيها Public profile لكل يورز وتقدر تشوفه وتتصفحه براحتك، وفي private profile الي بيكون فيه المعلومات الحساسة، ف مش معنى إنك بتقدر تغير البارامتر الي ف الـ URL ف حالة الـ Public profile ده معناه إنها ثغرة، حاول تفهم التطبيق كويس عشان تفرق ما بين إيه الي عادي وإيه الي مش عادي.

|=| أماكن تواجدها |=|

- لما تبعت رسالة أو كومنت أو بوست في جروب معين أو بوست في صفحة، تقدر تجرب النوع ده من الثغرات.
- في عمل upload لملف معين ووقتها تقدر تغير الـ ID وتـ upload local file.
- أي action بيعمله الـ admin بس جرب تعمله باليوزر العادي عن طريق إنك تبدل السيشن بتاعة الادمن باليوزر.
- أي default action حاول تخرج برا دائرة الـ default وتلعب معاه, لو المفروض ترفع صورة واحدة حاول ترفع صورتين مع بعض, لو مسموح تكتب كومنت واحد حاول تكتب أكتر, وهكذا.

|=| سيناريوهات |=|

- أول سيناريو معروف وهو انك لما تدخل على الـ setting بتاعة اميلك وتكون واخد ID برقم زي id=15 أو يكون id=taha هنا بتقدر تغير الرقم او الاسم وتدخل ع الـ setting بتاعة اليوزر ده، وده نادر إنك تلاقيه.

- أحيانا بقى مش بيتسخدموا إن يكون الـ id فيه رقم وده نادرا جدا, ف بيستخدموا حاجة اسمها (GUIDs) ودي بتبقى رقم عشوائي صعب تخمنه او توصله, طيب والحل؟ الحل إنك هتحاول تدور عليه في مكان ما , اكيد الموقع بيكون بيعرضه في حتة ما تبع اليوزر ده, يعني ولنفترض صفحة تغيير الاميل في الفيس بتبقى على حسب الـ id وبعده GUID ف انت صعب تخمنه وتعرف الـ GUID الخاص بالضحية, ف طبعا ممكن تجيبه من صفحته الشخصية ممكن يكون معروض وبالتالي تقدر تغير اميله.
- ساعات كتير وانت بتحاول تغير الـ id سواء رقم او اسم يقولك unauthoraized وإن ده مش مسموح لك, بس شوف الـ response أحيانا بيكون فيها Leaked data بخصوص اليوزر.
- يمكن تحويل الـ Horizontal للـ Vertical في بعض الأحيان, زي مثلا إنك توصل لاكونت ادمن ومن خلاله تقدر تعمل الي بيقدر يعمله الادمن.

|=| Code review for Broken access control |=|

- الثغرة دي من الثغرات الي مش بتعتمد على الـ input ولما يحصل validation هي هتتقفل، لا دي بتعتمد بشكل كبير على الـ Logic بتاع الموقع، وبالتالي الـ Code review للثغرة دي هي إنك تفضل تلعب مع الموقع كـ يوزر عادي وتخبط ف كل حتة وبعدين تفضل تغير في الـ Requests وتشوف الـ response ايه اتغير أو إيه حصل.

|=| أزاي تعمل تيست ليها |=|

- في خلال مرحلة الـ Privileges escalation testing كل المطلوب منك إنك تتأكد إن الـ user مش هيقدر يعدل صلاحياته لحاجة أعلى.
- زي ما عرفنا إن الثغرة بتحصل لما user يكون بيعمل action أو بيقدر يوصل لـ resources هو مش مسموح ليه بكده، والمفروض هنا التطبيق يمنعه، وده بيحصل بسبب خطأ في الـ application.
- درجة رفع الصلاحيات بتختلف تبعا لصلاحيات الـ attacker والصلاحيات الي يقدر ياخدها لما ينفذ الهجوم صح، زي مثلا إنه يكون يوزر ويوصل لأدمن ف هنا هو معاه صلاحيات ولكن أقل، غير لما يكون هو مش authenticated اصلا ويوصل لإنه يعمل action as admin وهو مش معاه أي صلاحيات.
- دايما معروف لما يحصل إنك تعلي الصلاحيات من يوزر أقل إلى يوزر أعلى بيُسمى Vertical privilege escalation زي ما في الصورة، لو Bob, Alice بقى حد فيهم آدمن ف ده vertical، إنما إن Bob يوصل لـ resources تبع Alice ف ده horizontal privilege escalation.

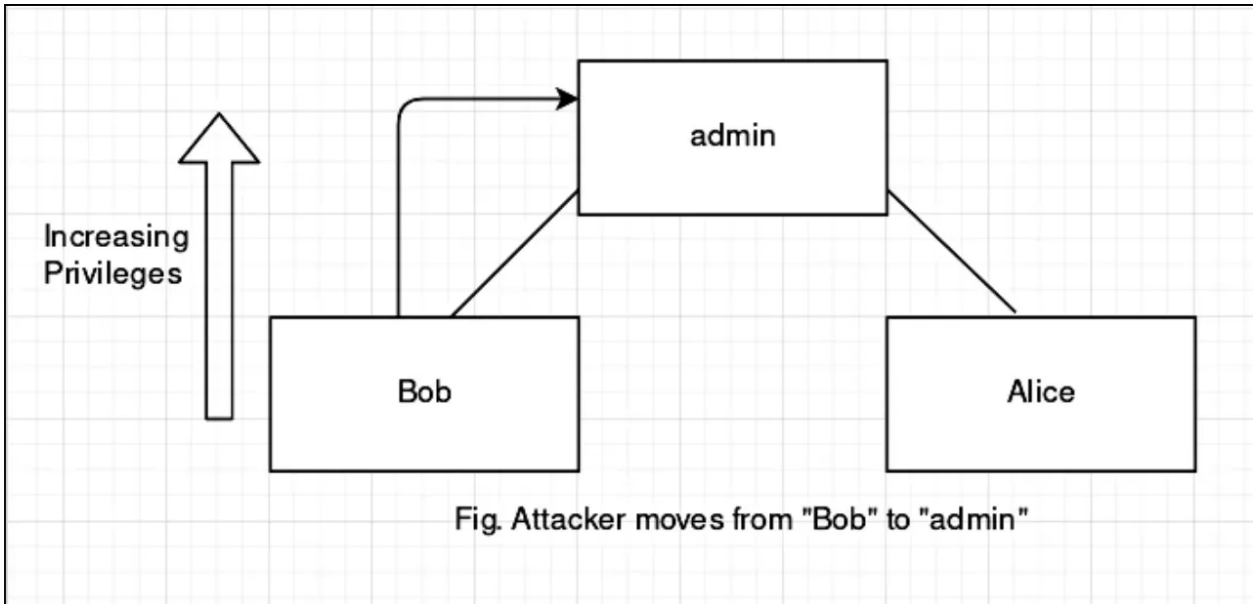

Fig. Attacker moves from "Bob" to "admin"

- في كل جزء في التطبيق بتقدر تضيف داتا للـ database (تبعت رسالة، تعمل إجراء دفع، تضيف حد صديق)، تستقبل داتا من الـ database (صفحتك الشخصية، معلومات عن order عملته)، أو تحذف حاجة من الـ database (رسالة، عضو من جروب، صفحة، صديق، ...)، مهم إنك تسجل كل الـ functions دي وبعدين تحاول توصلها بيوزر تاني يكون مش privileged إنه يوصلها، أو تحاول توصل منها لـ resource يوزر تاني.
- وهنا بعض السيناريوهات لإن لو كبتها المقالة هتكون طويلة جداً ف مفيش داعي **Scenarios**
- وعشان نتيست الـ vertical privilege escalation محتاجين كل page, request, function ونشوف لو نقدر نعملها access رغم إنها لـ higher privileged user.
- ممكن تمشي على الخطوات دي:
  - تسجل أكونت لكل Role موجود في الموقع زي normal user, admin
  - تحفظ لكل أكونت سيشن، وبعدين تفضل تغير سيشن اليوزر الأعلى باليوزر الأقل وتشوف لو تقدر تعمل نفس الـ action.
- في بعض التطبيقات بتستخدم Weak algorithm session زي مثلا إنه يستخدم (MD5(Password + UserID كـ session ID، وبالتالي سهل عليك كـ tester تخمنها أو تعملها Brute force.

Testing for Privilege Escalation

سيناريوهات عشوائية:

- جرب تضيف يوزر لصفحة او جروب وتخليه كآدمن وبعدين تشيله وتشوف هل هيتحذف ع طول من الجروب او الصفحة ولا يقدر يعمل أي حاجة لحد ما يخرج من الجروب؟.
- جرب وانت ف جروب او أي حاجة فيها أعضاء تكون مغلقة او سرية (مخصصة لمجموعة معينة), تحاول تدخل بيوزر مش موجود ف المجموعة ولو قلك unauthorized جرب تدخل بالـ Cookies بتاعة عضو موجود جوا.
- لو في URL ميقدرش غير الـ admin يعمله access جرب بيوزر عادي إنك تعمل نفس الـ action بتاعه.
- جرب تدخل على صفحة داخلية وانت مش authenticated ف هيطلب تسجيل دخول, جرب تعمل LFI في الـ path ولو فشل تدوس لوجين وتشوف هيدخلك ولا لا.
- احظر يوزر من جروب أو صفحة أو بوست وشوف لو اليوزر لسه قادر يدخل الجروب أو يزور الصفحة وهكذا.
- حاول تـ access endpoint أنت مش ليك access عليها وشوف هتقدر تشوفها ولا لا زي /admin.
- اعمل انفيت لحد لجروب أو صفحة أو شات ومن خلال الرابط هيقدر يدخل بعدين جرب تشيله وترجع تدوس بالاميل التاني على الرابط وتشوف هتقدر تدخل تاني ولا لا.
- لو في صفحة أنت ليك read-only permission حاول تغير الريكوست سواء من GET لـ POST أو العكس أو لـ PUT ونفس الكلام لو صفحة مش عارف تدخلها جرب تغير الريكوست.
- لو في 2 يوزر بـ permissions مختلفة جرب تتصفح الموقع كله بالـيوزر الأعلى وبعدين تفضل تبدل السيشن بتاعته بالسيشن بتاعة اليوزر الأقل صلاحية.
- لو بتحاول تـ access الادمن endpoints ف جرب تشوف الـ documentation أو تجرب الـ API المشهورة زي api/v2/members أو member أو user أو users وهكذا.

|=| IDOR |=|

|=| تعريفها |=| دي اختصار لـ Insecure Direct Object Refrences, ودي نوع من انواع ثغرات الـ Broken Access Control وهي شبيهة او مرتبطة أكتر مع الـ Horizontal access control يعني تـ Access resources.

|=| أسباب حدوثها |=|

- بتحصل نتيجة ان التطبيق بيستدعي او بيشاور على object معين بطريقة مباشرة
    - (user's file database – resources for website)
- دي بقى بتحصل لما يكون عندي بارامتر انا مقدرش اغيره -ده المفروض- ولو انا غيرته بيؤدي لثغرة IDOR زي مثلا URL بالشكل ده https://www.example.com/change_email?id=153 هنا البارامتر id بيدل عليا انا كيوزر ولو غيرته احتمال اغير اميل حد تاني.
- بالبلدي لما يكون عندي بارامتر فيه قيمة بتشاور على resource او معلومات خاصة بيا واغير البارامتر لقيمة تاني ف يديني resource خاصة بيوزر تاني او معلومات عنه.

|=| أنواعها |=|

Blind: دي الي بتقدر تغير فيها داتا يوزر تاني بدون ما تشوف الريسبونس بتاعة السيرفر.

Genetic: ودي الي بتشوف فيها الريسبونس بتاع السيرفر عادي, زي إنك تـ access data لشخص تاني وتشوفها أو تـ access ملف.

IDOR with Reference to Objects: ودي الي بتقدر تعدل أو تـ access داتا يوزر تاني عن طريق الـ ID (reference)

|=| أزاي تعمل تيست ليها |=|

- محتاج تلاقي كل البارامترز والأماكن الي بتشير او بتستدعي Objects directrly.
- بعض الأماكن(البارامتر الي بيستدعي ريكورد معين من الداتا بيس - أو صفحات - أو ملفات)
- /* وانت بتيست وبتغير قيمة بارامتر لقيمة تاني عشان تشوف هل بتقدر تـ access resources تاني او لا ركز ف انها متكونش دي عملية Business logic وإن ده الطبيعي بتاع التطبيق *\

[Testing for Insecure Direct Object References]

|=| تلاقيها فين |=|

- في تغيير الاميل والباسورد واي داتا مهمة
- في حذف صورة معينة أو جروب أو صفحة أو كومنت، في حذف أي حاجة لو لقيتها بتتحذف تبعا لـ ID أو UUID (حتى لو base64) في جرب تحط ID لصورة تاني (يُفضل لو معاك اميلين كل اميل فيه صورة عشان تعرف تتيست).
- لو ممنوع تكومنت أو تنزل صورة وهكذا, جرب تكتب كومنت في بوست تاني وبعدين تغير الـ ID للبوست الي ممنوع تكومنت عليه وتشوف هيظهر الكومنت ولا ولا وهكذا الصور.
- المعظم بيكون إن أنت ممنوع تعمل action في مكان معين (A) في بتروح تعمل الـ action في مكان تاني (B) وتغير بعدين الـ ID بتاع الـ action الي في (B) تخليه يروح لـ (A).
- أي action بتعمله بيعرضلك حاجات خاصة بيك جرب معاه الـ IDOR.
- لما يحصل وقف للاميل جرب ترجع الباسورد ووقتها هيقولك إن أميلك موقوف, ف جرب تعمل ريكوست بأميل مش موقوف وتبدل الي في الريكوست من A -> B زي password reset token أو غيره وتشوف هتقدر ترجع الأكونت ولا لا.

•

|=| طرق تخطي |=|

1- تغيير الـ GET إلى POST.

2- عن طريق الـ Parameter pollutoin بإنك تكتب البارامیتر مرتين مرة فيه الـ ID العادي ومرة فيه الـ ID الـ victim.

3- عن طريق wrapping in Array مثال

* {[Wrap ID with an array {“id”:111} --> {“id”:[111

* {{JSON wrap {“id”:111} --> {“id”:{“id”:111

{"*":"Send wildcard {"user_id

4- عن طريق الـ path traversal مثال www.example.com/edit?id=11../12

|=| ازاي تمنع ثغرة Broken Access Control |=|

- قلل إستخدام الـ CORS بشكل كبير.
- أقفل الـ directory listing في التطبيق واتأكد إن مفيش metadata أو backup موجود في الـ root path للموقع.
- اتأكد إن الـ JWT تبقى غير صالحة طالما سجلت خروج.
- أي مصادر غير الي محتاجها تكون Public ف أمنع الوصول ليها من الأول.
- إربط الـ Function والصفحات المهمة زي الـ Admin-panel بالـ user session مش بالـ user id وبكده هتمنع وصول أي user لـ resourses تبع يوزر تاني.
- الـ Session محتاجها تخليها unguessable وطويلة عشان ميحصلش brute force عليها.
- صفحات الأدمن زي الـ Admin portal غير اسمها وخليها حاجة مختلفة عشان يصعب تخمينها.
- غير أي default credentials لأي service أنت بتستخدمها.
-

مراجع:

https://medium.com/@shahjerry33/privilege-escalation-hello-admin-a53ac14fd388

https://owasp.org/Top10/A01_2021-Broken_Access_Control/
https://owasp.org/www-project-top-ten/2017/A5_2017-Broken_Access_Control
https://cheatsheetseries.owasp.org/cheatsheets/Authorization_Cheat_Sheet.html

https://owasp.org/www-project-web-security-testing-guide/latest/4-Web_Application_Security_Testing/05-Authorization_Testing/README

What is and how to prevent Broken Access Control | OWASP Top 10 2017 (A5) (مهم)

https://www.synack.com/blog/preventing-broken-access-control-the-no-1-vulnerability-in-the-owasp-top-10-2021/ (مهم برضو)

https://www.pullrequest.com/blog/how-to-catch-the-owasp-2021-number-1-broken-access-control-in-code-review-part-2/

https://owasp.org/www-project-web-security-testing-guide/latest/4-Web_Application_Security_Testing/05-Authorization_Testing/03-Testing_for_Privilege_Escalation

https://owasp.org/www-project-web-security-testing-guide/latest/4-Web_Application_Security_Testing/05-Authorization_Testing/04-Testing_for_Insecure_Direct_Object_References

https://www.bugcrowd.com/resources/webinars/broken-access-control-testing/