web hacker handbook

chapter 1 summary

#mohamed_ayad

بسم الله هنا سلسلة شرح الكتاب كامل الكتاب عبارة عن 21 شابتر هيتم تنزيل ملخص لكل شابتر باذن الله الملخص هيكون تقني عشان الموضوع ميكونش مجرد ترجمة حرفية للكلام و عشان نوصل لأقصى فائدة

يلا نبدا اول شابتر و بيكلم عن شوية اساسيات و مدخل لفهم بقية الكتاب من خلال عرض متسلسل للمعلومات لمساعدة القارئ من فهم الموضوعات بسهولة و عشان ميتعقدش و الموضوع يكون سليم باذن الله

بيقول ان الويب سايت في الاول كانت عبارة عن مجرد مخزن للمعلومات بدون اي تفاعل مع المستخدم static_docment وكل وظيفة المتصفح ان هو يجبلك المعلومات دي و يعرضهالك كل المستخدمين كانو بيتعاملو بنفس الطريقة و مكنش فيه اي authuntication علي الملفات لو attacker قدر gain access للسيرفر فهو بالتالي موصلش لاي معلومة مهمة و لكن هو يقدر يعدل الملفات اللي ع السيرفر و يعمل اللي هو عوز اما دلوقت فمعظم ال sites هي بقت app بيحتوي علي صفحات تسجيل و financial transaction و غيرها و طبعا د bring معاه الكتير من ال security threats

من ال common finctions اللي بتقدمها الweb_app s التسوق زي امازون و مواقع اجتماعية و بنوك و محركات بحث وغيرها وطبعا عندنا apps بتقدم خدمات معينة زي تطبيقات الHR اللي يقدر منها يشوف رواتب الموظفين و اداءهم في بيئة العمل طبعا كل الخدمات دي الشركة بتكون وخدها من شركة بتقدمها زي الclouding و هنا الشركة بتعتمد علي الintegrity بتاعة الشركة المقدمة للخدمة في الحفاظ علي معلوماتها secure عشان طبعا نتجنب التكلفة

ال benefits من ال apps دي :

الHTTP و د بروتوكول الاتصال الرئيسي و هو خفيف و conncetionless يعني مبيفتكرش حد

الBROWSER كل الwep_apps بتنشر واجهة المستخدم بتاعتها عشان تتفاعل مه المتصفح

معظم المواقع بتزعم ان مواقعها مؤمنه عشان مثلا عندهم SSL الواقع ان معظمها بيكون insecure و ممكن تراجع owasp top 10 طبعا الSSL هو تقنية ممتازة لحماية سرية المعلومات و صحة البيانات و بيحمينا من التجسس الا انه مش بيمنع اي من الثغرات اللي فوق دي

اهم المشاكل ان المستخدم بيقدر يsubmit اي arbitrary_input المستخدم ممكن يتفاعل مع اي بيانات بين ال server وال client زي الparameters,cookies,http_headers اي sec_control بيتم ف الserver_side بسهولة يتعمله circumvent كمان المستخدم يقدر يبعت اي تدفق من الrequests ويستخدم parameters في stages مختلفة من الconnection و بالتالي اي افتراض المبرمج حطه ممكن يviolate طبعا المستخدم ممكن يستخدم حجات كتير عشان يتواصل مع السيرفر مش شرط متصفح يعني و بالتالي يقدر يغير اسعار المنتجات او يسرق كوكيز(يستعير -_-) او يسمح بارميتر معين و بالتالي يستغل logic flow معينة او ي inject اي malicious code

اهم ال KEY_PROBLEM_FACTOR

1-عدم الوعي باهمية حماية تطبيقات الويب نظرا لان المعظم منشغل بحماية الشبكات و انظمة التشغيل

2-الcustom_development و بكدا كل appبيبقي ليه الunique defects

3-الdeceptive simlicity وهو ان اي حد لوقت يقدر يستخدم templets جاهزة و يصمم بيها موقع طبعا بدون فهم للكود ولا للsec issues د بيؤدي لbreaches كتير و طبعا اي حد يقدر يكتب كود بس مش اي حد يقدر يخليه secure

4-الrapidly evloving سرعة المجال و ان كل يوم بقي ف حاجة جديدة تخلي علي م المبرمج يخلص المشروع و يامنه من الثغرات الحالية تكون في تقنيات جديدة ظهرت بتbypass كل شغله

5-الresource and time constrains طبعا مش بيكون feasible لمعظم الشركات انها كل م تخلص جزء من الكود تروح تعمله test وطبعا الشركات عشان بتركز ع سرعة تسليم المشروع مش بتدي وقت كافي للpentester عشان يختبر الكود بشكل سليم

6-الoverextended technology الحاجة اللي كانت ف الاساس معموله عشان حاجة معينة دلوقت بستخدم لغرض تاني خالص

7-الicreament demand on functionality التركيز علي كثرة المهام اللي بيقوم بيها الapp

8-أالnew security perimeter الاول عشان تقدر تخترق اي حاجة كان لازم توصل لaccess ع الشبكة بتاعتهم اما دلوقت ف انت بمجرد م قدرت ت bypass اي restricts ف الموقع فا نت بكدا تقدر ت compromise الback-end sys

وكمان ممكن بعض الapps بمجرد الدخول عليها يقدر المهاجم يخلي الbrowser بتاع الvictim يعمل attack علي الشبكة بتاعته اللي هو عليها

the _end