Introduction

Wednesday, January 17, 2024 12:53 PM

Web apps are interactive apps in web browsers

typically use client-server architecture to run and handle interactions

Web apps vs websites

websites were static and did not change in real time

developers needed to change a page

these types of sites are Web 1.0

Web 2.0 are web apps w/ dynamic content based on user interaction

can also

- be malware

- run on any display size
- run on any platform without being optimized

Web apps vs Native OS apps

Web apps are platform independant

can run in any browser on any OS

no install needed because they are executed on the remote server

version unity - all users using web app are the same version

- dont need to push to each user
- updated in single location

- updated in single location
what during builds for each platform

Native OS apps have better operation speed and can use OS libraries and hardware

not limited to browser's capabilities

Web app distribution

open-source web apps:

- wordpress
- opencart
- Joomla

proprietary

- wix

- Wix
- Shopify
- DotNetNuke

Security risks of web apps

Vast attack surface - accessible by anyone w/ internet

many scanning tools to find vulnerabilities

attacks can lead to serious losses because servers may host other sensitive info or are linked to sensitive dbs

devs need to test apps and ensure new changed don't cause new

how changed dom

bugs

OWASP web security testing guide

- typically start on frontend HTML, JS, CS
- then look at core functions and interactions w/ server to enumerate services

Attacking web applications

not very common to find externally facing host exploitable to a known public exploit

not uncommon to find flaws that lead to code exec

often find SQLi on web appr that

or AD for auth

can usually use to get passwords

but can get emails

which are often usernames

can be used for password spraying

real-world examples:

- SQLi → get AD usernames and do password spraying against VPN or email

- File inclusion → reading source code to find hidden page or directory which exposes function

directory which exposes ...
to gain remote code execution

- unrestricted file upload → upload file for unlimited control over server

- insecure direct object referencing (IDOR) → when combo w/ broken access control can be used to access user's files or functionalities

- broken access control → abuse POST request when creating new user to make new admin user