DEVSECOPS BOOTCAMP

BUILDING RUGGED SOFTWARE

YEAR ONE / WEEK THREE / COMMON WEB APP VULNS

Agenda

- “The Core Security Problem with Web Apps”
- List the OWASP Top 10 Web Application vulnerabilities
- Detailed discussion of OWASP Top 3 web app vulns
- My favorite web app vulnerability …
- Resources for you

“The Core Security Problem” (with web apps)

“Because the client is outside of the application's control, users can submit arbitrary input!” -Web Application Hacker’s Handbook (WAHH)

- Users can modify all data transmitted between the client and the server
- Users can send HTTP requests in any order they want, and can submit parameters in a different stage than the application expects
- Users can tool capable of communicating via http to interact with the web application … they are not restricted to using only a web browser!

Open Web Application Security Project (OWASP) Top 10 (2013)

1. **Injection**
2. Broken Authentication and Session Management
3. **Cross-Site Scripting (XSS)**
4. Insecure Direct Object References
5. Security Misconfiguration
6. **Sensitive Data Exposure**
7. Missing Function Level Access Control
8. Cross-Site Request Forgery (CSRF)
9. Using Components with Known Vulnerabilities
10. **Unvalidated Redirects and Forwards**

OWASP #1 - Injection

- Injection flaws occur when an application sends untrusted data to an interpreter
- Attacker sends simple text-based attacks that exploit the syntax of the targeted interpreter
- Almost any source of data can be an injection vector, including internal sources
- Injection flaws are very prevalent, particularly in legacy code
- **They are often found in SQL**, LDAP, Xpath, or NoSQL queries; OS commands; XML parsers, SMTP Headers, program arguments, etc...
- NOTE: It's not just text-based … binary blobs can also be an injection vector (serialized data, malicious archives, image files, etc...)

OWASP #2 – Broken Auth & Session Mgmt

- Developers frequently build custom authentication and session management schemes, but building these correctly is hard!
- Attackers use leaks or flaws in the authentication or session management functions to impersonate users
- NOTES:
- Don't forget about anonymous users, users who haven’t completed registration, or “expired users” … does your auth / session management handle those?
- Attacking a site that uses ”mutual authentication”? I've seen many a DoD site that took my valid SSL Identity Cert (stolen from a server exploitation during same assessment), failed to map me to a human, and …

OWASP #3 – Cross-Site Scripting (XSS)

- XSS flaws occur when an application includes user supplied data in a page sent to the browser without properly validating or escaping that content
- Attackers can execute scripts in a victim’s browser to hijack user sessions, deface web sites, insert hostile content, redirect users, hijack the user’s browser
- Attacker sends text-based attack scripts that exploit the interpreter in the browser. Almost any source of data can be an attack vector, including internal sources such as data from the database
- Reflected (aka First Order) XSS : Victims must be lured (phish, watering hole, etc …)
- Stored (aka Second Order) XSS : Persistent attack! User is guaranteed to be using site
- DOM-based XSS :All client-side! ... the HTML response does not contain the attacker's script, instead .. it is written into the page after rendered, and then executed!

My fav web app vuln – External XML Entity (XXE)

- An XML External Entity attack targets applications that parse XML input
- This attack occurs when XML input containing a reference to an external entity is processed by a weakly configured XML parser
- This attack may lead to the disclosure of confidential data, denial of service, file system enumeration, port scanning from the perspective of the machine where the parser is located, and other system impacts
- Attacks can include disclosing local files, which may contain sensitive data such as passwords or private user data
- Since the attack occurs relative to the application processing the XML document, an attacker may use this trusted application to pivot to other internal systems

Resources

- If you are interested in learning more about web application vulnerabilities, the must-read (and read again x10) book is the "Web Application Hacker's Handbook" (WAHH) by Stuttard and Pinto
- Their web site is http://mdsec.net
- They teach a great 2 day class at BH USA every year … it is a great way to take what you have learned from the book, and apply it (with Burp)
- Join your local OWASP Chapter and attend meetings/briefings/CTF's
- OWASP web site is: https://www.owasp.org
- Setup an alternate profile in your browser of choice using Burp as local proxy. Anytime you get curious, copy and paste URL into alternate browser profile ... and start reverse engineering!