Authorisation and Access Control

Daniel Bosk[1]

Avdelningen för informations- och kommunikationssytem (IKS),
Mittuniversitetet, Sundsvall.

access.tex 1674 2014-03-19 14:39:35Z danbos

Översikt

1. Secure Password Storage
   - Why salts?
2. Authorisation
   - Principle of least privilege
   - Access Control Lists (ACL)
   - Centralised Authorisation Routines
   - Static Content
   - Don't trust the user
3. Software with vulnerabilities
   - What to do?

Översikt

Mittuniversitetet
MID SWEDEN UNIVERSITY

Översikt

Why salts?

Översikt

Översikt

Principle of least privilege

- The main point with the "principle of least privilege" is to deny access to things you normally don't need.
- Why should your webserver have read-permission to the system password database?
- It will never attempt to read it anyway, so why bother?
- Because you never know what might happen to your webserver communicating with the outside world – it can be forced to anything, then the operating system must prevent it.

Principle of least privilege

- Why should we have “drop table”-permissions for reading the password database?
- A slip-up in any function accessing the database can then allow the attacker to do anything.
- Otherwise, the attacker must find a vulnerabiblity to exploit in the only functions with write permissions to that particular table.

Principle of least privilege

Keys to the kingdom again

- But what happens when all parts run as the same no-privilege user?
- Then they have rights to each other, so we're back at it.
- Separate different applications with different system users.
- This problem cannot arise inside the database.
- But if the webserver and database server runs as the same user, the webserver can just kill the database server and read the files directly from disc instead of using the API.

Översikt

Access Control Lists (ACL)

- An ACL specifies which users should have which permissions.
- It usually has a list of permissions; e.g. read, write. Sometimes also append, insert etc.
- Each user or role is entered with a certain set of permissions.
- There are ACLs in both operating system (OS), for running processes and file systems, but also inside databases.

Access Control Lists (ACL)

- Apply the principle of least privilege.
- This is usually not the case, most files are readable by everyone when created.
- The same applies to databases, unsecure by default.

Översikt

Centralised Authorisation Routines

- Use centralised functions in your application to manage authorisation, i.e. to enforce ACLs.
- Centralisation in the application means less code, less code means less potential for bugs.
- Don't copy-paste authorisation code to all functions, use one centralised function which handles the ACL.

Centralised Authorisation Routines

- The ACLs can be represented as an authorisation matrix.
- I.e. a huge table listing all users, actions and resources along with whether the user is allowed.
- This can be implemented in a database.
- All parts of the webapplication must conform to this, use the centralised functions to enforce it.

Centralised Authorisation Routines

- Make sure to check both actions and resources.
- Is this user allowed to edit personal settings? Yes, he is, he's logged in.
- Whose personal settings did the logged-in user edit?

Centralised Authorisation Routines

- When implementing the authorisation checks, watch out for "positive authorisation"!
- I.e. don't assume things will succeed, always write for default deny.
- If your code is aborted, in any line, it should fail the authorisation.

Översikt

Static Content

- Remember to protect files in the webserver's file system, e.g. PDFs and images if they are not publicly available.
- Either set some ACLs for them to disallow fetching them using direct URLs.
- Another possibility is to generate the content on the fly.

Översikt

Don't trust the user

- Don't trust anything from the client-side.
- Store all critical data on server-side, all access control related things should be on the server – and check every time!

Översikt

Mittuniversitetet
MID SWEDEN UNIVERSITY

Översikt

What to do?

- Keep track of version numbers for all libraries and frameworks used in the application.
- If a bug is found in any of them, see if they apply to you and update the library. Maybe you don't use that particular functionality.

Referenser