Ethical Student Hackers

Denial of Service

The Legal Bit

- The skills taught in these sessions allow identification and exploitation of security vulnerabilities in systems. We strive to give you a place to practice legally, and can point you to other places to practice. These skills should not be used on systems where you do not have explicit permission from the owner of the system. It is VERY easy to end up in breach of relevant laws, and we can accept no responsibility for anything you do with the skills learnt here.

- If we have reason to believe that you are utilising these skills against systems where you are not authorised you will be banned from our events, and if necessary the relevant authorities will be alerted.

- Remember, if you have any doubts as to if something is legal or authorised, just don't do it until you are able to confirm you are allowed to.

- Relevant UK Law: https://www.legislation.gov.uk/ukpga/1990/18/contents

Code of Conduct

- Before proceeding past this point you must read and agree to our Code of Conduct - this is a requirement from the University for us to operate as a society.
- If you have any doubts or need anything clarified, please ask a member of the committee.
- Breaching the Code of Conduct = immediate ejection and further consequences.
- Code of Conduct can be found at **shefesh.com/conduct**

Denial of Service

General Idea:

Overwhelm the server with requests. So that the server slows down, or crashes.

Targets the availability of a system.

User Specified Object Creation

Vulnerable Area:

If a user is able to submit how many items, or objects to create.

+

There is no upper limit

Server will try to create all of the objects, encountering memory issues, leading to crashes and slower performance.

This also works if the user specifies a upper limit for a loop counter

Object Release Failure

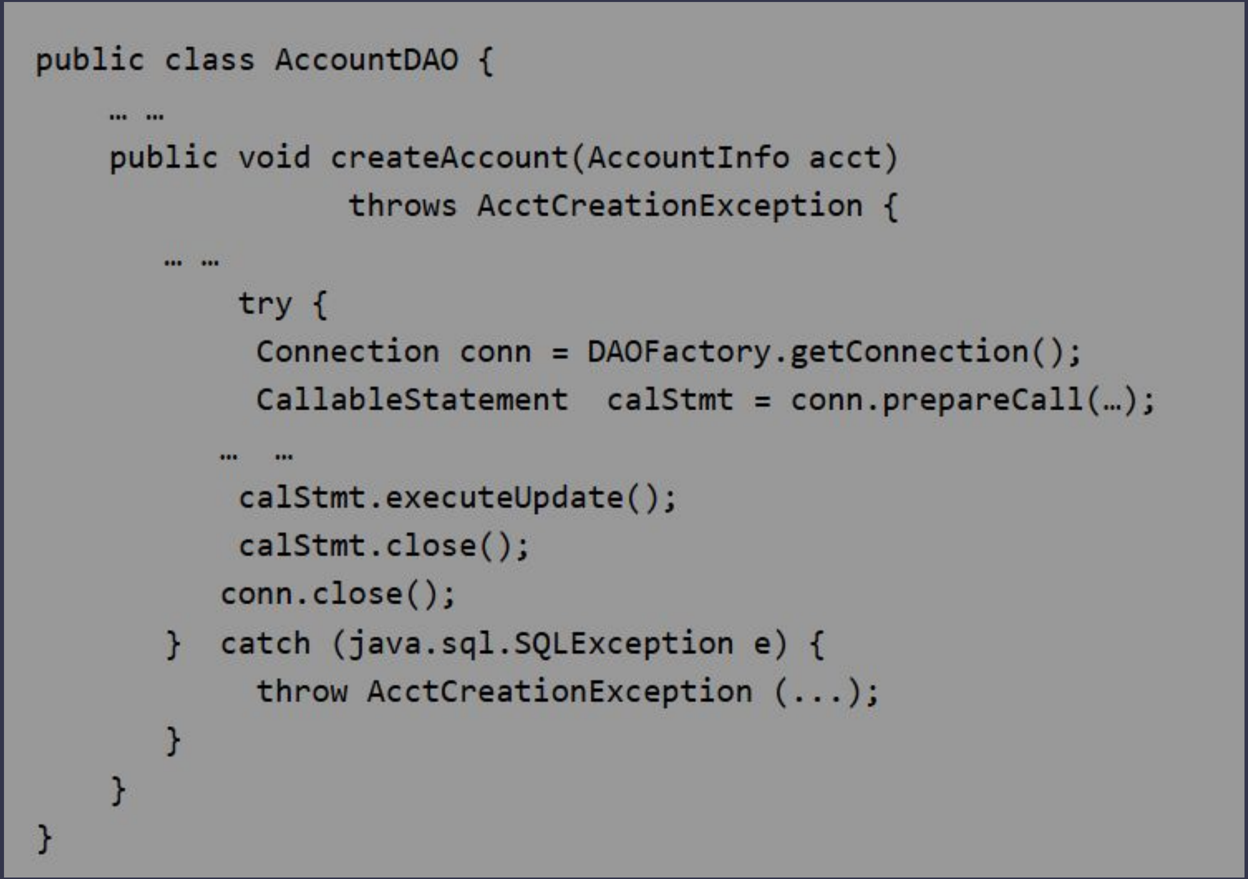
When a user accesses a file, or database field, it is locked to that user, until the user finishes using it, and it is released to be used by other users.

If the user is able to prevent the release of the file or database field, then it is unavailable for users.

How?
Some errors/exceptions may not explicitly release the object
Memory leaking in c/c++, can commandeer block of memory - effective locking them, or crash program

Practically?
Try, catch, finally

Example



```
public class AccountDAO {
    … …
    public void createAccount(AccountInfo acct)
                 throws AcctCreationException {
       … …
           try {
            Connection conn = DAOFactory.getConnection();
            CallableStatement  calStmt = conn.prepareCall(…);
          …   …
           calStmt.executeUpdate();
           calStmt.close();
          conn.close();
       }  catch (java.sql.SQLException e) {
            throw AcctCreationException (...);
       }
    }
}
```

Network DoS

We’ve looked at breaking the application.

What about the network?

If you send as many requests as possible, it will consume the network bandwidth, preventing regular traffic (other users) from accessing the server.

Funky stuff I couldn’t categorise

Slow http requests:
You can send fragmented http requests, and the server will allocate resources towards receiving it. By slowly sending the fragments, it is tying up server resources.

Large assumption of DoS:
That there is a single point of failure

Network Switches:
Switches have a MAC table that stores what Ip addresses match to what MAC addresses on the network. If you send requests from lots of different MAC addresses - it maxes out the table, causing the usual network traffic unable to go through the router

Zip bombs:
You can send extremely large files (terabytes upwards) by compressing it in a zip file, then sending it to a server. If it auto unzips the file, it will hit a memory limit, as no server has hundreds of terabytes of RAM (and defo not in this economy!)

Practical Time!!!

http://35.179.149.69/

https://overthewire.org/wargames/

Feedback

Please leave your feedback :) We want to know what we can do to improve.

Please leave constructive and honest feedback only.

https://forms.gle/VTYd74K5BHqbC7F68

Inclusions Concerns

If there's anything preventing you from enjoying our sessions, please let our Inclusions Officer know. You can contact them by email or fill in the form below:

jgledhill2@sheffield.ac.uk

https://forms.gle/Qct6Wyfesv8dWmej7

Upcoming Sessions

What’s up next?

www.shefesh.com/sessions

Next week: Exploiting Randomness

The week after that: Advanced password cracking

Any Questions?

SHEFFIELD ETHICAL STUDENT HACKERS

www.shefesh.com

Thanks for coming!