Authentication flaws and responsible disclosure

Jonathan Haddock CITP MBCS CISSP MSc
codeHarbour
14th September 2023

https://blog.jonsdocs.org.uk
@joncojonathan

I've worked across a number of roles, starting in service desk / desktop support, and each has benefited from experience gained in the others. I've had a passion for security for the last 15 years or so, something that started from my time running IT in a secondary school. Students will certainly keep you on your toes! I'm now a *Senior Information Security Officer* working for a software development company in the private sector.

If you want to get into IT security, I recommend you specialise in an area you find interesting but make sure you keep your hand in other areas too so you have a broad skill set and knowledge. This will allow you to have informed discussions with different teams, in a way they (and you) can understand.

I also blog about cyber security, development, IT, and occasionally local history at https://blog.jonsdocs.org.uk.

What is authentication?

- Checking that the identity you're claiming is yours
- Could be as simple as asking someone their name
  - Could be as complex as asking for independently validated identity documents
- In computing, often involves checking a username and password pair match an identity in a system

- Authentication is about *identities*
- I am Jonathan – you could authenticate that by checking with my Mum, or someone that you trust, that also trusts me
- When we *authenticate* something we check it's legitimate – in this case we check that the person saying "I'm Jonathan" is actually Jonathan
- In computing we often say someone can claim an identity (username or account) if they know the password associated with that account

Authentication flaws

- Authentication bypass
  - "I can get around the check"
- Forgetting to check
  - "Woohoo, I'm in!"
- Not checking that the identity is valid for the system
  - Maybe by not tying the identity database to the system that's being protected - more on that in a bit...
- Other flaws will exist, but aren't the subject of this talk

- An *authentication flaw* is a scenario where someone can operate under an identity that is not theirs
- Using a different identity might give extra privileges (e.g. allow the user to use an admin account), or access to data they wouldn't usually have

What is authorisation?

- Authorisation involves checking you're allowed to do a specific thing
  - I want to create
  - I want to delete
- Could be checking that you're allowed a free drink at codeHarbour 😊 💪
  - If your name is on the list then you're *authorised* to have the drink

Note that authorisation requires authentication to have taken place – you can't make sure someone is allowed to do something if you don't know who they are.

Authorisation flaws

While not the focus of this talk, some authorisation flaws include:

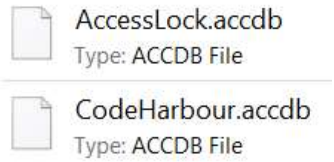
- Not checking at the point of use (design flaw)
  - "A user has got to this page somehow, therefore they *must* be allowed to use it"
- Time of check vs time of use differences
  - "Jonathan logged in as an admin. While logged in as an admin his permissions got revoked. Jonathan still performed an administrative action as the system did not check again"

Scenario

- An application has been installed that controls fobs for doors
- To make changes you have to login with a valid username and password
- We have three doors configured
  - Front Door
  - Alex Lab
  - Restaurant

Demo: Authentication flaw

- Browse to the door control software database folder
  - `C:\Videx\PROA MS\Database`
- We have two files
  - `AccessLock.accdb`
  - `CodeHarbour.accdb`



Let's see the demo!

Demo: What happened?

- Open the vulnerable software - PROA MS Videx v3.8.0.0
- Password is needed to access the CodeHarbour site
  - Or is it?
- Delete `AccessLock.accdb`
- Login with the default username and password
  - admin / admin

Demo: What happened?

- CodeHarbour site opens – configuration still present!
- A new `AccessLock.accdb` gets made with the default credentials

What does that mean?

- We determined the `AccessLock.accdb` file was controlling access to the application only
- Accessing a site database file didn't have protection
  - The site database didn't have the password in it
- You could access any version 3.8.0.0 (possibly lower) database this way
  - Don't even need to bring a file with a known username and password

What is responsible disclosure?

- Talking to the vendor before you publish your findings
- Keeping your findings secret until the patch is released
  - Or until a reasonable time has passed
  - 90 days is often a suggested time frame, but the severity of an issue might make a researcher try to reduce that
- Working with the vendor to confirm a fix

What happened next?

- I contacted the installation company and let them know of the flaw

Thank you for the feedback. You may want to contact the developers XPR about this directly, as they are the ones that make the MiAccess product and the PROA MS software. You can find their contact details at https://www.xprgroup.com/contact/

- So I contacted the developers

Next contact...

- After about three days...

Please find a sendspace link below to download the new version of PROA MS (4.0.0). This alters the database structure so that the site password is stored within the database for that site and not the separate AccessLock file.

...

It will no longer be possible to reset the password by deleting the AccessLock file, and this file can be deleted without any effect as it is no longer used by the software.

Demo: Let's check

- I've already upgraded the software
  - No-one wants to watch progress bars!
- After the upgrade the password was returned to the default
  - I set a new password
- Can we delete the `AccessLock.accdb` file?
- Is the new password still required to get into the CodeHarbour database?

What have we covered?

- Authentication is checking *identity*
- Authorisation is checking an *identity* is allowed to do a particular thing
- Authentication flaws can allow you to bypass authentication completely
- Responsible disclosure allows a software vendor time to fix a problem
- Good vendors work with security professionals and researchers to ensure fixes work

Questions?