CAPI/CATI:

› Trained interviewers, effects on

- Screen layout, key options, QxQ help, error messages

› Not accessible to other users

- Better security

Less control in web interviews

> Ideally:

- Intended respondent logs in via Management System
- Starts interview
- Answers all questions and
- Finishes/leaves correctly

> Less ideal:

- Restarts
  - intended or
  - unintended (X-out or time out)
- Unintended respondents

Web security code

› There is a need for specific code to deal with security on the web

Mostly we try to concentrate that in the management system parts, e.g. with authentication/login

However, as the Blaise interview has its own URL there is a need for specific security-related Blaise code as well: passing in all security-related parameters in the URL is not secure

One solution: Management System inserts a value in Blaise data just before start of interview, then use that in a 'valid interview' check in Blaise code

That way just having the URL is not enough

If we have to include web security checks in the Blaise code we try to:

❯ Keep it as separate as possible from 'normal' code

❯ Make it as flexible as possible

- Accessible when needed
- Reusable between projects

Use of Resource Database and management block (1)

Approach in paper:

› Use the Resource Database and a separate management block in the Blaise code

› Map fields in management block to field references in the Resource Database

Use of Resource Database and management block (2)

When starting an interview from the management system:

› The management system sets a field value in the management block, using the API

› This value gets copied to the related field reference (Resource Database)

› The field reference defined in the Resource Database is used in expressions that may

- Set other variables

- Quit interview

- Set return URL

Advantage of the approach

Using the mapping between Resource Database and management block allows us to use the same expression code (for security) even when using a different management block

- By adapting the mapping between field references and management block variables

It adds a layer of abstraction, adding to the re-usability

Conclusions

Security for web interviews is paramount to keep data and the server infrastructure safe.

We described one possible method to increase security, by:

- Not including all necessary information in the interview URL

- Adding 'security checks' in the Blaise code

- Using the resource database to improve re-usability

Questions?