Logging

What is logging?

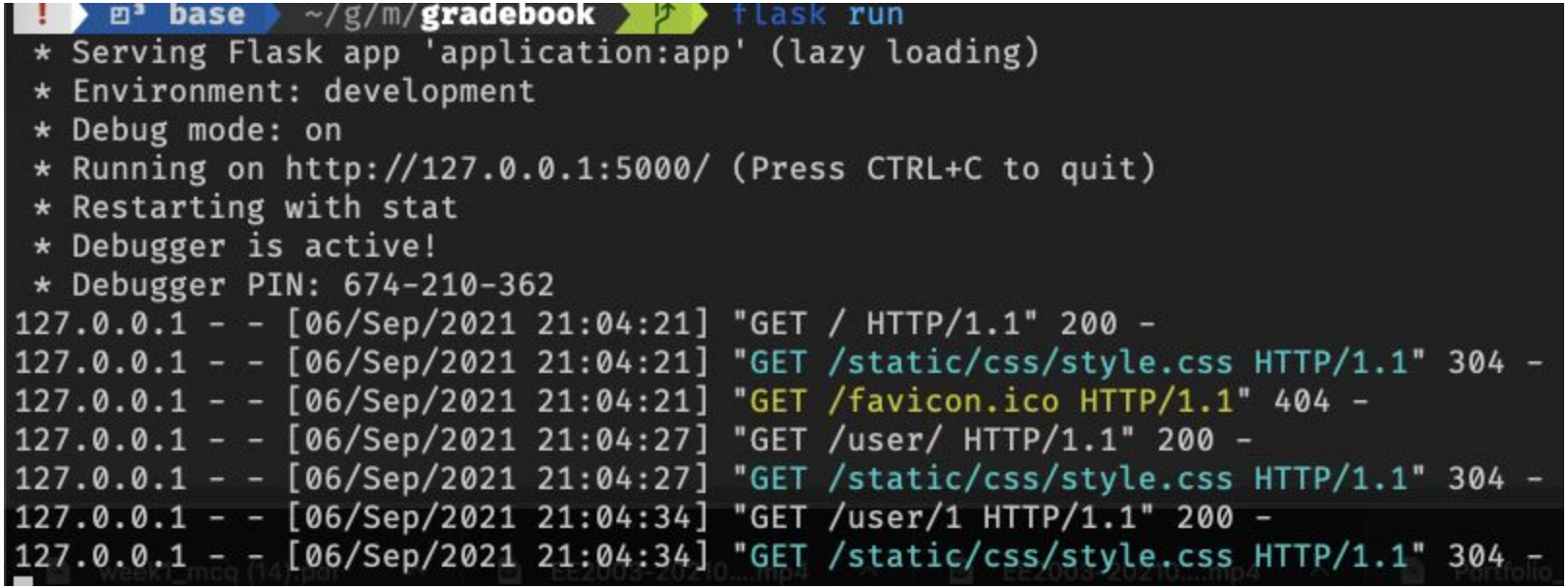
- Record all accesses to app
- Why?
  - Record bugs
  - Number of visits, usage patterns
  - Most popular links
  - Site optimization
  - Security checks
- How?
  - Build into app - output to log file
  - Direct output to analysis pipeline

Server logging

- Built in to Apache, Nginx, …
- Just accesses and URL accessed
- Can indicate possible security attacks:
  - Large number of requests in short duration
  - Requests with “malformed” URLs
  - Repeated requests to unusual endpoints

Application level logging

- Python logging framework
  - Output to file, other “stream” handlers
- Details of application access
  - Which controllers
  - What data models
  - Possible security issues
- All server errors


! base ~/g/m/gradebook flask run
* Serving Flask app 'application:app' (lazy loading)
* Environment: development
* Debug mode: on
* Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
* Restarting with stat
* Debugger is active!
* Debugger PIN: 674-210-362
127.0.0.1 - - [06/Sep/2021 21:04:21] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [06/Sep/2021 21:04:21] "GET /static/css/style.css HTTP/1.1" 304 -
127.0.0.1 - - [06/Sep/2021 21:04:21] "GET /favicon.ico HTTP/1.1" 404 -
127.0.0.1 - - [06/Sep/2021 21:04:27] "GET /user/ HTTP/1.1" 200 -
127.0.0.1 - - [06/Sep/2021 21:04:27] "GET /static/css/style.css HTTP/1.1" 304 -
127.0.0.1 - - [06/Sep/2021 21:04:34] "GET /user/1 HTTP/1.1" 200 -
127.0.0.1 - - [06/Sep/2021 21:04:34] "GET /static/css/style.css HTTP/1.1" 304 -

Log rotation

- High volume logs - mostly written, less analysis
- Cannot store indefinitely
  - Delete old entries
- Rotation:
  - Keep last N files
  - Delete oldest file
  - Rename log.i to log.i+1
  - Fixed space used on server

Logs on custom app engines

- Google app engine
  - Custom logs
  - Custom reports
- Automatic security analysis

Time series analysis

- Logs are usually associated with timestamps
- Time series analysis:
  - How many events per unit time
  - Time of specific incident(s)
  - Detect patterns (periodic spikes, sudden increase in load)
- Time-series databases
  - RRDTool, InfluxDB, Prometheus, …
  - Analysis and visualization engines

Summary

Security is key to successful applications!

- Requires good understanding of principles
  - Crypto
  - SQL, OS vulnerabilities, …
- Good frameworks to be preferred
- Analyze, Identify, Fix