SECURE CODING

TEAM 12 - PHASE 5

MEMBERS

Alexander Lill
Lorenzo Donini
Florian Mauracher
Mahmoud Naser

USE-CASES AND VULNERABILITIES

USE-CASES PHASE 1

Usecase	Status
Customer / Employee registration (including sending e-mail with TANs)	WORKING
Customer / Employee login	WORKING
Customer / Employee logout	WORKING
Customer / Employee views bank account details of customer	WORKING
Customer / Employee views transaction history of customer	WORKING
Customer money transfer via HTML form (using TAN)	WORKING
Customer money transfer via uploading transaction batch file (using TAN)	WORKING
Employee approves transfers larger than 10.000 EUR	WORKING
Employee approves registration of customer or of other employee	WORKING
Customer / Employee downloads transaction history of customer as PDF	WORKING
Transaction verification page	WORKING
Search feature for user accounts	WORKING
Employee rejects transfers larger than 10.000 EUR	WORKING
Employee rejects registration of customer or of other employee	WORKING

USE-CASES PHASE 3

Usecase	Status
Password recovery	WORKING
Encrypted TAN generation and delivery via e-mail to customer	WORKING
Download of SCS after registration	WORKING
Transfer using SCS (Single transaction)	WORKING
Transfer using SCS (Batch transaction)	WORKING

VULNERABILITIES PHASE 1

Vulnerability	Status
Directory listing and file extension handling	FIXED
HTTPS	FIXED
User registration process	PENDING
Weak lockout mechanism	FIXED
Bypassing authentication schema	FIXED
Weak passwords	FIXED
Bypassing authorization schema	FIXED
Unsecure cookies	FIXED
Session fixation	FIXED
Exposed session variables	FIXED

Vulnerability	Status
XSRF	FIXED
Session timeout	FIXED
XSS	FIXED
SQL injection	FIXED
Command injection	FIXED
Integer overflow	FIXED
Business logic data validation	FIXED
Process timing	FIXED
Application misuse	FIXED
Upload of malicious files	FIXED
Clickjacking	FIXED

VULNERABILITIES PHASE 3

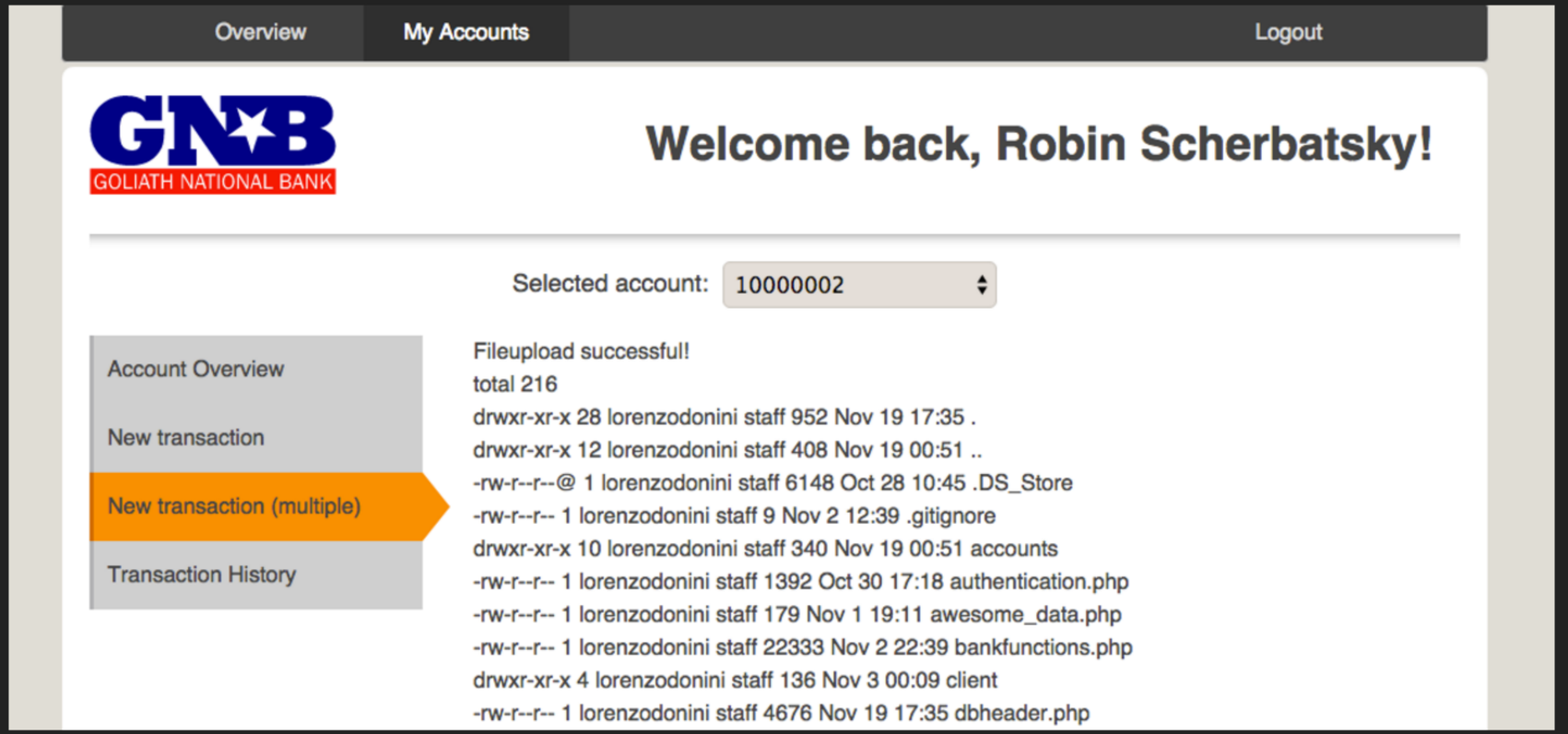
Vulnerability	Status
File extension handling	FIXED
User registration process	FIXED
Error codes	FIXED
Weak SSL/TLS cipher	FIXED
Business logic data validation	FIXED
Application mis-use	FIXED


TOP SECURITY
FEATURES

Feature	Secure against
Proper server configuration	Information leakage
HTTPS	Man-In-The-Middle attacks
Strong password	Brute-force attacks
Secure password	Database dumps
CSRF Tokens	One-click-attacks
Password reset with two-factor authentication	Account theft
Lockout mechanism	Brute-force attacks
CAPTCHAs	Fake user registration
Resource mapper	File inclusion attacks
Balance calculated in database	Circumvention of work flows
Password protected TANs	MITM attempt to intercept emails
Secure cookies	Session hijacking
Input sanitization	XSS, Command injection
Prepared SQL statements	SQL injection

LESSONS LEARNED

COMMAND INJECTION IS EVIL!



PROJECT MANAGEMENT

- Applications like Trello make life easier
- Agile programming FTW

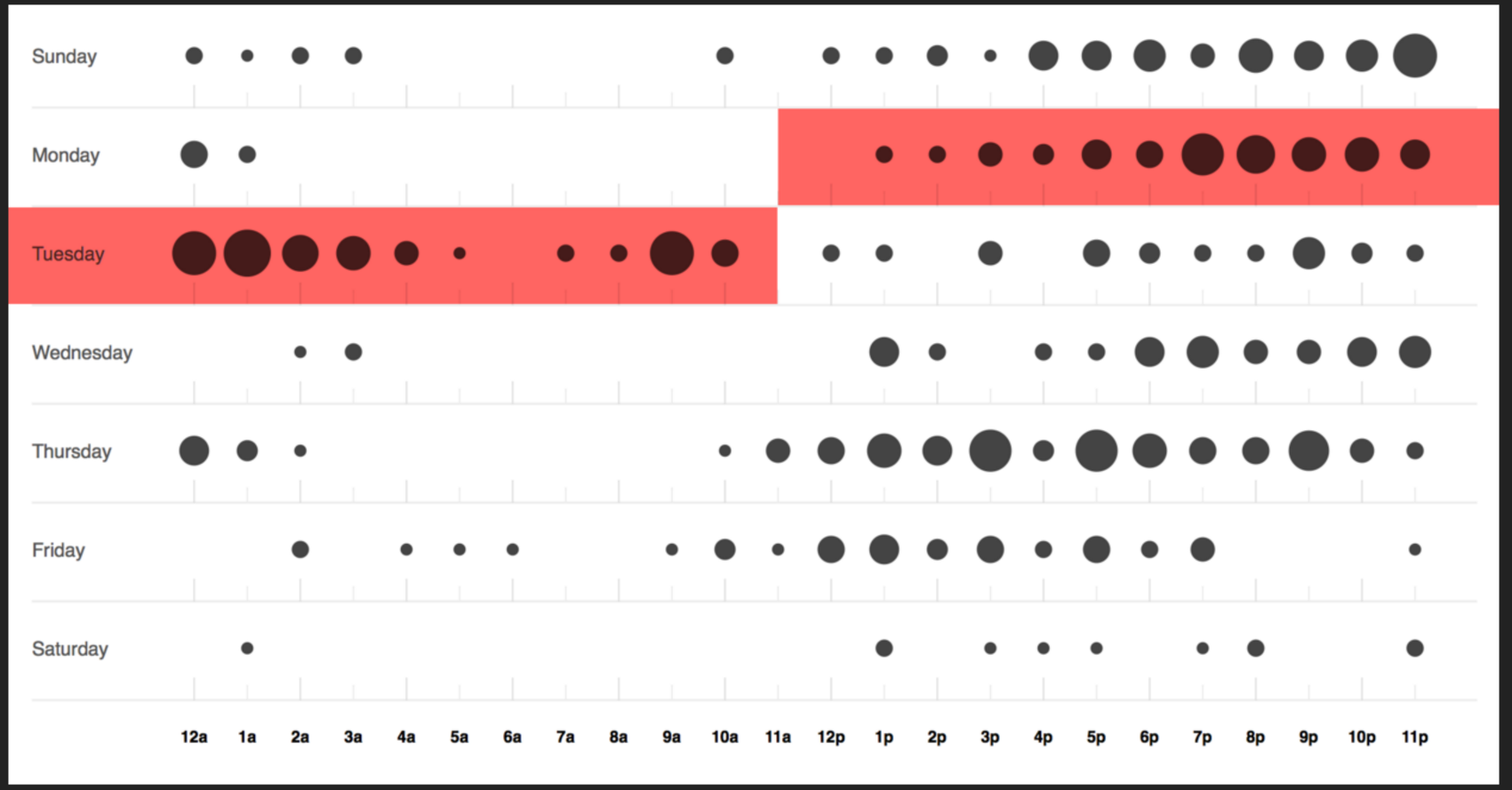
APPLY SOFTWARE DESIGN PRINCIPLES FROM THE BEGINNING!

TESTING IN TARGET ENVIRONMENT IS IMPORTANT

&

AUTOMATED TESTING IS AWESOME

TIME MANAGEMENT?



TEAMWORK

THIS PROJECT IN NUMBERS...

Party pizzas & takeouts	360€
Total amount of commits	460+
Average hours of work per phase	120
Exchanged group messages	3000+
Trello cards	70+
Lame jokes	countless
Days since the beginning of the project	102

DO YOU THINK YOUR APPLICATION IS SECURE?

DO YOU THINK YOUR APPLICATION IS SECURE?

WOULD YOU PUT YOUR OWN MONEY IN THE BANK BUILT BY YOUR TEAM?

CHALLENGE ACCEPTED.

THANKS FOR YOUR ATTENTION!