SECURE CODING

TEAM 12 - PHASE 3

MEMBERS

Alexander Lill
Lorenzo Donini
Florian Mauracher
Mahmoud Naser

USE-CASES

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

THE HTTPS MIGRATION

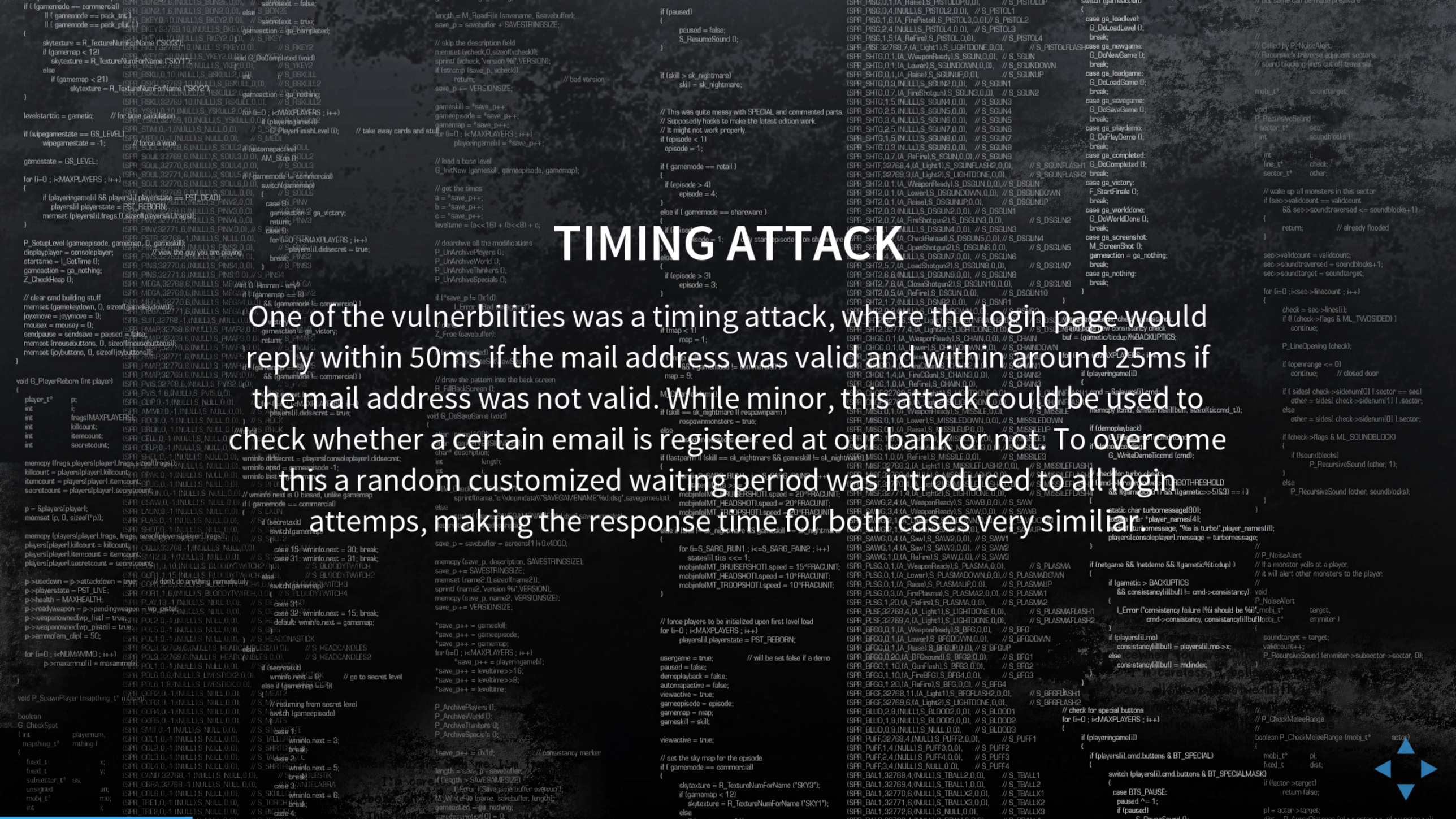
Moving the website from HTTP to HTTPS solved a large amount of issues that was related to clear text data.

MOVING THE WEBSERVER DOCUMENTROOT

One of the major vulnerabilities was the fact that a few files were accessible through the webserver DocumentRoot. These files included sensitive infomration regarding the configuration of the system. Our solution was to change the webserver DocumentRoot to the folder that only contains the PHP code and no sensitive information.

TIMING ATTACK

One of the vulnerbilities was a timing attack, where the login page would reply within 50ms if the mail address was valid and within around 25ms if the mail address was not valid. While minor, this attack could be used to check whether a certain email is registered at our bank or not. To overcome this a random customized waiting period was introduced to all login attemps, making the response time for both cases very similar.

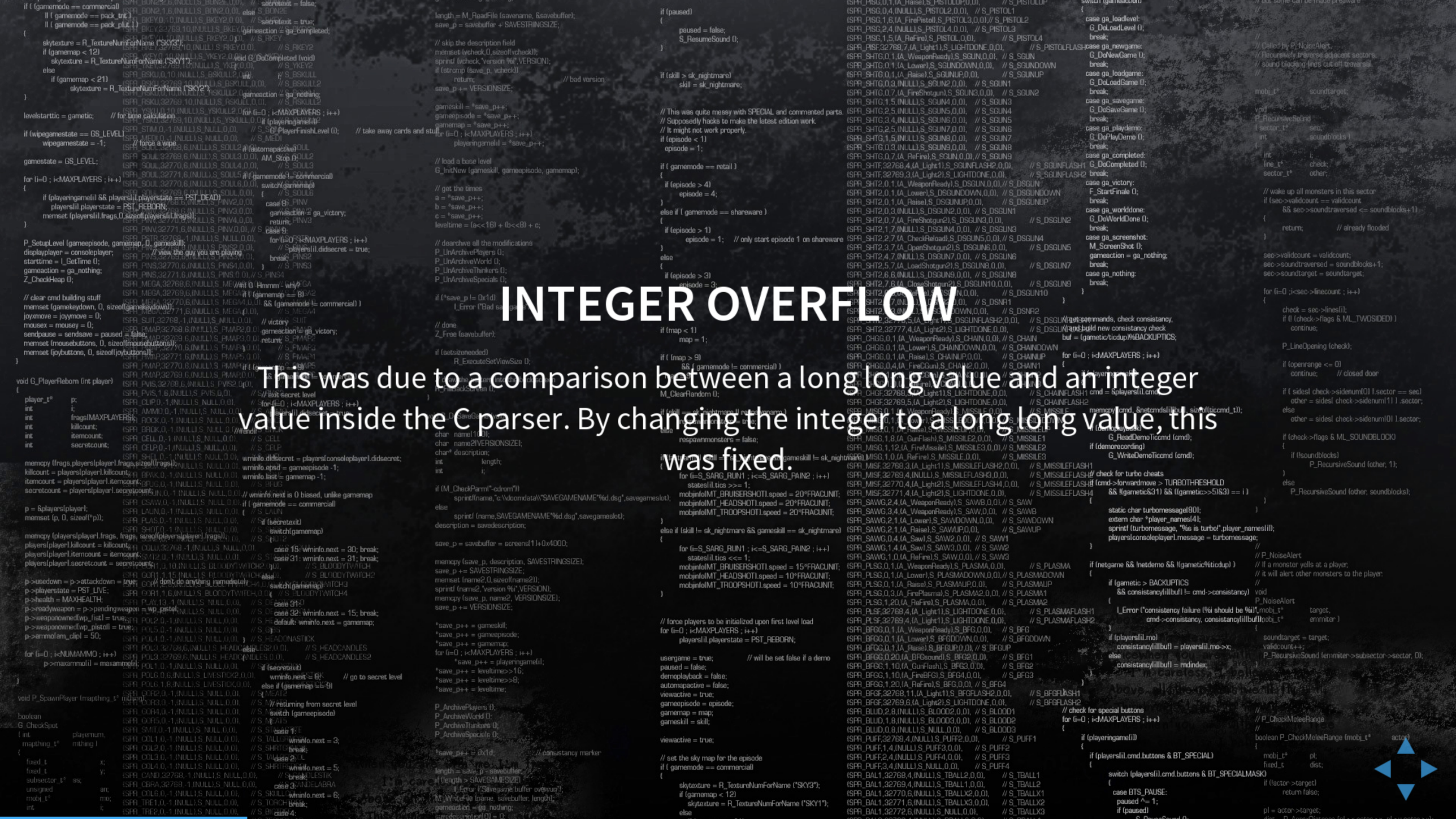


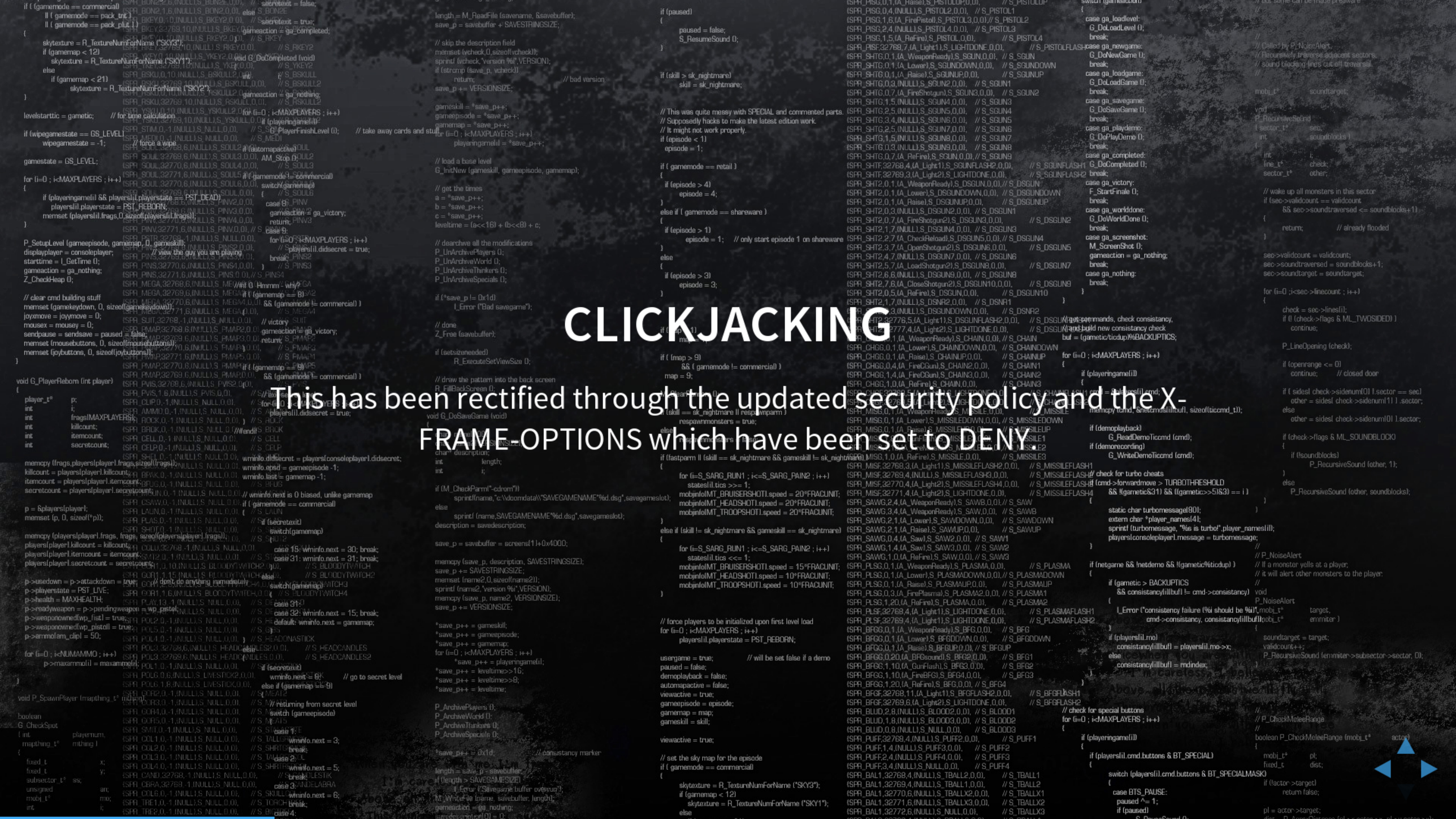
UNSANITIZED INPUT

Creating a unified sanitizing function was the solution to combating unsaniziaton.

INTEGER OVERFLOW

This was due to a comparison between a long long value and an integer value inside the C parser. By changing the integer to a long long value, this was fixed.




CLICKJACKING
This has been rectified through the updated security policy and the X-FRAME-OPTIONS which have been set to DENY.

THANKS FOR YOUR ATTENTION!

FIXES

OTG-CONFIG-003

FILE EXTENSIONS HANDLING FOR SENSITIVE INFORMATION

Changed files	
N/A	N/A

This vulnerbility was due to an error in setting up the github directory, this was remediated by editing the DocumentRoot value in the apache settings

OTG-CONFIG-007

HTTP STRICT TRANSPORT SECURITY

Changed files	
N/A	N/A

This check will no longer be needed as the HTTP option is no longer an option on the website, as the website is only available in HTTPS.

OTG-AUTHN-001

CREDENTIALS TRANSPORTED OVER AN ENCRYPTED CHANNEL

Changed files	
N/A	N/A

This has been fixed by the implemenation of HTTPS on the website, and credentials are no longer sent in clear text.

OTG-AUTHN-003

WEAK LOCKOUT MECHANISM

Changed files	
/gnb/project/employee/manage_blocked.php	New file
/gnb/project/js/employee.js	+lines 69-84
/gnb/project/models/user.php	+lines 141-154
/gnb/project/db.php	+lines 287-366

A lockout mechanism was implemented: after failing a login 5 times in a row (status is kept in database), a user account automatically gets blocked. A new section in the employee area was added in order to allow employees to unblock user accounts which have been blocked.

OTG-AUTHN-004

BYPASSING AUTHENTICATION SCHEMA

Changed files	
/gnb/project/login.php	+lines 19
/gnb/project/logout.php	+lines 5

The issue here was the constant SESSION_ID. This has now been fixed, and a new SESSION ID is generated every time.

OTG-AUTHN-005

REMEMBER PASSWORD FUNCTIONALITY

Changed files	
N/A	N/A

This was also remediated through the implementation of HTTPS, see OTG-AUTHN-001.

OTG-AUTHN-007

WEAK PASSWORD POLICY

Changed files	
/gnb/project/js/registration.js	+lines 21-45
/gnb/project/registration/registration_request.php	+lines 3-15

In order to prevent weak passwords, the user is now forced to enter a password between 8 and 20 characters long, containing at least 1 number and 1 letter. Any special character is allowed. The password strength check is performed both on client side and server side during the registration process.

OTG-AUTHZ-001

DIRECTORY TRAVERSAL/FILE INCLUDE

Changed files	
N/A	N/A

This was fixed by the same fix implemented in OTG-CONFIG-003.

OTG-AUTHZ-002

BYPASSING AUTHORIZATION SCHEMA

Changed files

/gnb/project/employee: client_details.php, client_transaction_details.php, employee_area.php, employee_overview.php, manage_clients.php, manage_registration.php, manage_transfer.php, transfer_details.php, search_client.php	+lines 5-21
/gnb/project/accounts: account_overview.php, download_transactions.php, my_accounts.php, new_transaction.php, new_transaction_multiple.php, transaction_history.php, transaction_view.php, verify_transaction.php	+lines 5-16
/gnb/project/client/client_overview.php	+lines 6-21
/gnb/project/registration: registration_default.php, registration_pin.php, registration_request.php	+lines 5-14

Pages, sections and frames are no longer directly accessible, since checks are performed at the beginning of each php file: only users with the correct permissions are allowed to access specific pages.

OTG-AUTHZ-004

INSECURE DIRECT OBJECT REFERENCES

Changed files	
N/A	N/A

Please refer to OTG-CRYPST-003 for this fix, since the vulnerability was based on the usage of an unencrypted communication channel.

OTG-SESS-001

BYPASSING SESSION MANAGEMENT SCHEMA

Changed files	
N/A	N/A

This was also remediated through the implementation of HTTPS, see OTG-AUTHN-001. SESSION ID is no longer sent in clear text.

OTG-SESS-002

COOKIES ATTRIBUTES

Changed files	
N/A	N/A

This was also remediated through the implementation of HTTPS, see OTG-AUTHN-001. SESSION ID is no longer sent in clear text.

OTG-SESS-003

SESSION FIXATION

Changed files	
/gnb/project/login.php	+lines 19
/gnb/project/logout.php	+lines 5

This was fixed by the same fix in OTG-AUTHN-004.
SESSION ID is now renewed each time.

OTG-SESS-004

EXPOSED SESSION VARIABLES

Changed files	
N/A	N/A

This was also remediated through the implementation of HTTPS, see OTG-AUTHN-001. SESSION ID is no longer sent in clear text.

OTG-SESS-005

CROSS SITE REQUEST FORGERY

Changed files	
/gnb/project/accounts/new_transaction.php	+lines 17-24, 54
/gnb/project/accounts/new_transaction_multiple.php	+lines 21-27, 66-73, 130
/gnb/project/accounts/verify_transaction.php	+lines 16-21, 59, 108, 146, 165

This issue has been resolved by adding a randomly generated token to all the forms. This token is checked on receiving the values of the form to detect Cross Site Request Forgery.

OTG-SESS-007

SESSION TIMEOUT

Changed files	
config/php5/apache2/php.ini	line 1515

This has been fixed by modifying the PHP settings file.

OTG-INPVAL-001

REFLECTED CROSS SITE SCRIPTING

Changed files	
/gnb/project/genericfunctions.php	new file (+lines 10-103)

Reflected XSS worked on the verify_transaction.php page by printing the description inserted by the user on the previous page. By performing input sanitization on every request parameter, this attack is not longer possible.

OTG-INPVAL-002

STORED CROSS SITE SCRIPTING

Changed files	
/gnb/project/genericfunctions.php	new file (+lines 10-103)

Similarly to the reflected XSS, the site was vulnerable because of lack of input sanitization. All user inputs get now properly sanitized, removing html characters.

OTG-INPVAL-005

SQL INJECTION

Changed files

/gnb/project/db.php	new file (+lines 1-1092)
/gnb/config/database/setup.php, /gnb/project/accounts/account_overview.php, /gnb/project/accounts/download_transactions.php, /gnb/project/accounts/my_accounts.php, /gnb/project/accounts/new_transaction.php, /gnb/project/accounts/transaction_history.php, /gnb/project/accounts/transaction_view.php, /gnb/project/accounts/verify_transaction.php, /gnb/project/authentication.php, /gnb/project/awesome_data.php, /gnb/project/employee/client_details.php, /gnb/project/employee/client_transaction_details.php, /gnb/project/employee/manage_registration.php, /gnb/project/employee/manage_transfer.php, /gnb/project/employee/search_client.php, /gnb/project/employee/transfer_details.php, /gnb/project/models/account.php, /gnb/project/models/transaction.php, /gnb/project/models/user.php, /gnb/project/registration/registration_request.php, /gnb/project/resource_mappings.php	modified all lines with "DB::i()->"
/gnb/project/dbheader.php	removed file

This issue has been resolved by using prepared statements instead of pure `mysql_*` commands. This was accomplished by using PDO and its functionality for building prepared statements. This validates the input for its data type and prevents SQL injections.

OTG-INPVAL-013

COMMAND INJECTION

Changed files	
/gnb/project/accounts/new_transaction_multiple.php	-lines 20-27 +lines 21-30

Since command injection was possible by injecting commands into filenames, this is no longer possible as each uploaded file gets automatically renamed. Even though the c parser receives additional parameters, these are passed directly by the application and cannot be injected by an attacker.

OTG-INPVAL-014

BUFFER OVERFLOW

Changed files	
gnb/project/lib/ctransact/src/ctransact.c	+line 284

The application was never vulnerable to buffer overflow, but only to integer overflow. This was due to a comparison between a long long value and an integer value inside the c parser. By changing the integer to a long long value, this is no longer possible.

OTG-INPVAL-015

INCUBATED VULNERABILITIES

Changed files

See fix slides OTG-INPVAL 001, 002, 005

By having fixed the SQL injection and stored XSS vulnerabilities, this vulnerability was automatically fixed. Please refer to the fix slides concerning OTG-INPVAL 001, 002, 005.

OTG-CRYPST-003

SENSITIVE INFORMATION SENT VIA UNENCRYPTED CHANNELS

Changed files	
N/A	N/A

This was also remediated through the implementation of HTTPS, see OTG-AUTHN-001. SESSION ID is no longer sent in clear text.

OTG-BUSLOGIC-001

BUSINESS LOGIC DATA VALIDATION

Changed files	
/gnb/project/dp.php	+lines 1234-1237

In order to prevent a user incrementing his balance by sending money to his own account, we added a check before performing a transaction. A user is not allowed to use his own account as a destination anymore.

OTG-BUSLOGIC-002

ABILITY TO FORGE REQUESTS

Changed files	
N/A	N/A

This was fixed with the same fix as OTG-SESS-005, see section for more details.

OTG-BUSLOGIC-004

PROCESS TIMING

Changed files	
/gnb/project/db.php	+line 224,238-246

This was fixed by adding a random waiting time to all login requests.

OTG-BUSLOGIC-007

DEFENSES AGAINST APPLICATION MIS-USE

Changed files	
N/A	N/A

Since the described vulnerability was the same as the one presented in OTG-BUSLOGIC-001, please refer to the fix made in that section.

OTG-BUSLOGIC-009

UPLOAD OF MALICIOUS FILES

Changed files	
/gnb/project/accounts/new_transaction_multiple.php	-lines 20-27 +lines 21-30

Each uploaded file now gets automatically renamed. Furthermore, after a batch transaction process finished, the uploaded file gets removed from the uploads folder.

OTG-CLIENT-009

CLICKJACKING

Changed files

/gnb/config/apache2/httpd.conf	+lines 1-7

This has been rectified through the updated security policy and The X-FRAME-OPTIONS has been set to DENY.

USE-CASES

PASSWORD RECOVERY

Goal	Reset a users password in case it is not remembered any longer.
Actors	All users (Customers and Employees)
Pre-Conditions	User account must exist and must know his PIN.
Main course of execution	On the login page click on "Forgot your brocode?" and enter your email address. You will receive instructions to reset your password by mail.
Alternate course	n/a
Post conditions	Password is set to a new password by the user.
Exceptions	Clicking on the received reset hash took longer than 1 day. It is necessary to get repeat the whole procedure.
Data format used	n/a

ENCRYPTED TAN GENERATION AND DELIVERY VIA E-MAIL TO CUSTOMER

Goal	Send the TANs in an encrypted PDF to the customer.
Actors	Customers that are using standard TANs.
Pre-Conditions	Customer is approved and did select "TANs" as banking method at the registration.
Main course of execution	Create new account, wait for approval by employee, log in and copy the shown password in order to open the encrypted PDF that was received by mail.
Alternate course	n/a
Post conditions	The customer has access to his TANs provided by the decrypted PDF.
Exceptions	n/a
Data format used	Encrypted PDF file

DOWNLOAD OF SCS AFTER REGISTRATION

Goal	Download the SmartCardSimulator after SCS was selected as the banking method.
Actors	Customers that are using the SCS.
Pre-Conditions	Customer is approved and did select "SCS" as banking method at the registration.
Main course of execution	Create new account, wait for approval by employee and log in. Write down the provided PIN and download the provided Java-Application "SmartCardSimulator".
Alternate course	n/a
Post conditions	The customer has access to the SCS and can generate TANs using his PIN, the destination account number andthe transaction amount OR a transaction batch file including the aforementioned information.
Exceptions	n/a
Data format	Java Application (requires Java 1.7)

TRANSFER USING SCS (SINGLE TRANSACTION)

Goal	Create transactions using the SmartCardSimulator.
Actors	Customers that are using the SCS.
Pre-Conditions	Customer is approved, did select "SCS" as banking method at the registration, downloaded the SCS and remembers the PIN.
Main course of execution	Customers enter the transaction details (destination account, transaction amount) and their PIN into the web interface and the SCS, and copy the TAN generated by the SCS into the web interface. The transaction is then posted and executed.
Alternate course	see "Transfer using SCS - Batch transaction"
Post conditions	The customer did create transactions using the SmartCardSimulator.
Exceptions	n/a
Data format used	Java Application (requires Java 1.7), batch processing format see "Batch transfers should allow multiple transfers via the same uploaded file"

TRANSFER USING SCS (BATCH TRANSACTION)

Goal	Execute multiple transactions sequentially by uploading a batch transaction file. The legitamacy of these transactions is verified by a TAN generated by the SCS.
Actors	Customer
Pre-Conditions	Customer must be logged in, have sufficient funds for all transactions and remember his PIN displayed at registration.
Main course of execution	Logged in Customer switches to "My Accounts" page, selects an account, selects "New transaction (multiple)", chooses a file. He also enters a tan, which is generated by additionally opening the batch file in the SCS application in combination with his pin.
Alternate course	Single transaction method. Entering every transaction individually via the web form as described above
Post conditions	None
Exceptions	Invalid batch file format. Insufficient funds.
Data	Data format explained in detail on the next slide and on the "New transaction

BATCH TRANSACTION: DATA FORMAT

- Each transaction in a separate line
- Fields separated only by a comma
- No quoting of whitespace required
- The complete input between two commas gets treated as value
- Example: DST_ACCOUNT,AMOUNT,DESCRIPTION

TIME-TRACKING

ALEXANDER LILL

Task	Time
Research fix: SQL Injection	1h
Fix: SQL Injection - Login / Misc	1h
Fix: SQL Injection - User functions	1h
Fix: SQL Injection - Account functions	1h
Fix: SQL Injection - TAN functions	1h
Fix: SQL Injection - Transaction functions	1h
Fix: SQL Injection - Overview functions	0,5h
Use DB as Singleton object and change all calls	1h
Adjust DB schema to new requirements	1h
Add PIN functionality to DB and PHP	1h
Fix: Lockout - Add unblock user functionality	1h
Fix: Lockout functionality	1h
Testing Lockout and unblock functionality	1h
Feature: SCS TAN handling (DB)	1h
Feature: Implement reset password functionality (DB)	1h
Prepare presentation	1h
Prepare deliverables	1h
Fixing balanice initialization bugs	1h
Adding empty templates for vulnerabilits to presentation	1h
Fix: Process timing	1h
TOTAL	**19,5h**

LORENZO DONINI

Task	Time
Group meeting	1h
Fix: resource mappings	1h
Fix: session	0.5h
Fix: password	0.5h
Fix: lockout mechanism	1h
Features/Use-cases testing	1h
Features/Use-cases bugfixing	1h
Registration: new business logic	1h
Registration: new GUI	1h
Registration: testing & fixing	1h
Java SCS: GUI	1h
Java SCS: Business logic	1h
Java SCS: TAN generation	1h
Java SCS: Testing	1h
Transactions: single transactions logic changed	1h
Transactions: batch transactions logic changed	1h
Initialization of client balance added	1h
Presentation: core slides	1h
Presentation: fix slides	1h
TOTAL	**18h**

FLORIAN MAURACHER

Task	Time
Group meeting	1h
Read BBTesting report Team5	0.5h
Discuss Phase3 and coordinate work	1h
Fix mail delivery	1h
Secure apache2 configuration	1h
Secure php configuration	1h
Prepare VM for Phase3	1h
Develop mechanism for TAN generation	1h
Discuss mechanism for TAN generation	1h
Fix vulnerabilities in fileupload	1h
Multiple transactions, one TAN (PHP)	1h
Multiple transactions, one TAN (C)	1h
Verify SCS TANs for batch transactions	1h
Debug SCS logic for batch transactions	1h
Fix timestamps for batch transactions	0.5h
Presentation: Fixes	1h
Presentation: Review	1h
Prepare delivery VM	1h
TOTAL	**17h**

MAHMOUD NASER

[F]ix, [R]esearch, [I]mplement	
[T]est, [P]resentation, [D]esign	
[R] PW PDF	1h
[I] FPDI library	1h
[R] bugs with FPDI	1h
[I] FPDI bug fixes	1h
[I] FPDF library	1h
[T] FPDF PW protection	1h
[T] PDF PW on GNB Website	1h
[F] PDF PW bugs	1h
[D] PW reset logic	0.5h
[D] PW reset pages	1h
[D] PW forgot logic	0.5h
[D] PW forgot pages	1h
[D] email template	1h
[C] Token generetaion mechanisim	1h
[F] PW email bugs	1h
[C] Transaction history fields	1h
[C] PDF transaction history fields	1h
[F] Transaction bugs	1h
[P] adding content	1h
TOTAL	**18h**