INF226 – Software Security

Håkon Robbestad Gylterud

2019–08–19

2018 British Airways breach

According to BBC (2018-09-07):

> *Alex Cruz[a] told the BBC that hackers carried out a "sophisticated, malicious criminal attack" on its website.*
> *The airline said personal and financial details of customers making or changing bookings had been compromised. About 380,000 transactions were affected, (· · ·)*

[a] Head of British Airways

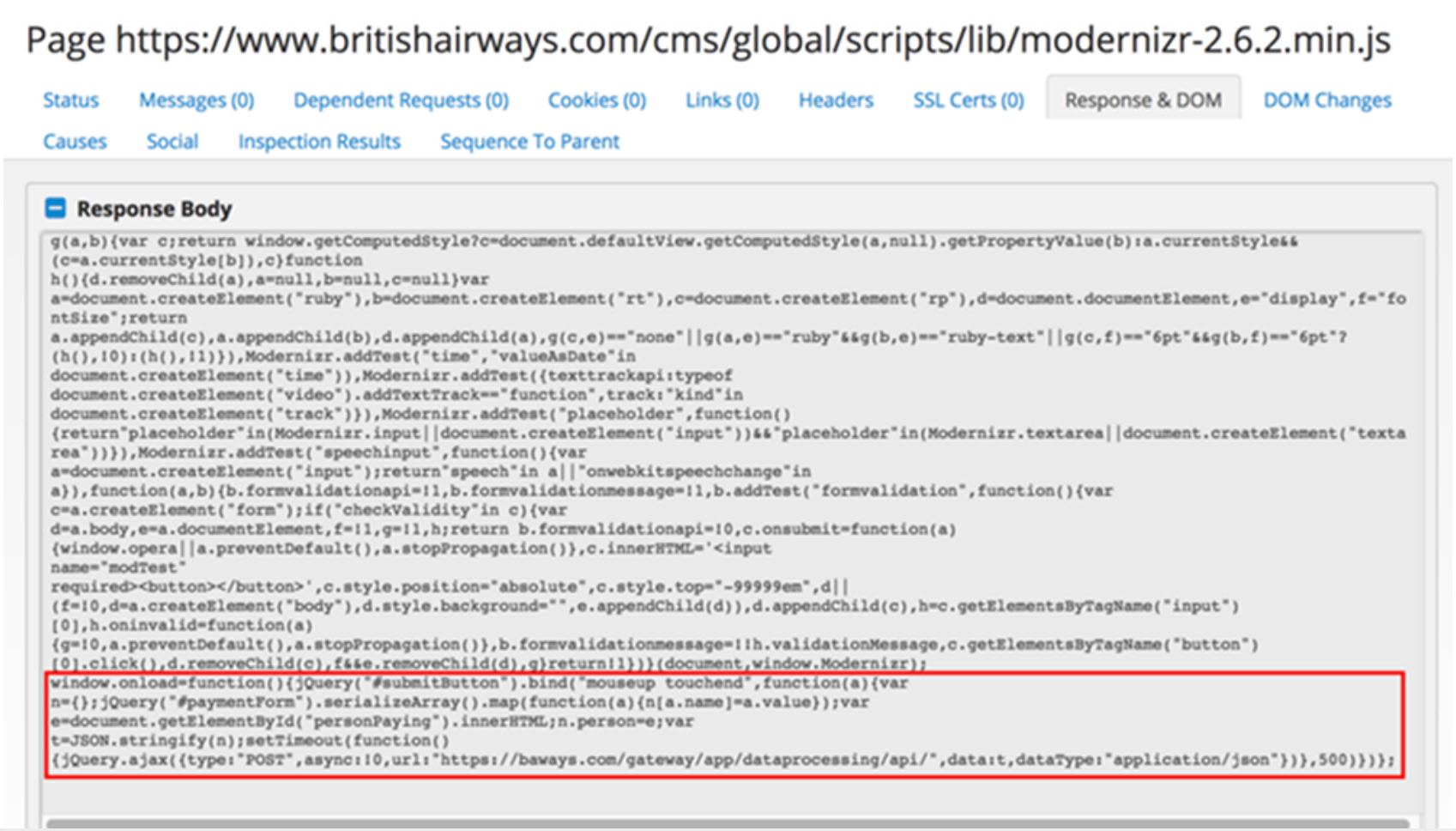
2018 British Airways breach

The ICO[1] informs:

> *Following an extensive investigation the ICO has issued a notice of its intention to **fine British Airways £183.39M*** for infringements of the General Data Protection Regulation (**GDPR**).*
>
> *The proposed fine relates to a cyber incident notified to the ICO by British Airways in September 2018. This incident in part involved user traffic to the British Airways website being diverted to a fraudulent site. Through this false site, customer details were harvested by the attackers. Personal data of approximately **500,000 customers** were compromised in this incident, which is believed to have begun in June 2018.*

[1] ICO corresponds to the Norwegian Datatilsynet.

2018 British Airways breach



INF226 Autumn 2019

Lectures

Håkon Robbestad Gylterud hakon.gylterud@uib.no

Group sessions

Benjamin Chetioui	benjamin.chetioui@uib.no
Erlend Bøhler Nærbø	erlend.nerbo@student.uib.no
Joakim Moss Grutle	joakim.grutle@student.uib.no

Learning outcomes

The student

- understands security issues relating to system development,
- knows software development techniques to avoid security problems,
- can **explain the most common weaknesses in software security** and how such problems can be mitigated in software,
- can identify common security threats, risks, and attack vectors for software systems, and
- knows best practices to defend software systems.

Learning outcomes

The student

- can use **tools to discover security problems** in software,
- masters, theoretically and practically, programming techniques to develop secure, safe, reliable, and robust systems, and
- can assess the security of given source code or application.

Two approaches to software security

What are the security holes in the program?

How to design the program securely?

Terms and definitions

Defining security

For the purpose of this course:

Definition

Software security is the ability of software to function according to intentions in an adverserial environment.

Variables

For the purpose of this course:

Definition

Software security is the ability of software to function according to **intentions** in an adverserial environment.

Variables

For the purpose of this course:

Definition

Software security is the ability of software to function according to intentions in an **adverserial environment**.

Example

Demonstration.

Requirements, assumptions and mechanisms

Logic

Prove that every triangle is a polygon!

Logic

Prove that every triangle is a polygon!

Logical arguments have three parts:

- Conclusion
- Assumptions
- Deduction

Software security has three parts

Figure 2: Requirements, assumptions and mechanisms

Software security has three parts

Figure 2: Requirements, assumptions and mechanisms

1. **Identify security requirements** which capture the intentions for the software.
2. **Make explicit the assumptions** about the environment the software will run.
3. **Design mechanisms** which satisfy the requirements given the assumptions.

Example: Instant messaging (for private persons)

Think of:

- Two security requirements.
- Assumptions related to these requirements.
- Mechanisms to satisfy these requirements.

Vulnerabilities and exploits

Definition

A **vulnerability** in a software is a circumstance in which the program fails to behave according to intentions.

Definition

An **exploit** of a vulnerability is a procedure which upon execution leads to the circumstance described by the vulnerability, thus demonstraiting the insecurity of the program.

Next time

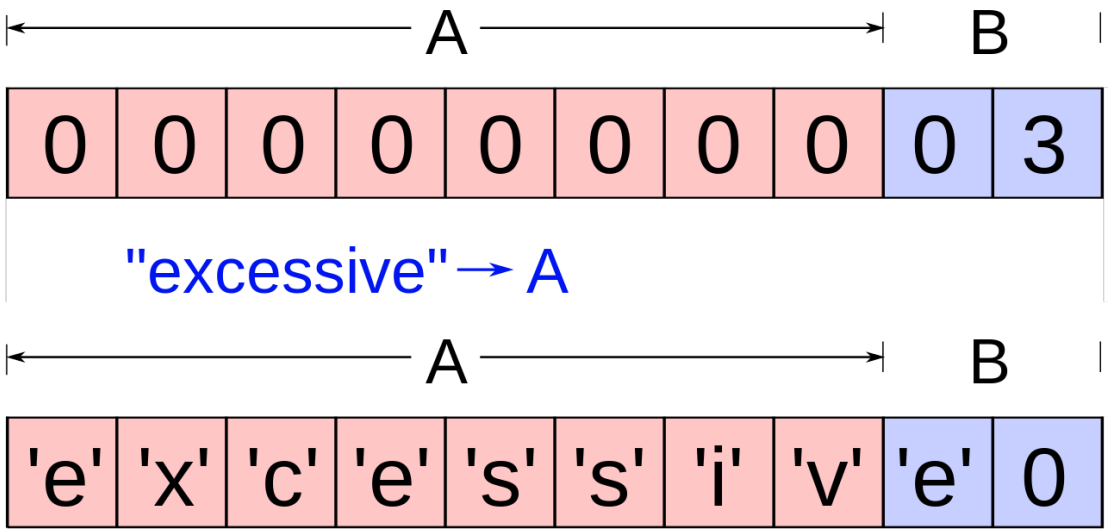


Figure 3: Buffer overflows!

Learning material

Course books

- *Secure and resilient software development*, Merkow
- *Security for software engineers*, Helfrich

Other sources

Can be found at the syllabus page on MittUiB.

Evaluation

Mandatory assignment

30% of your grade is determined by the mandatory assignments.

Assignment 1	Buffer overflow and SQL injection	15th of September
Assignment 2	Security analysis of software	13th of October
Assignment 3	Writing security critical code	10th of November

You are expected to do these during the group sessions.

Exam

- Written exam
- 70% of the grade