# PA193 - Secure coding principles and practices

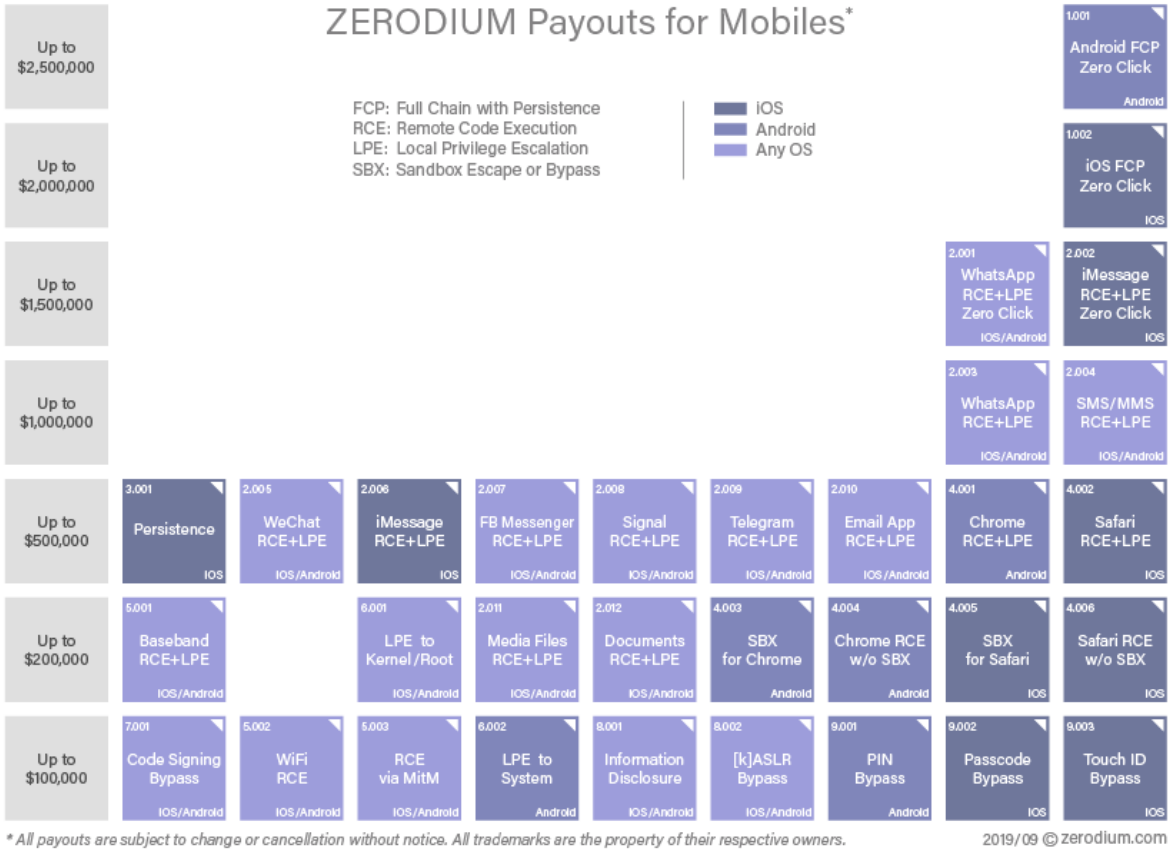
## Security Code Review

*Seminar*

**Łukasz Chmielewski** *chmielewski@fi.muni.cz*
Centre for Research on Cryptography and Security, Masaryk University

CROCS
Centre for Research on Cryptography and Security

# Profit? Risks? Ethics?




https://zerodium.com/program.html

# Bug Bounties

## Rewards Program Hall of Fame

Every year we select top 10 security researchers who have made an impact on helping us improve the security of our products and services, and we show our gratitude to them with the Hall of Fame. We would like to thank them for disclosing the vulnerability reports responsibly and working with us throughout the process.

2023 2022 2021 2020

| Rank | Name | SVE | | | | |
|---|---|---|---|---|---|---|
| 1 | Daniel Komaromy of TASZK Security Labs | SVE-2023-0541 | SVE-2023-0539 | SVE-2023-0538 | SVE-2023-0537 | SVE-2023-0536 |
| | | SVE-2023-0535 | SVE-2023-0534 | SVE-2023-0533 | SVE-2023-0532 | SVE-2023-0531 |
| 2 | Mohamed Taha | SVE-2023-1795 | SVE-2023-1540 | SVE-2023-1539 | SVE-2023-1538 | SVE-2023-1537 |
| | | SVE-2023-1536 | SVE-2023-1535 | SVE-2023-1534 | SVE-2023-1533 | SVE-2022-3016 |
| | | SVE-2022-2889 | SVE-2022-2888 | SVE-2022-2887 | SVE-2022-2886 | SVE-2022-2885 |
| | | SVE-2022-2880 | SVE-2022-2879 | SVE-2022-2878 | SVE-2022-2877 | SVE-2022-2876 |
| | | SVE-2022-2875 | SVE-2022-2874 | SVE-2022-2873 | SVE-2022-2872 | SVE-2022-2870 |
| | | SVE-2022-2869 | SVE-2022-2868 | SVE-2022-2867 | SVE-2022-2866 | SVE-2022-2865 |
| | | SVE-2022-2862 | SVE-2022-2861 | SVE-2022-2860 | SVE-2022-2859 | SVE-2022-2858 |
| | | SVE-2022-2857 | SVE-2022-2856 | SVE-2022-2855 | SVE-2022-2851 | SVE-2022-2850 |
| | | SVE-2022-2846 | SVE-2022-2845 | SVE-2022-2844 | SVE-2022-2843 | SVE-2022-2842 |
| | | SVE-2022-2841 | SVE-2022-2840 | SVE-2022-2839 | SVE-2022-2838 | SVE-2022-2834 |
| 3 | Oversecured Inc | SVE-2023-1595 | SVE-2023-0993 | SVE-2023-0989 | SVE-2023-0987 | SVE-2023-0963 |
| | | SVE-2023-0938 | SVE-2023-0928 | SVE-2023-0760 | SVE-2023-0759 | SVE-2023-0668 |
| | | SVE-2023-0667 | SVE-2023-0653 | SVE-2023-0622 | SVE-2023-0611 | SVE-2023-0593 |
| | | SVE-2023-0072 | SVE-2022-2399 | SVE-2022-2398 | SVE-2022-2338 | SVE-2022-2328 |
| | | SVE-2022-2320 | SVE-2022-2296 | SVE-2022-2280 | SVE-2022-2278 | SVE-2022-2261 |
| | | SVE-2022-2212 | SVE-2022-2118 | SVE-2022-1931 | SVE-2022-1672 | |

https://security.samsungmobile.com/hallOfFameInfo.smsb

# Outline

- Many simple exercises
  - looking at common mistakes in pairs.
- Topics:
  - Protecting Data, Preventing Cross-Site Scripting, Code Quality,
  - Memory Best Practices, Parameterized Statements,
  - Indirect Object References, and Input Validation…
- Explanation for the Assignment.
- That is all ☺

# Disclaimer

- You often do not know some technology or details or some function.
- Try to guess what might be wrong.
- We discuss it together so if you have idea but you are not sure:
  - Try and we will discuss it!

# SIMPLE EXERCISES

# Simple Exercises

- Form pairs (e.g., with your neighbour)
- Look and code together (before ready to answer the question)
- Two roles:
  - Educator – explains the answer to the given question to his/her pair
  - Sceptic – tries to find any flaw or weak point in Educator's reasoning
- Together try to find an answer on what is wrong in the code.
  - What can be a root of the issue?
  - Propose a correction.
- Switch roles after every question (from next slide)

# Exercise (1): what is wrong with this class?

```
public class Account {
double principal,rate; int daysActive,accountType;
public static final int STANDARD=0, BUDGET=1, PREMIUM=2,
PREMIUM_PLUS=3;
}
...
public static double calculateFee(Account[] accounts)
{
double totalFee = 0.0;
Account account;
for (int i=0;i<accounts.length;i++) {
 account=accounts[i];
        if(account.accountType==Account.PREMIUM|| account.accountType
== Account.PREMIUM_PLUS )
        totalFee += .0125 * ( // 1.25% broker's fee
        account.principal*Math.pow
        (account.rate,(account.daysActive/365.25))
        - account.principal); // interest-principal
}
return totalFee;
}
```

"Applied Software Project Management" by Andrew Stellman and Jennifer Greene

# Exercise (2): what is wrong and how to improve it?

```
String updateServer = request.getParameter("updateServer");
if(updateServer.indexOf(";")==-1 && updateServer.indexOf("&")==-1){
    String [] commandArgs = {
        Util.isWindows() ? "cmd" : "/bin/sh",
        "-c", "ping", updateServer
    }
    Process p = Runtime.getRuntime().exec(commandArgs);
}
```

https://owasp.org/

# Exercise (2): what is wrong and how to improve it?

```
String updateServer = request.getParameter("updateServer");
if(ValidationUtils.isAlphanumericOrAllowed(updateServer,'-','_','.')){
    String [] commandArgs = {
        Util.isWindows() ? "cmd" : "/bin/sh",
        "-c", "ping", updateServer
    }
    Process p = Runtime.getRuntime().exec(commandArgs);
}
```

# Exercise (3): what is wrong and how to improve it?

```
String updateServer = request.getParameter("updateServer");
String cmdProcessor = Utils.isWindows() ? "cmd" : "/bin/sh";
String command = cmdProcessor + "-c ping " + updateServer;

Process p = Runtime.getRuntime().exec(command);
```

# Exercise (3): what is wrong and how to improve it?

```
String updateServer = request.getParameter("updateServer");
List<String> commandArgs = new ArrayList<String>();
commandArgs.add("ping");
commandArgs.add(updateServer);
ProcessBuilder build = new ProcessBuilder(commandArgs);
```

# Exercise (4): what is wrong and how to improve it?

```
String query = String.format("SELECT * FROM users WHERE usr='%s' AND pwd='%s'", usr, pwd);
Connection conn = db.getConn();
Statement stmt = conn.createStatement();


ResultSet rs = stmt.executeQuery(query);
```

# Exercise (4): what is wrong and how to improve it?

```
String query = "SELECT * FROM users WHERE usr = ? AND pwd = ?";
Connection conn = db.getConn();
PreparedStatement stmt = conn.preparedStatement(query);
stmt.setString(1, usr);
stmt.setString(2, pwd);
ResultSet rs = stmt.executeQuery(query);
```

# Exercise (5): what is wrong and how to improve it?

```
printf("Enter the master password:\n");
gets(userPass);

if(strncmp(userPass,MASTER_PASSWORD,9)==0){
    printf("PASSWORD VERIFIED\n");
}
```

# Exercise (5): what is wrong and how to improve it?

```
printf("Enter the master password:\n");
fgets(userPass,9,stdin);

if(strncmp(userPass,MASTER_PASSWORD,9)==0){
    printf("PASSWORD VERIFIED\n");
}
```

# Exercise (6): what is wrong and how to improve it?

```
char userPass[5];

printf("Enter the master password:\n");
fgets(userPass,9,stdin);

if(strncmp(userPass,MASTER_PASSWORD,BUFFER_SIZE)==0){
    printf("PASSWORD VERIFIED\n");
}
```

# Exercise (6): what is wrong and how to improve it?

```
int BUFFER_SIZE = 9;
char userPass[BUFFER_SIZE];

printf("Enter the master password:\n");
fgets(userPass,BUFFER_SIZE,stdin);

if(strncmp(userPass,MASTER_PASSWORD,BUFFER_SIZE)==0){
    printf("PASSWORD VERIFIED\n");
}
```

# Exercise (7): what is wrong and how to improve it?

```
int len = 0, total = 0;
while(1){
    fgets(buff1, MAX_SIZE, stdin);
    int len =  strnlen(buff1, MAX_SIZE);
    total += len;
    if(total <= MAX_SIZE) strncat(buff2, buff1, len);
    else break;
}
```

# Exercise (7): what is wrong and how to improve it?

```
int len = 0, total = 0;
while(1){
    fgets(buff1, MAX_SIZE, stdin);
    int len =  strnlen(buff1, MAX_SIZE);
    total += len;
    if(total < MAX_SIZE) strncat(buff2, buff1, len);
    else break;
}
```

# Exercise (8): what is wrong and how to improve it?

```
if(strncmp(userPass,MASTER_PASSWORD,BUFFER_SIZE)==0){
    printf("PASSWORD VERIFIED\n");
}
else{
    printf("Invalid password:");
    printf(userPass);
}
```

# Exercise (8): what is wrong and how to improve it?

```
if(strncmp(userPass,MASTER_PASSWORD,BUFFER_SIZE)==0){
    printf("PASSWORD VERIFIED\n");
}
else{
    printf("Invalid credentials.");
}
```

# Exercise (9): what is wrong and how to improve it?

```
String usr = request.getParameter("usr");
String pwd = request.getParameter("pwd");
User user = UserColl.find(usr);

if(user.getPassword().equals(pwd)){

    //password verified
```

# Exercise (9): what is wrong and how to improve it?

```
String usr = request.getParameter("usr");
String pwd = request.getParameter("pwd");
User user = UserColl.find(usr);
String givenValue = Utils.PBKDF2(pwd, user.getSalt(), user.getIterations());
if(user.getPassHash().equals(givenValue)){

    //password verified
```

# Exercise (10): what is wrong and how to improve it?

```
String url = "http://my-service.cloud.biz/Login?usr="+usr+"&pwd="+pwd;
URL obj = new URL(url);
HTTPURLConnection con = (HTTPURLConnection) obj.openConnection();
con.setRequestMethod("GET");
con.setRequestProperty("User-Agent", USER_AGENT);
```

# Exercise (10): what is wrong and how to improve it?

```
String url = "https://my-service.cloud.biz/Login";
URL obj = new URL(url);
HTTPURLConnection con = (HTTPURLConnection) obj.openConnection();
con.setRequestMethod("POST");
con.setRequestProperty("User-Agent", USER_AGENT);
```

# Exercise (11): what is wrong and how to improve it?

```
var transaction = {"custName":custName,"address":custAddress,"creditCardNumber":custCC.CCPAN};


s3.putObject({
    "Bucket": "ACME-customer-billing",
    "Key": "todayTransactions",
    "Body": JSON.stringify(transaction),
    "Content-Type": "application/json"
},
function(err,data){
});
```

# Exercise (11): what is wrong and how to improve it?

```
var transaction = {"custName":custName,"address":custAddress,"creditCardNumber":dataCleaner.removeCCPAN(custCC)};
var encTransaction = cryptUtils.AES256GCM(transaction, secretsManager);
s3.putObject({
    "Bucket": "ACME-customer-billing",
    "Key": "todayTransactions",
    "Body": JSON.stringify(encTransaction),
    "Content-Type": "application/json"
},
function(err,data){
});
```

# Exercise (12): usage of HTML encoding, what is wrong and how to improve it?

```
<div class="form-group">
    <label for="search">Search:</label>
    <input type="text" class="form-control" id="search" name="search">

    <input type="submit" id="submit" class="btn" value="Search">
    <div class="alert alert-danger <%=alertVisibility%>">
        Cannot find <%=request.getParameter("search")%>
    </div>
</div>
```

# Exercise (12): usage of HTML encoding, what is wrong and how to improve it?

```
<div class="form-group">
    <label for="search">Search:</label>
    <input type="text" class="form-control" id="search" name="search">

    <input type="submit" id="submit" class="btn" value="Search">
    <div class="alert alert-danger <%=alertVisibility%>">
        Cannot find <%=StringEscapeUtils.escapeHtml4(request.getParameter("search"))%>
    </div>
</div>
```

# Exercise (13): what is wrong and how to improve it?

- The application is implementing its own client side rendering of the input instead of taking advantage of a JS framework.

```
$get("/profile", function(data, status){
    if(data!=null){
        var dataArgs = data.split(",");
        if(dataArgs.length > 1){
            var displayName = dataArgs[0];
            var displayNameDiv = $("#displayNameDiv")[0];
            displayNameDiv.innerHTML = displayName;
            var avatarImg = $("#avatarImg")[0];
            avatarImg.src = dataArgs[1];
        }
    }
});
```

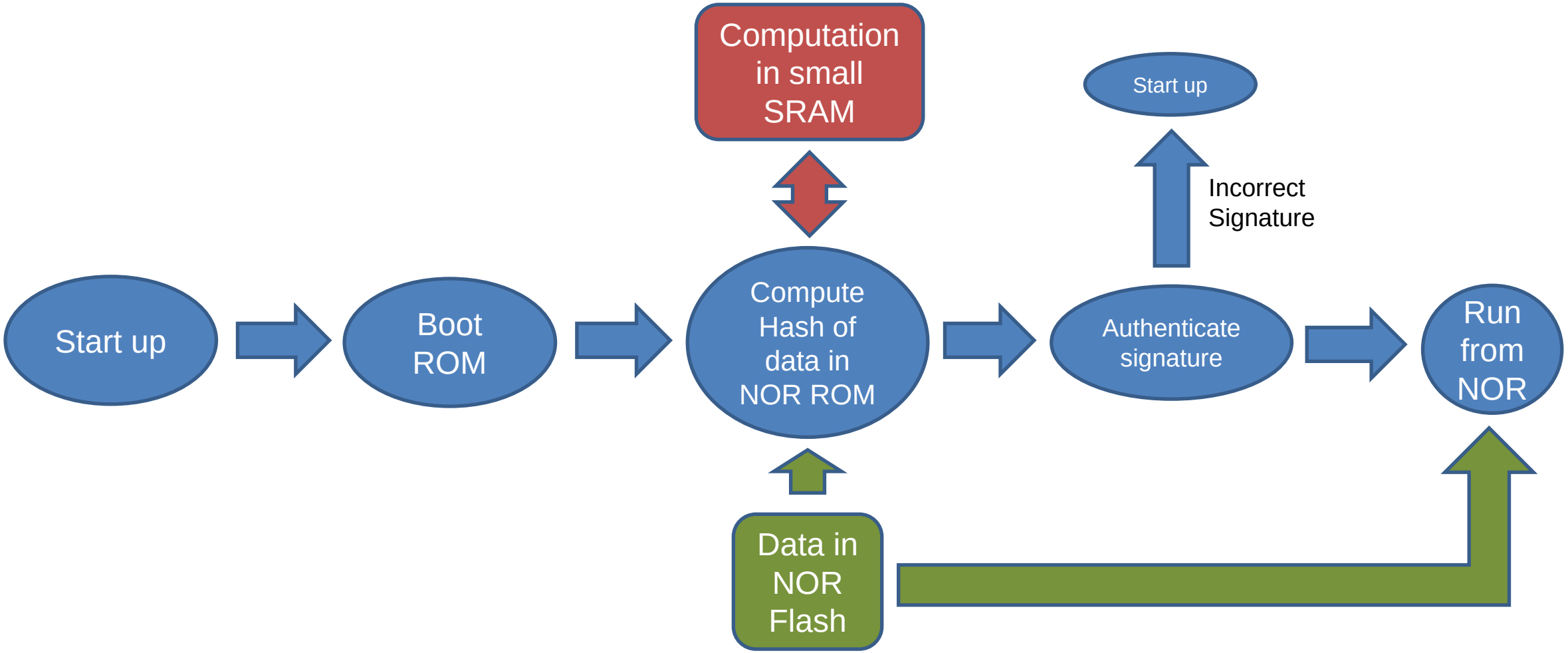
# Exercise (14): what is wrong and how to improve it?

```
String file = request.getParameter("file");
file = "public/"+file;
InputStream input = null;
BufferedReader reader = null;
StringBuilder sb = new StringBuilder();
input = getServletContext().getResourceAsStream(file);
```

# Exercise (14): what is wrong and how to improve it?

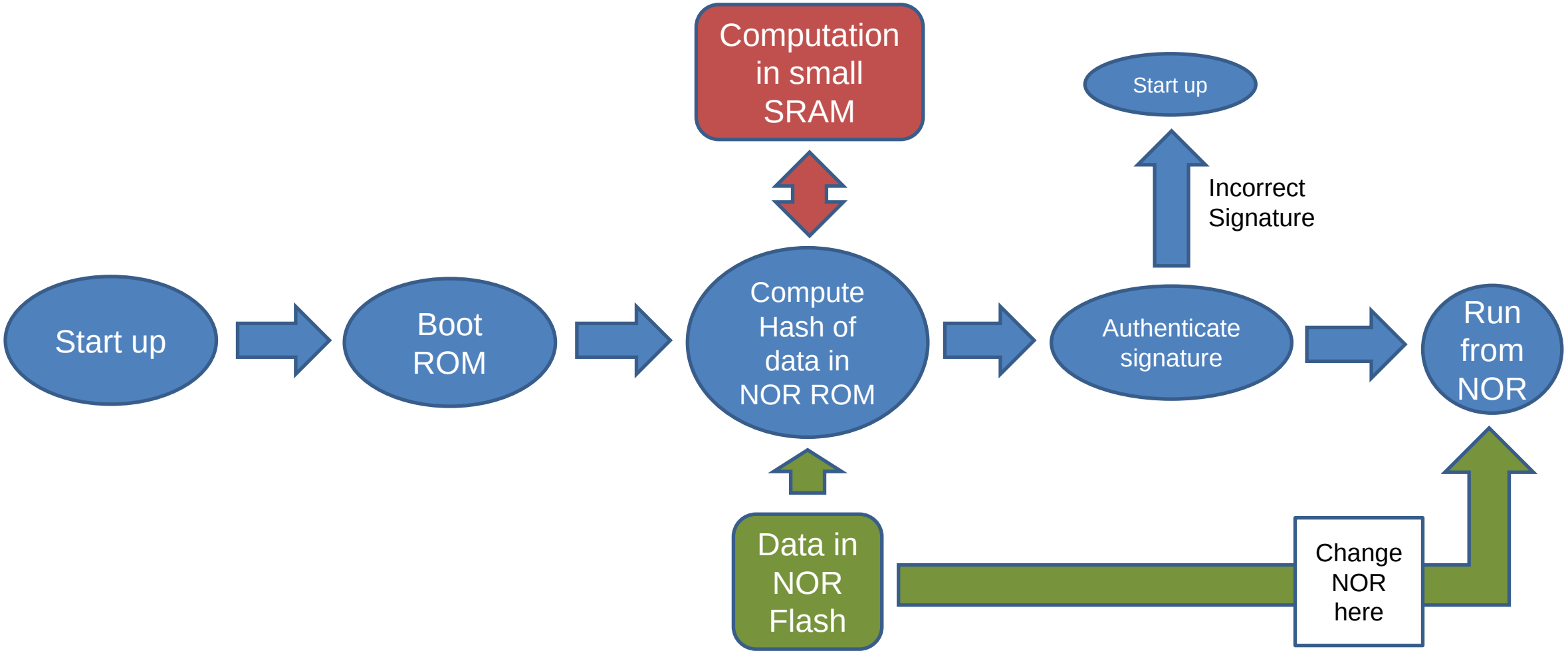
```
String fileId = request.getParameter("fileId");
file = "public/"+availableFiles[fileId];
InputStream input = null;
BufferedReader reader = null;
StringBuilder sb = new StringBuilder();
input = getServletContext().getResourceAsStream(file);
```

# Exercise (15): what is wrong with the design and how to improve it?



# Exercise (15): what is wrong with the design and how to improve it?

20 Ways Past Secure Boot" by Job de Haas (2014)
https://www.youtube.com/watch?v=74SzIe9qiM8&ab_channel=TROOPERScon



# Optional Exercise (16): what is wrong and how to improve it?

- Have a look at the following files:
  - https://github.com/praetorian-inc/DVRF/blob/master/Pwnable%20Source/ShellCode_Required/socket_cmd.c
  - https://github.com/praetorian-inc/DVRF/blob/master/Pwnable%20Source/ShellCode_Required/socket_bof.c
- What is wrong with them?
- Try to imagine findings summary for them (like in the lecture).
- Example:

**Problem identification: DSA-1571-1 openssl**
**Severity**: critical
**Risk:** high - directly exploitable by external attacker
**Problem description**: crypto/rand/md_rand.c:276 & 473 – The random number generator in Debian's openssl package is predictable. This is caused by an incorrect Debian-specific change to the openssl package. One of the sources of a randomness based on usage of uninitialized buffer *buff* is removed.
**Remediation**: revert back to usage of uninitialized buffer *buff*

# Optional Exercise (17): what is wrong and how to improve it?

- In load nitro firmware memory in:
- https://github.com/OP-TEE/optee_os/blob/3.14.0/core/pta/bcm/elog.c

# Optional Exercise (17): what is wrong and how to improve it?

- Solution:
- https://github.com/OP-TEE/optee_os/security/advisories/GHSA-hhrc-h9xj-hppv
- Real issue that was found.
- What is the impact?

# Harder Task?

Spot The Vuln - Is it Clear?

```
1  #define clear(s,x) memset((x), 0, sizeof(s));
2
3  struct hdr { int flags; short len; };
4  struct opts { char opt1; char opt2; char opt3; };
5  struct packet {
6      struct hdr h;
7      struct opts o;
8  };
9
10 int main() {
11     struct packet* p = malloc(sizeof(struct hdr) + sizeof(struct opts));
12     /* Do some processing */
13     clear(struct packet, p);
14     return 0;
15 }
```

dayzerosec

# ASSIGNMENT – CODE REVIEW

# Assignment 6: Source Code Review

- 2 sub-exercises
- `pin.c`
  - Incomplete 32-bit SIM smartcard application in C in the JavaCard style.
    - Exposed functions are being called directly from the APDU handler. That code sets all the lengths and offsets correctly.
    - The APDU handler and the main functions are skipped here since they are not relevant from the security point of view.
  - Find all the possible bugs. Scope: logical and side-channel issues
  - 5 points.
- `server_articles.c`, `server_setup.sh`
  - Find all the possible bugs. Scope: concentrate on logical issues
  - 5 points for finding at least 6 significantly different issues, including 3 high-severity ones
    - it needs to be justified why they are high severity.
  - Bonus: 1 (or even 2) extra points for finding more issues.
- For found issues: asses severity, risk, etc., like in the lecture;
  - also give recommendations on how to improve.
- There is no need to use automatic tooling, but you can do it if you would like to.

# Assignment 6 – what to submit

- Report any issues found in the format presented in the lecture.
- Try to be compact but clear!
- Specify which editor or IDE you use. Also if any static analysis tools you used (for the second exercise).
- Submit **before 16.5.2024 23:59** into IS HW vault
  - Soft deadline: -3 points for every started 24 hours
- Good luck!!!
- Consultation
  - Regular consultation on Friday 09.30 – 11.00 in my office: A406.
  - Email me to make an appointment: chmiel@fi.muni.cz.

# Conclusions

- A lot of different topics for source code review
- Just a shallow glance
- Many topics not touched, like boot loaders, crypto libraries, etc.
- Good luck with the exercise!

# Questions ?