

Hack to the future

Marinus Kuivenhoven
Senior Security Specialist

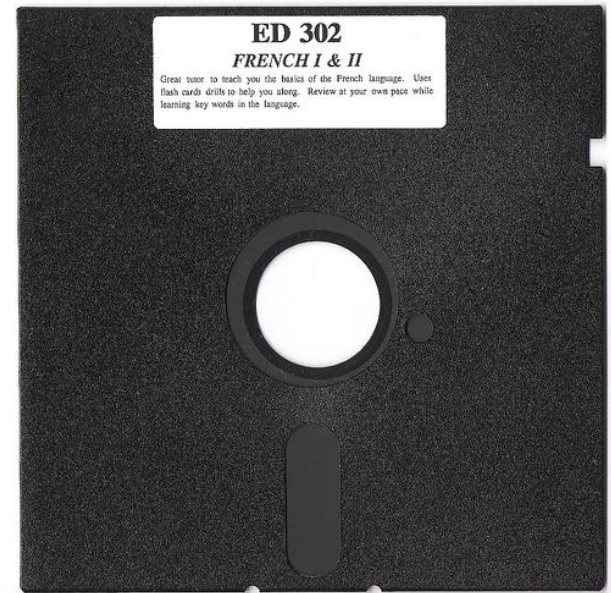


#hitb2012ams

Commodore 64



Commodore 1541 floppy drive



Copy protection

- Difference in resources to buy and use vs. copy and use



Copy protection C64 #1

- **Bad Sector (1983)**



Copy protection reboot

- **Xbox 360 (2005)**



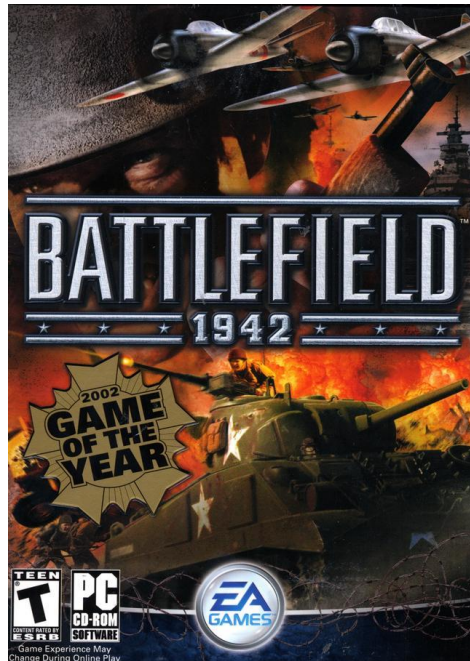
Copy protection C64 #2

- **Gap Bytes (1985)**



Copy protection reboot

- PC games with SafeDisc (2002 -)



Copy protection C64 #3

- **Trackalignment (1987)**



Copy protection reboot

- **PlayStation 2 (2000)**
 - Afwijkende ruimte tussen blocks



So..

**We don't learn from
other people mistakes**

The 'F'-test

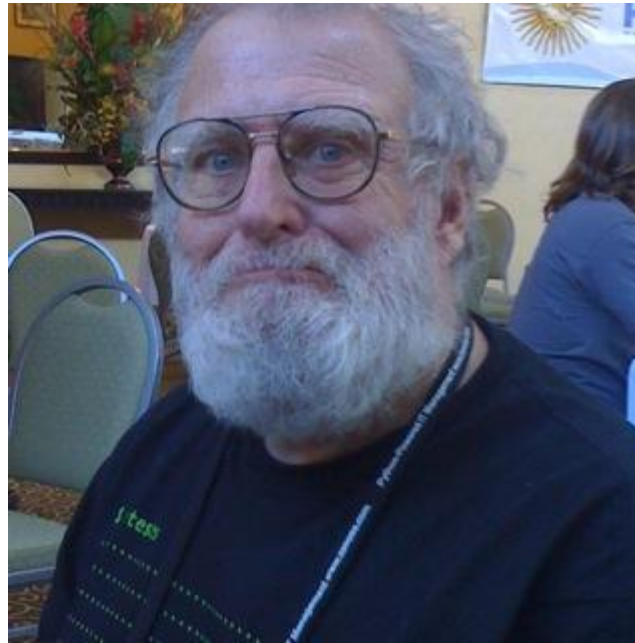
An infection is the invasion of body tissues by disease-causing microorganisms, their multiplication and the reaction of body tissues to these microorganisms and the toxins that they produce. Hosts normally fight infections themselves via their immune system.

Assignment:

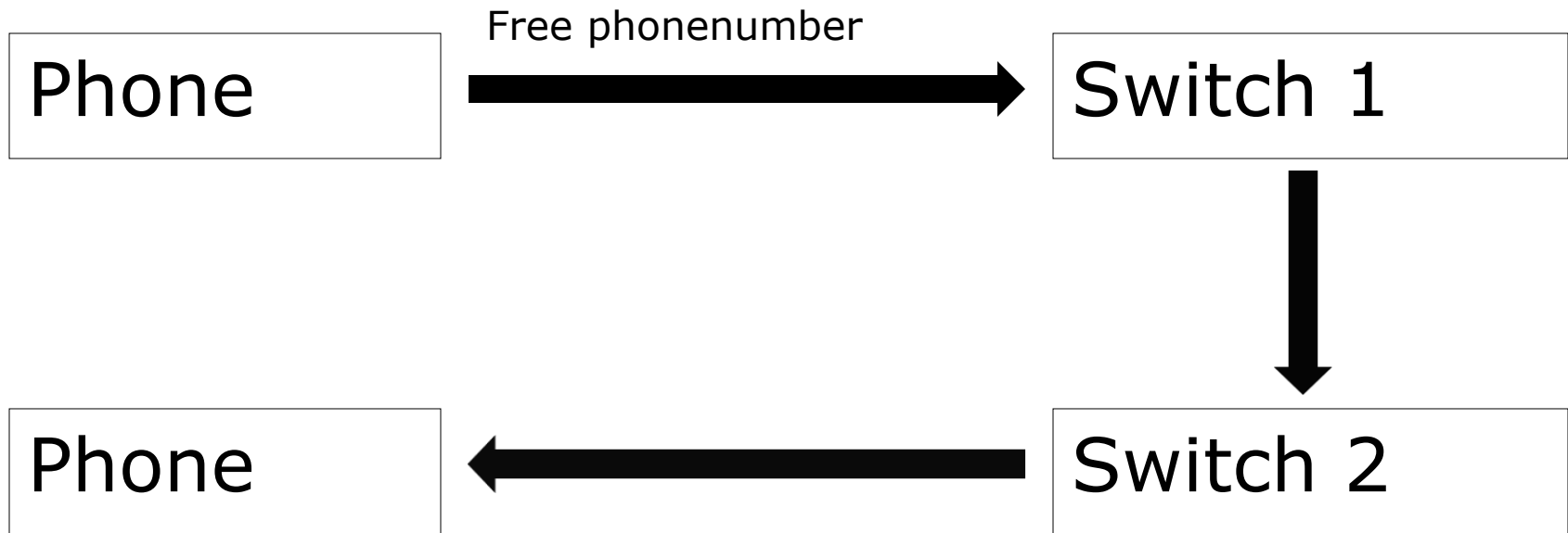
Count the number of times the character 'F' appears on this slide. Write it on a paper, put your fingers up or yell the amount. Don't cheat and count it in one go.

Phone + Freak = Phreak

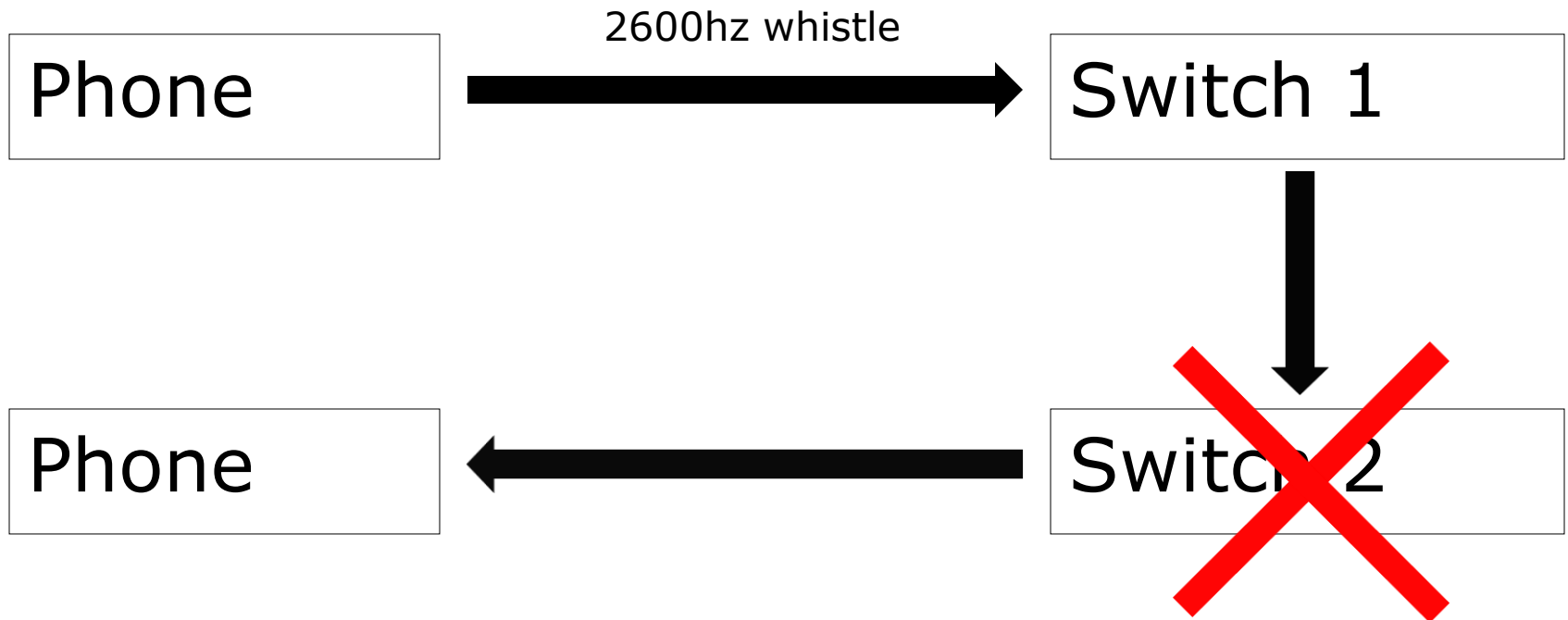
"Joybubbles" & "Captain Crunch"



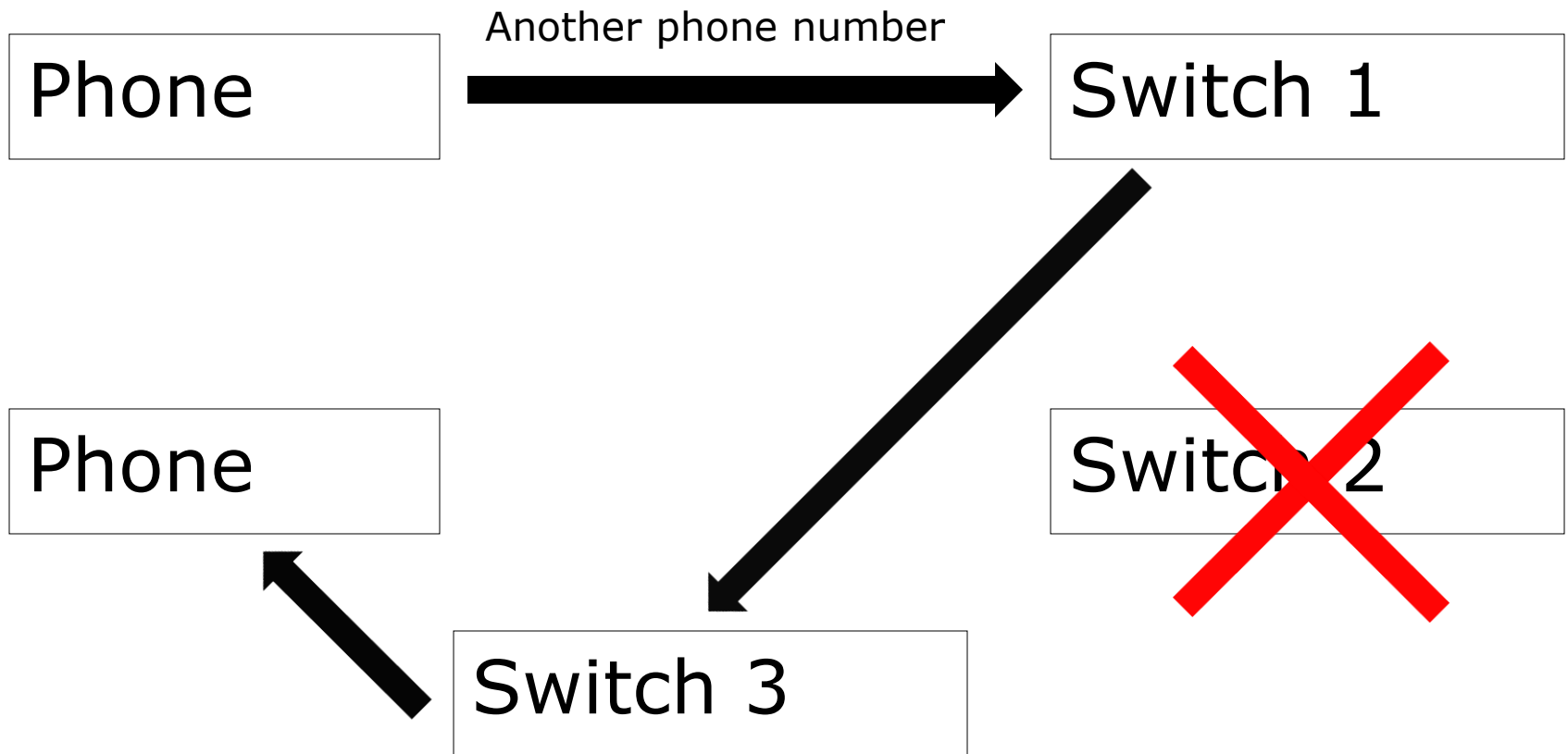
Bluebox (1960)



Bluebox (1960)



Bluebox (1960)



Trygve Reenskaug

- **DBO, MVC, OO, UML, ect..**



Database Oriented Application (1965)

Front-end:

USERNAME:[PERSONA]

PASSWORD:[SECRET12]

Back-end:

Is access the account permitted when :

the username is 'PERSONA'

en

the password is 'SECRET12';

SQL injection (1997)

Front-end:

USERNAME:[PERSONA]

PASSWORD:[IDUNNO or 1=1]

Back-end:

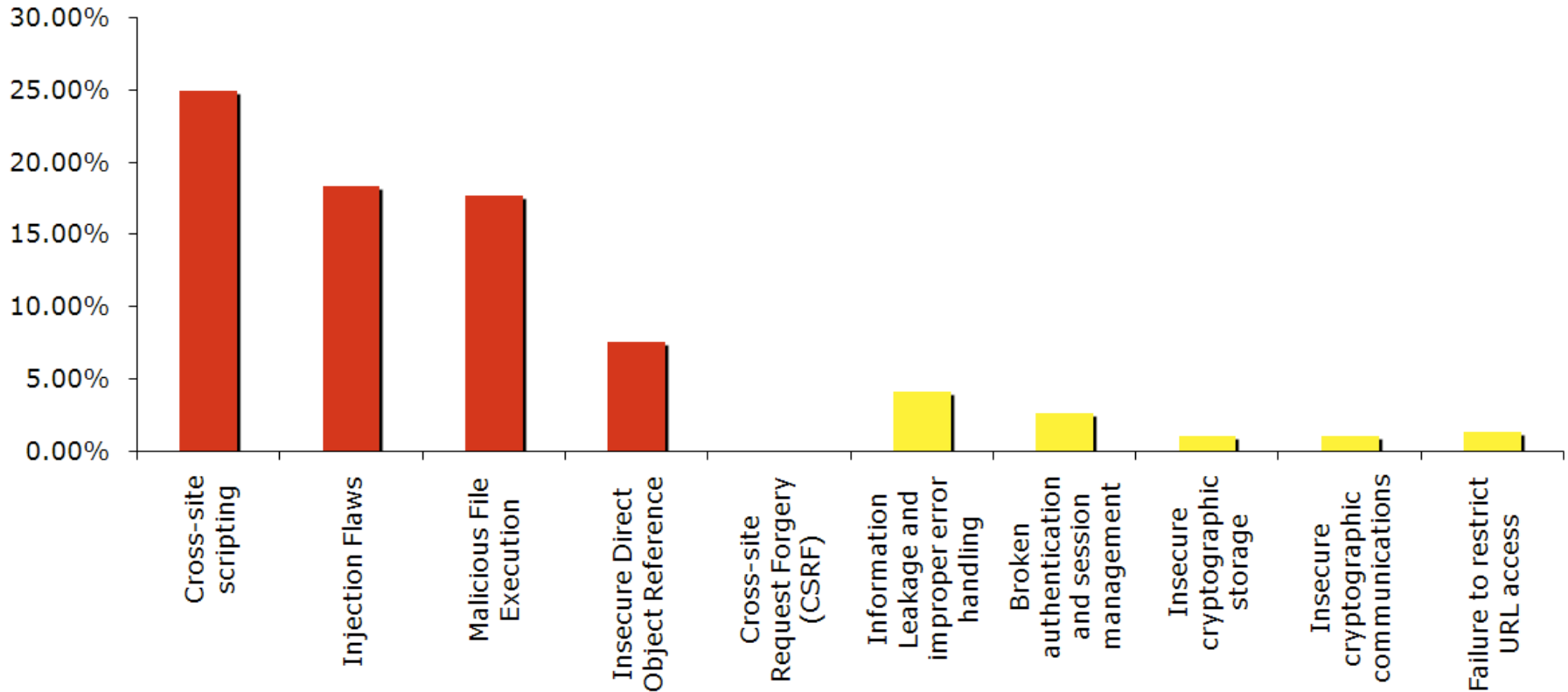
Is access to the account permitted when:

the username is 'PERSONA'

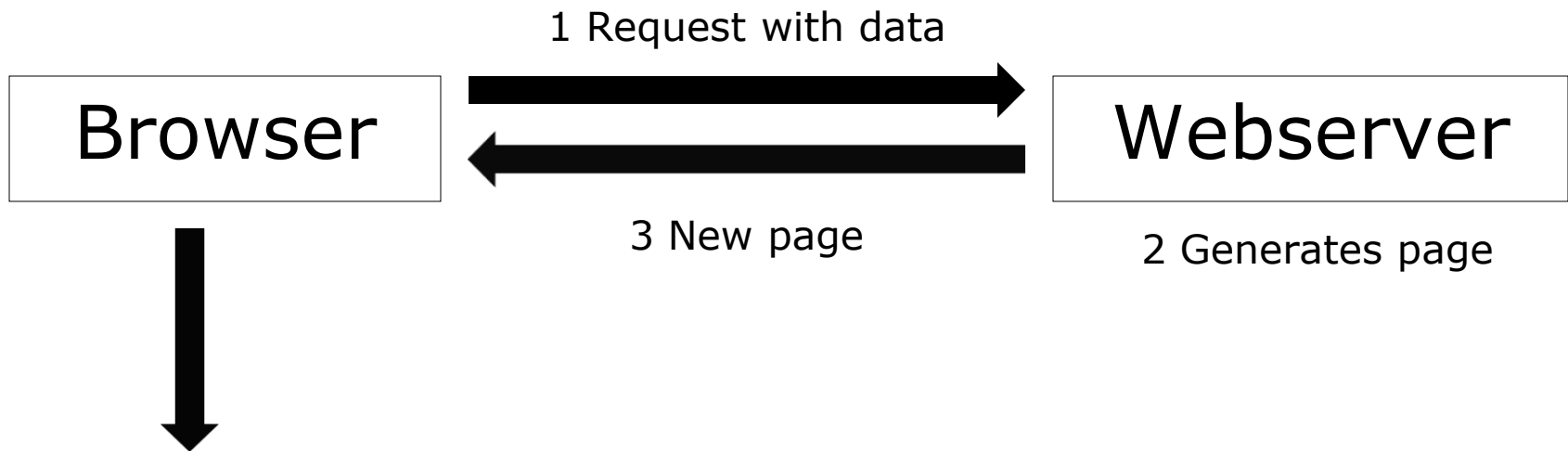
and

the password is IDUNNO or 1=1;

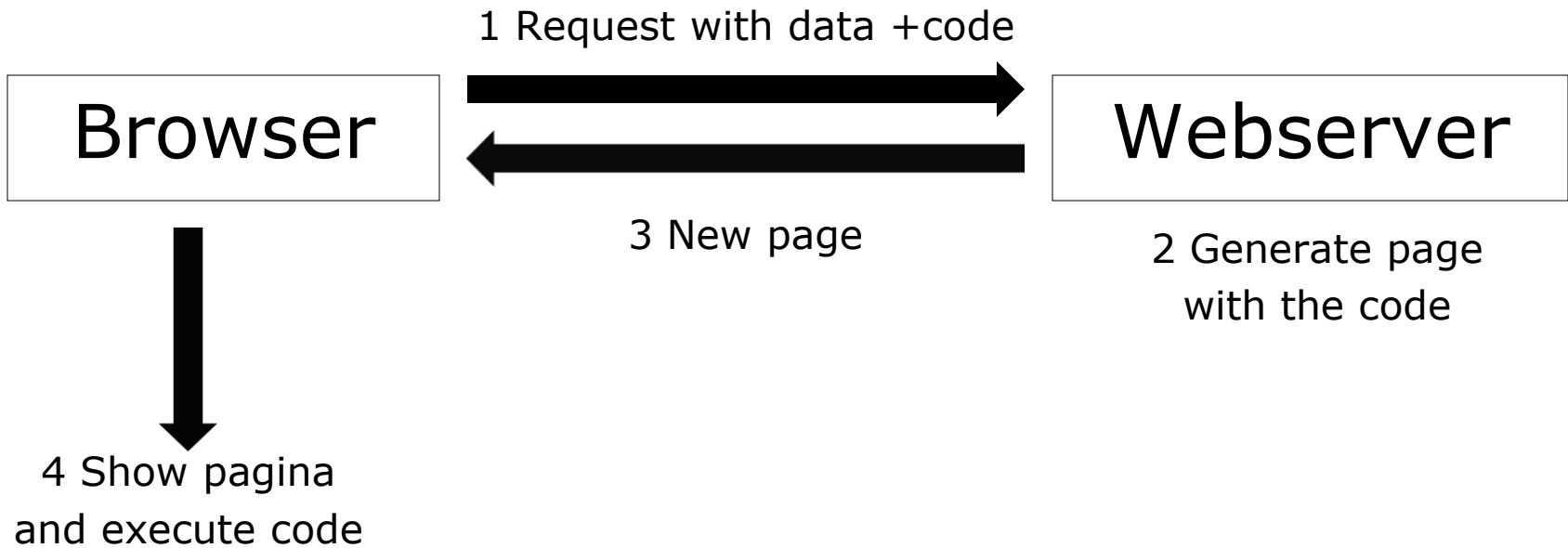
Attacks on webapplications



Dynamic HTML+Javascript (1995)



Cross-site scripting (1996)



Cause

Phone

Voice + Tones

Switch

Application

Query + Data

Database

Application

Content + Javascript

Browser

Cause

Phone

Voice + Tones

Switch

Application

Query + Data

Database

Application

Content + Javascript

Browser

Presentator

Text + Assignment

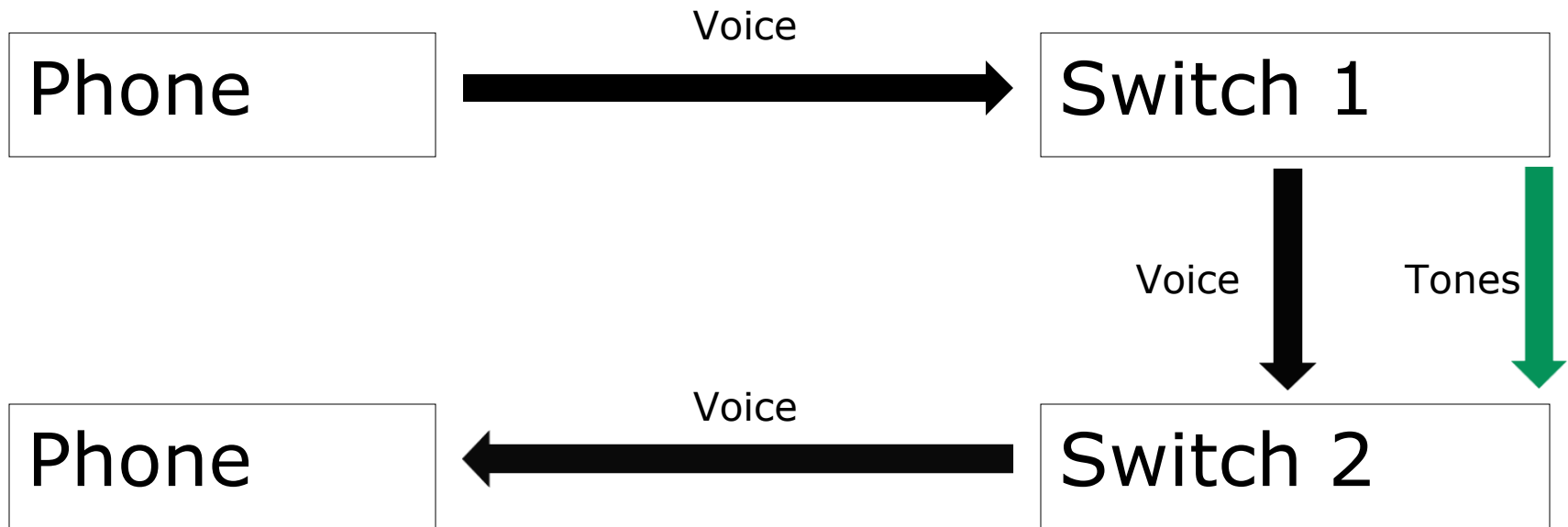
Audience

Root cause

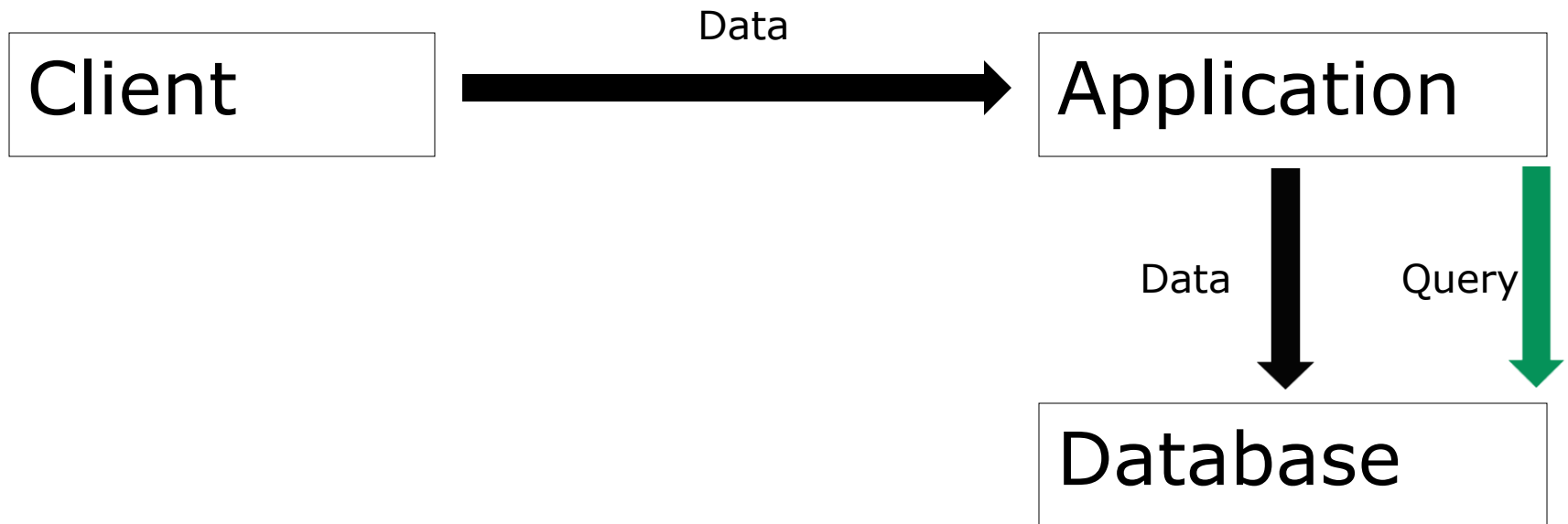


The interpreter can't distinguish between data and logic

Out of band communication - Telefoon



Out of band communication - Database



Out of band communication - Database

1:

```
stmt = 'select something from table where column = ' + variable;
```

2:

```
stmt = 'select something from table where column = ?';  
Bind(1,variable);
```

1:

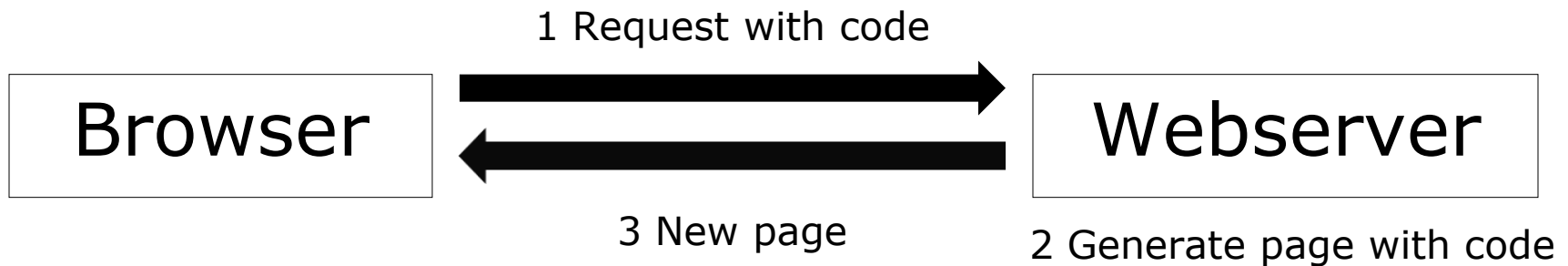
```
var: a, -> hash = 0cc175b9c0f1b6a831c399e269772661  
var: b, -> hash = 92eb5ffee6ae2fec3ad71c777531578f
```

2:

```
var: a, then hash = 4a8a08f09d37b73795649038408b5f33  
var: b, then hash = 4a8a08f09d37b73795649038408b5f33
```

Out of band communication - Browser

HTTP only has in band communication



Focus on incidents..

The screenshot shows the Tweakers.net website interface. At the top, there's a navigation bar with categories like FRONTPAGE, ARCHIEVEN, PRICEWATCH, FORUM, COMMUNITY, JOBS, and INLOGGEN. Below this is a secondary navigation bar with links for Archieven, Tags, Nieuws, Redactieblogs, Reviews (highlighted), Meuktracker, Plans, and Benchmarks. A search bar is on the right. On the left, there's a sidebar with menu items: Core, Pro, Games, Electronics, Mobile, Nieuws, and Vraag & Aanbod. The main content area features a Microsoft advertisement with the text "BUILT FOR THE FUTURE. READY NOW." and a man's face. Below the ad is the article title "Sql-injectie en xss: de beste verdediging" by Joost Schellevis, dated Monday, March 19, 2012. The article's sub-header is "Vertrouw nooit een gebruiker". The text discusses the importance of user trust and mentions a security researcher, Alexander Hoole, at the RSA Conference. A code block shows a PHP snippet for user authentication using `mysql_real_escape_string`. To the right of the article is a large grey box with the text "WHAT'S NEXT?". Below the article is a table of contents titled "Inhoudsopgave" with five items: 1. Inleiding, 2. Sql-injecties en xss, 3. Vertrouw nooit een gebruiker, 4. Voorbereiding, 5. Tot slot. At the bottom right, there are links for "Reviews I/O", "Reactie plaatsen", and "Printversie".

Core

Pro

Games

Electronics

Mobile

Nieuws

21:38 Besluit over standaardisering nan...

30-03 Betwiste Phone House-tablets ni...

30-03 Betalingsverwerker getroffen do...

Productreviews

03:11 Asus E35M1-M review door sdk1...

00:03 NGS Vip Wireless Mouse review ...

22:43 Fanatec Fanatec CSR Racing wh...

Vraag & Aanbod

09:10 A: Sigma Flash EF-500 DG Super...

09:09 A: Dell Ultrasharp 3008WFP € 650,-

08:59 A: Logitech Harmony 885 Advan...

Jobs

30-03 Sr.Technical Application Consulta...

30-03 Web Developer @ ClusterVision BV

30-03 Systeembeheerder met leidingge...

Etc.

07:53 Shopreview: MyCom

01:26 Shopreview: GigaVolt.nl

FRONTPAGE ARCHIEVEN PRICEWATCH FORUM COMMUNITY JOBS INLOGGEN

Archieven Tags Nieuws Redactieblogs **Reviews** Meuktracker Plans Benchmarks Zoeken...

Index » Reviews » Pro » Maatschappij » Privacy & beveiliging » Sql-injectie en xss: de beste verdediging » Vertrouw nooit een gebruiker hosted by TRU

BUILT FOR THE FUTURE. READY NOW. Microsoft

Sql-injectie en xss: de beste verdediging

Door Joost Schellevis, maandag 19 maart 2012 08:00, views: 150.582

Vertrouw nooit een gebruiker

"Het probleem is dat het al bij de start misgaat", zei HP-beveiligingsonderzoeker Alexander Hoole op de RSA Conference, een beveiligingsconferentie die eind februari in San Francisco van start ging. Hoewel sql-injectie vrij eenvoudig te voorkomen is, ontbreekt bij programmeurs toch vaak de benodigde kennis: "In leerboeken staat het al fout."

mysql_real_escape_string

De belangrijkste les is: wantrouw alle data die van de gebruiker afkomstig is. Het beste is om altijd aan te nemen dat een gebruiker te kwader trouw is - de kans dat een van de bezoekers dat ook daadwerkelijk is, is gezien het grote aantal datalekken redelijk groot.

Om te voorkomen dat MySQL-opdrachten in een variabele worden verstopt, kan in PHP de functie `mysql_real_escape_string` worden gebruikt. Die zorgt ervoor dat bepaalde karakters, waarmee het onderscheid tussen data en MySQL-code worden gemaakt, onschadelijk worden gemaakt. Dit noemt men 'escaping': voor een teken als een apostrof wordt dan een backslash geplaatst, zodat dat teken door MySQL als data wordt gezien, en niet als deel van de query.

```
$username = $_POST["username"];  
$password = crypt($_POST['password']);  
$q = "SELECT * FROM users WHERE  
username = ' . mysql_real_escape_string($username) . "'  
AND cryptePassword = ' . mysql_real_escape_string($password) . "'";  
$r = mysql_query($q);
```

WHAT'S NEXT?

Inhoudsopgave

1. Inleiding
2. Sql-injecties en xss
3. **Vertrouw nooit een gebruiker**
4. Voorbereiding
5. Tot slot

Reviews I/O

Reactie plaatsen Printversie

..specific a blacklist of incidenten

- **OWASP top 10**

- Injection
- Cross-Site Scripting (XSS)
- Broken Authentication and Session Management
- Insecure Direct Object References
- Cross-Site Request Forgery (CSRF)

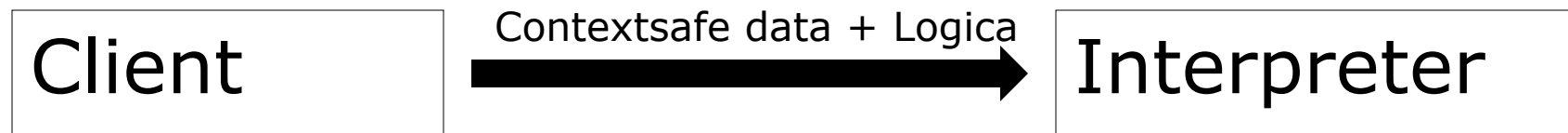
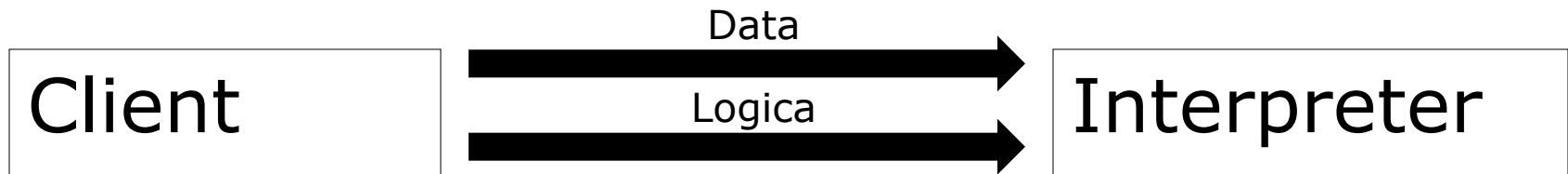
- **SANS CWE/25**

- Improper Neutralization of Special Elements used in an SQL Command ('SQL Injection')
- Improper Neutralization of Special Elements used in an OS Command ('OS Command Injection')
- Improper Neutralization of Input During Web Page Generation ('Cross-site Scripting')
- Unrestricted Upload of File with Dangerous Type
- Cross-Site Request Forgery (CSRF)
- URL Redirection to Untrusted Site ('Open Redirect')

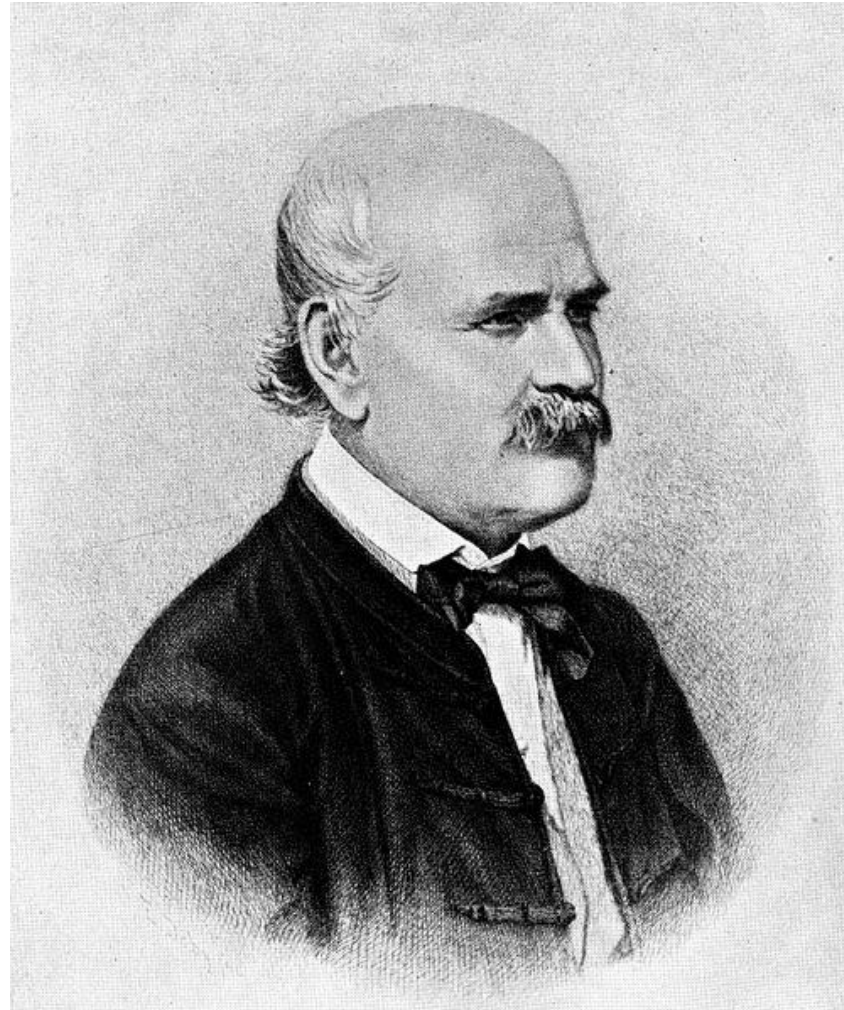
- **WASC 24+2**

- Cross-Site Scripting
- Cross-Site Request Forgery
- SSI Injection
- SQL Injection
- XPath Injection
- XQuery Injection

Principe



Ignaz Semmelweis



Listerine



Thus..

We have the right ideas

We know the solutions

We just use them in the wrong way

And present them in the wrong way

We can learn from history how to do this right

More on this stuff

C64:

Michael Steil / c64.org

Nate / root.org

XSS/SQL/Injection flaws:

www.owasp.org

Engressia / Draper / Semmelweis / Lister:

www.wikipedia.org

