AJAX Sikkerhetstesting

owasp medlemsmøte 27.08.08

Foredrag basert på Masteroppgave

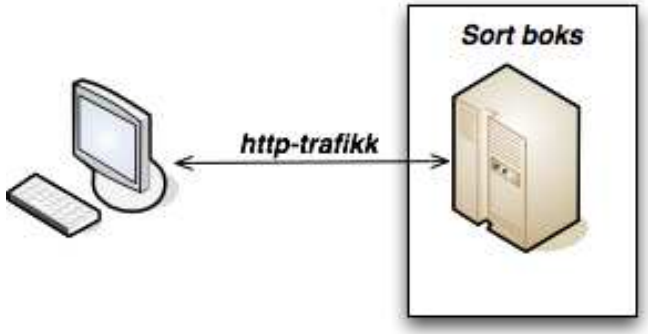
- *"Effektiv sikkerhetstesting av webapplikasjoner med rik klientkode"*

- NTNU våren 2008
- Veilder: Kåre Presttun, Mnemonic

Agenda

- Sikkerhetstesting
- AJAX sikkerhetstesting
  - problem identifisering
- Verktøy gjennomgang

“Effektiv sikkerhetstesting av webapplikasjoner med rik klientkode”



- Effektiv => Automatisert
- Rik klientkode = JavaScript
  - Ikke eksterne komponenter (Flash etc.)
- “Sort boks”-testing:
  - Mangler innsyn i kode på serversiden
  - Kun adresse: http://www.shop.com/

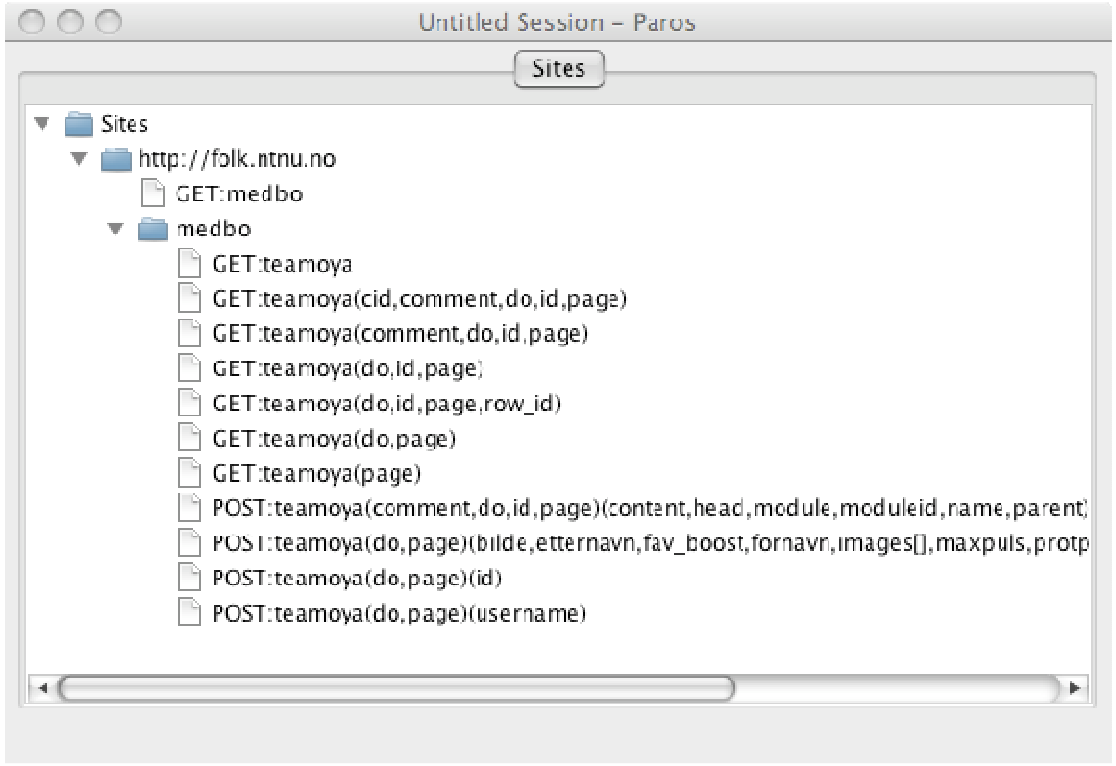
Automatisert sikkerhetstesting

- Tradisjonelle webapplikasjoner:

1. Protokolldrevet (tradisjonell) crawl
   - Interessante elementer: a, form, input
   - Gir oversikt over endepunktene på tjenersiden:
     - /shop
     - /pay?mode
     - …

Automatisert sikkerhetstesting

- Tradisjonelle webapplikasjoner:



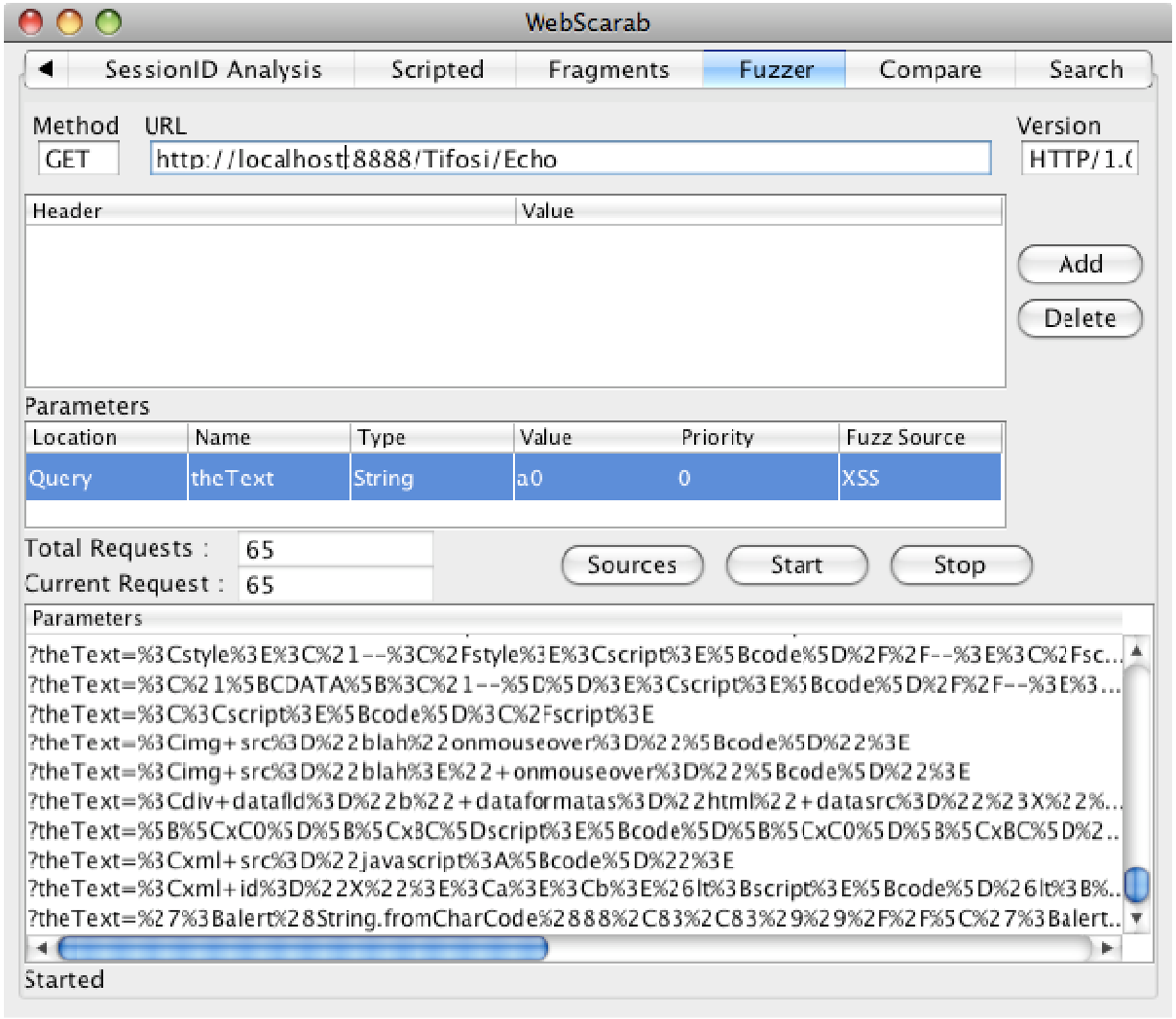
Automatisert sikkerhetstesting

- Tradisjonelle webapplikasjoner:

  2: Parameter-fuzzing
  - /shop?param1=%3CSCRIPT%3Ealert(%22Paros%22);%3C/SCRIPT%3E
  - /shop?param1=%27%27%3B%21%2D%2D%22%3C%58%53%53%3E
  - ...

  3: Analyse av svar fra server

Parameter-fuzzing i WebScarab



Automatisert sikkerhetstesting

- Webapplikasjoner med rike klienter:
  1. Protokolldrevet crawling
     - Fungerer ikke
  2. Parameter-fuzzing
     - Fungerer
  3. Analyse av svar fra server
     - Fungerer

Identifisering av problemer (1)

1. XHR-objekter bygger endepunkter dynamisk

AJAX:

```
var xmlhttp = new getXMLObject();

function ajaxFunctionRemoteCall() {
  if(xmlhttp) {
    xmlhttp.open("GET","AJAXEcho",true);
    xmlhttp.onreadystatechange  = handleServerResponse;
    xmlhttp.setRequestHeader('Content-Type', 'application/x-www-form-
urlencoded');
    xmlhttp.send("theText=buttonClicked");
  }
}
```

http://localhost/AJAXEcho?theText=buttonClicked

Identifisering av problemer (2)

1. Lenker er ikke entydig definert

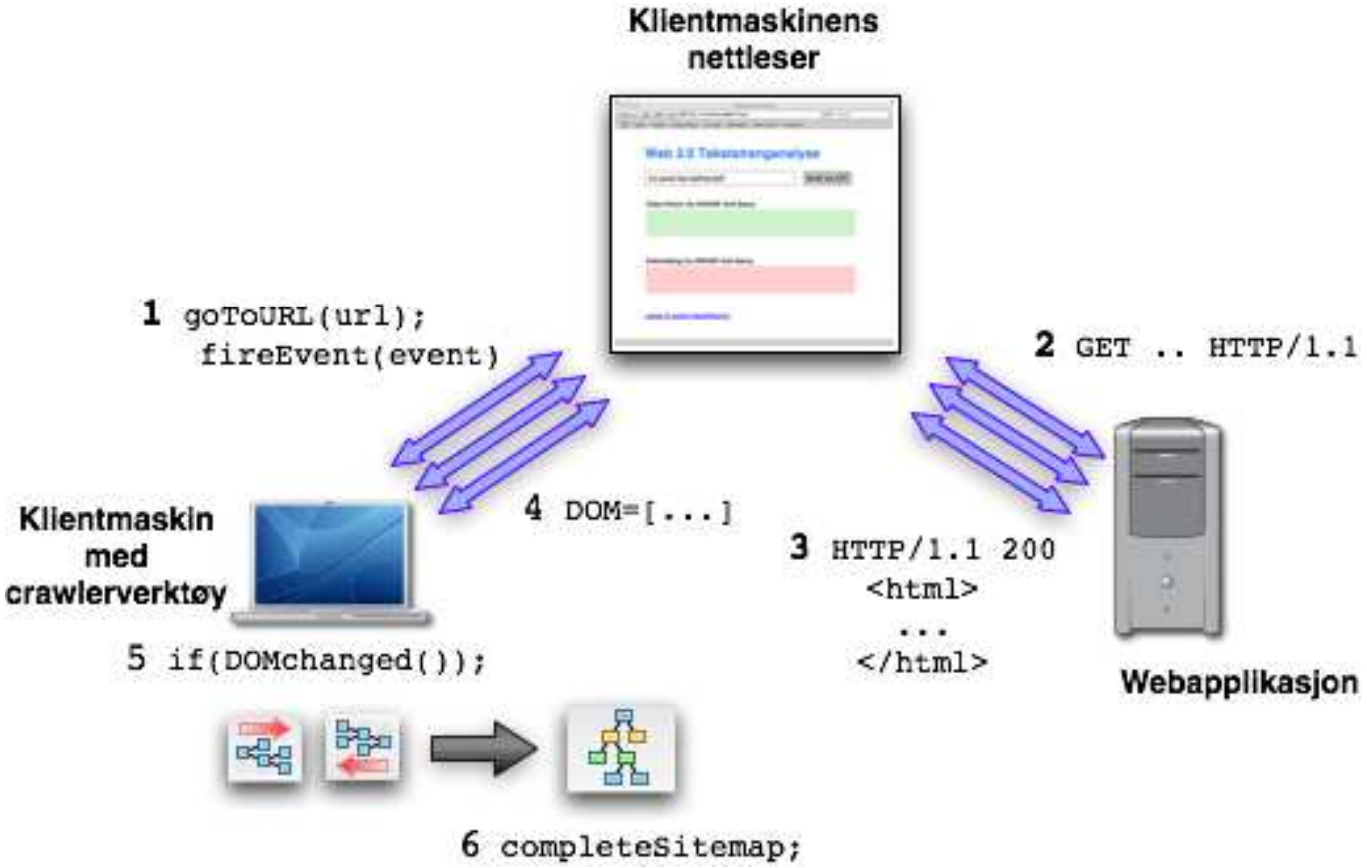
```
<a href=“#” onClick=“OpenNewsPage();” >
<div onClick=“OpenNewsPage();” >

<input type=“submit” class=“news” />
<div class=“news” >
<!– JQuery -->
$(".news").click(function(){
    $(“#content”).load(“news.html”);
});
```

Demo

Løsning: Event-drevet crawl

- Nettleseren blir brukt som et ledd i crawlingen
  - Egentlig: rendringsmotoren i nettleser

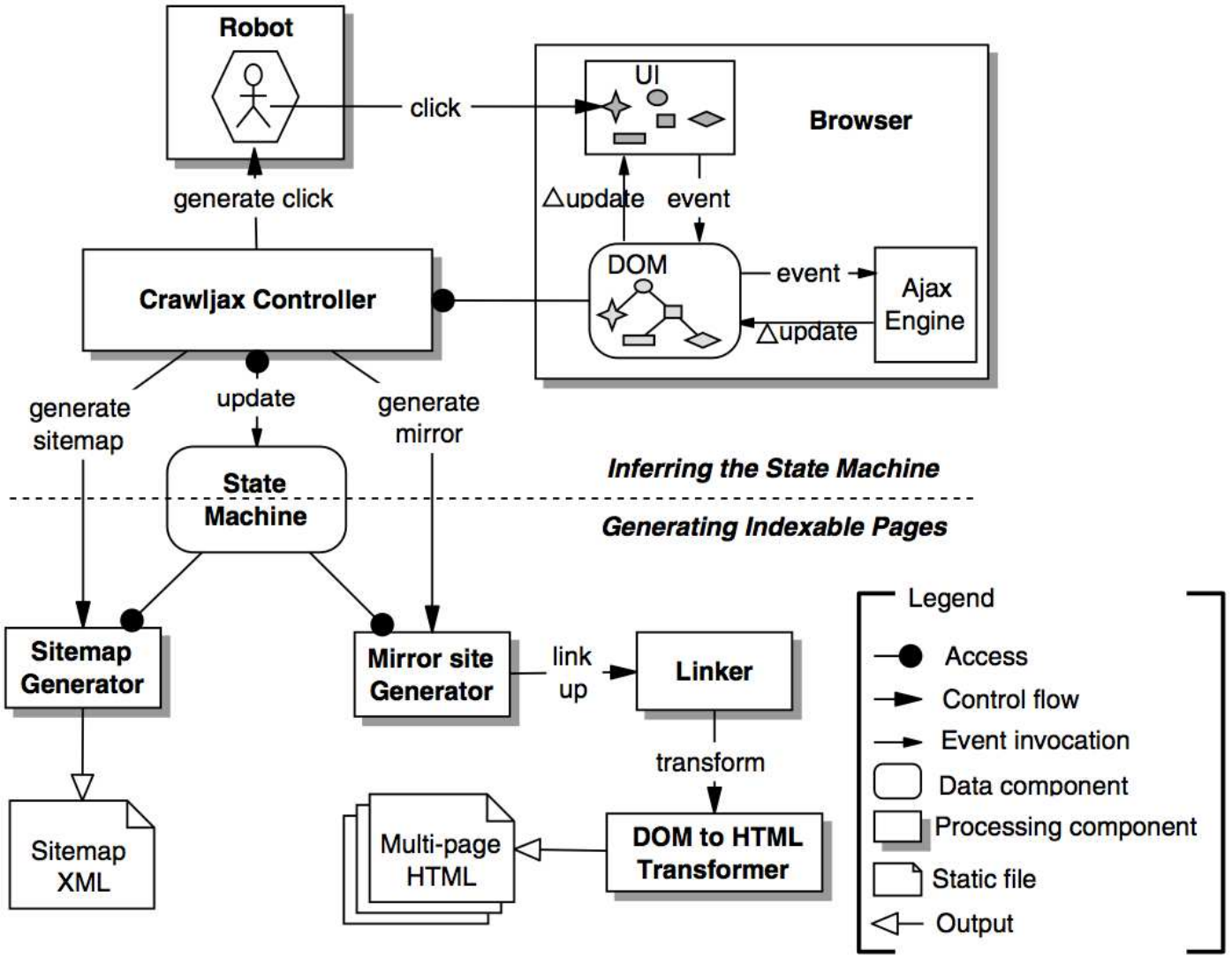
Event-drevet crawl



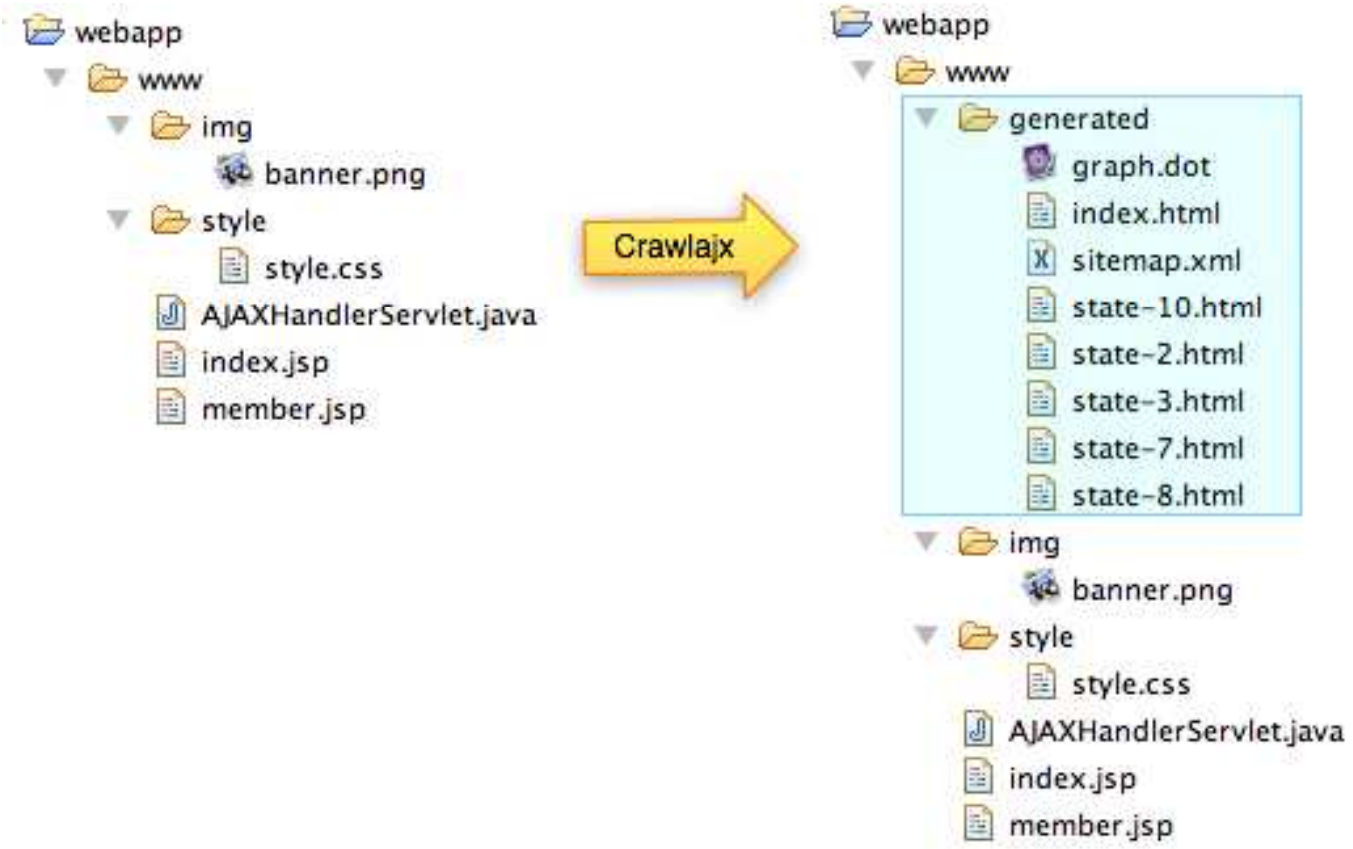
Crawljax (1)

- Hensikt:
  - ”*Gjøre innhold i AJAX-baserte webapplikasjoner indekserbart for søkemotorer*”
- Open source, Java
- Bygger på Watij m.fl.

Crawljax (2)


Robot
click
UI
Browser
generate click
△update
event
DOM
event
Ajax Engine
Crawljax Controller
△update
generate sitemap
update
generate mirror
State Machine
Inferring the State Machine
Generating Indexable Pages
Sitemap Generator
Mirror site Generator
link up
Linker
transform
Sitemap XML
Multi-page HTML
DOM to HTML Transformer
Legend
Access
Control flow
Event invocation
Data component
Processing component
Static file
Output

Crawljax (3)


webapp
www
img
banner.png
style
style.css
AJAXHandlerServlet.java
index.jsp
member.jsp
Crawlajx
webapp
www
generated
graph.dot
index.html
sitemap.xml
state-10.html
state-2.html
state-3.html
state-7.html
state-8.html
img
banner.png
style
style.css
AJAXHandlerServlet.java
index.jsp
member.jsp

Demo

Oppsummering

- Event-drevet crawl => avdekke flere endepunkter i AJAX-applikasjoner
- Crawljax
  - er ikke optimalisert for sikkerhetstesting, krever konfigurering
  - ”web-testing” er tidkrevende
  - Ikke bare sikkerhetstesting