Down the rabbit hole of

20-year long vulnerability

Olgierd Pieczul

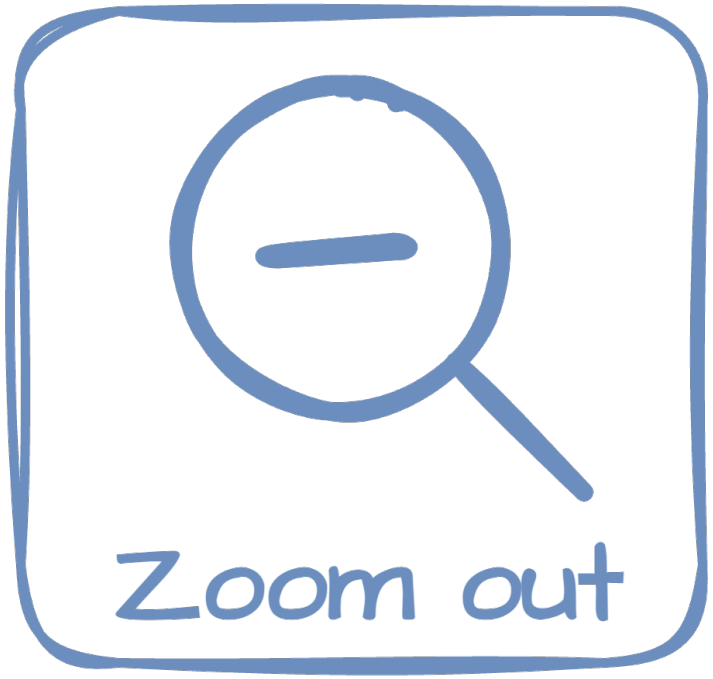
Security Architecture

Oracle Cloud Infrastructure

Disclaimer: I will lie (a bit)

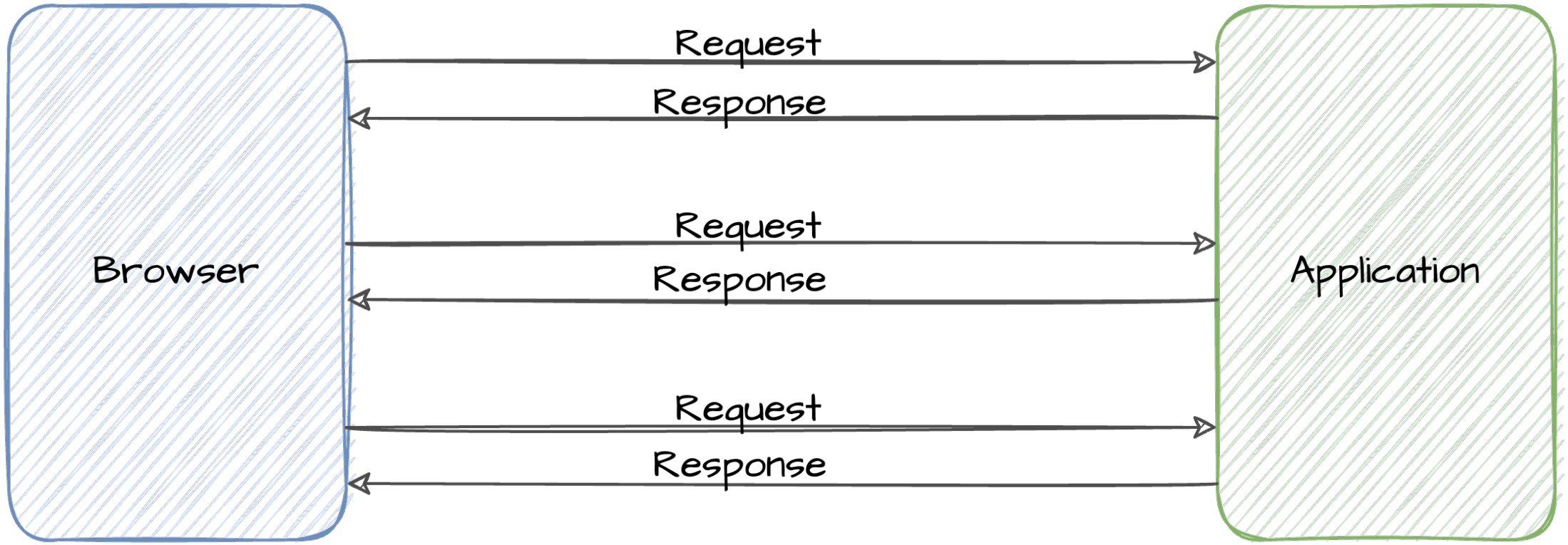
Flow

Web applications
Apache Struts
Handling parameters
Vulnerabilities



Technical challenges
Slippery slope
Clean abstractions
Vulnerability reports

Web applications


Browser
Request
Response
Request
Response
Request
Response
Application

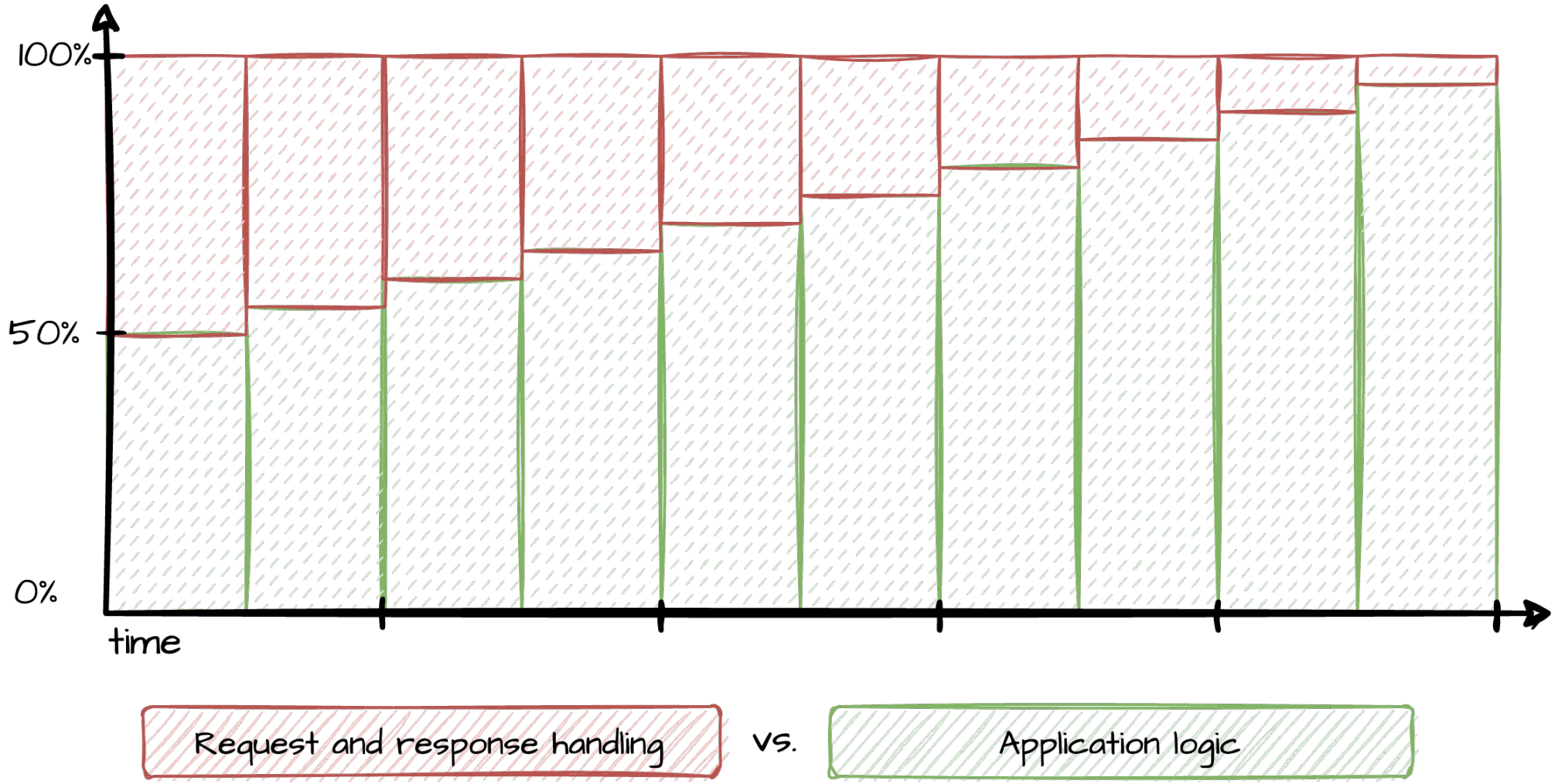
Zoom: 2x

Web applications, deeper

Humble beginnings: CGI/BIN

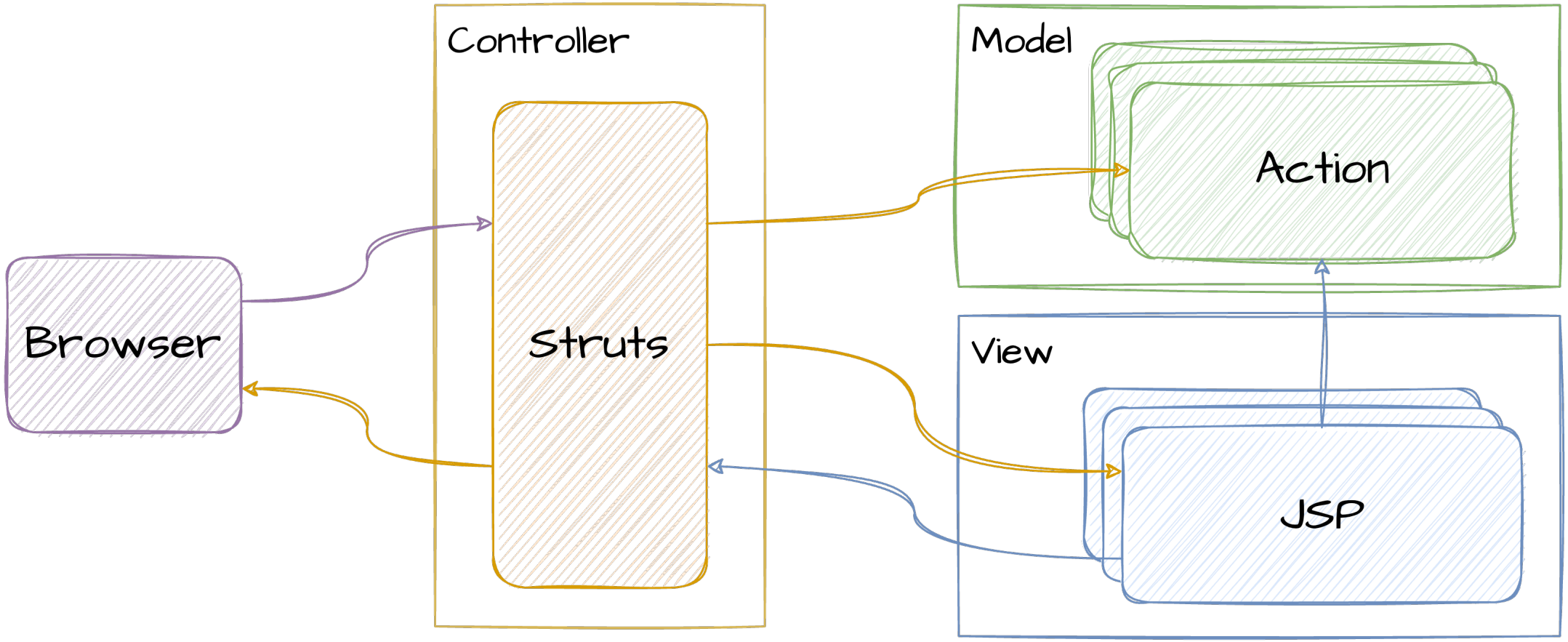
```
#!/usr/local/bin/perl
$in_string = <STDIN>; $in_string =~ tr/\+/ /s;
@params = split (/&/, $in_string); # tokenize

foreach $out_str (@params) {
  @pair = split (/=/, $out_str); # split
  $param{$pair[0]} = $pair[1];
}

foreach $key (keys %param) { #decode
  $key =~ s/%(..)/pack("c",hex($1))/ge;
  $param{$key} =~ s/%(..)/pack("c",hex($1))/ge;
}

print "200 ok\n";
print "content-type: text/html\n\n";
print "<html>\n";
```

Frameworks to the rescue



Apache Struts

MVC in Struts

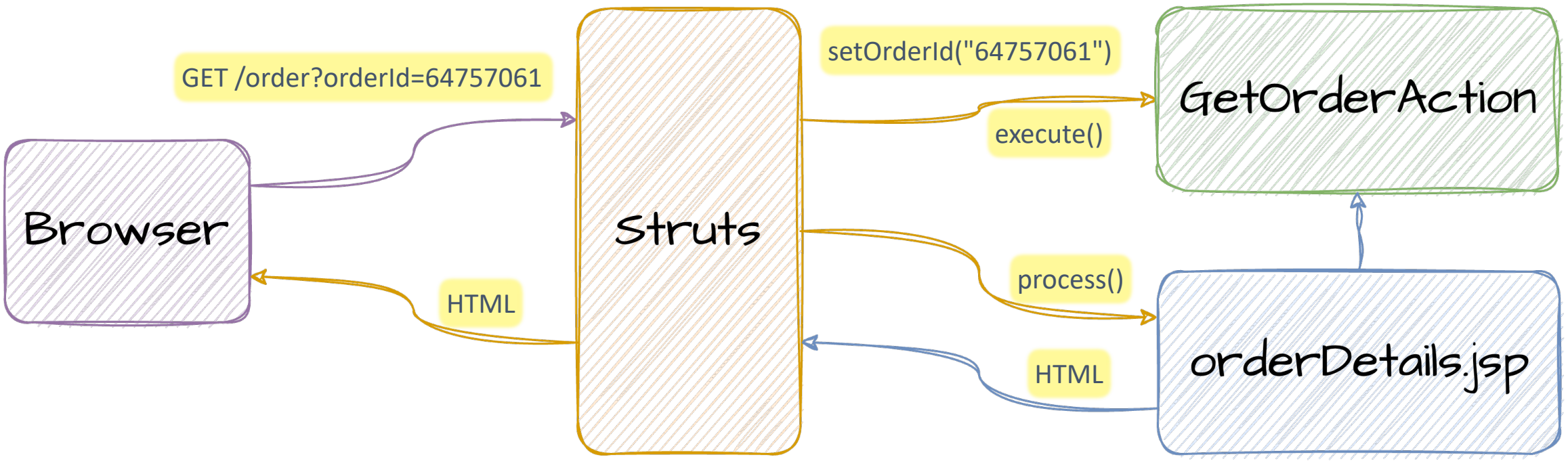


Zoom: 4x

Sample app

```
public class GetOrderAction {
  private String id;
  private Order order;

  public setOrderId(String id) {
    this.id = id
  }

  public String execute() {
    order = GetDB().getOrderById(id)
    if (order != null) return SUCCESS;
  }

  public Order getOrder() {
    return order;
  }
}
```

```
<html>
  <head>
    <title>Order details</title>
  </head>
  <body>
    <h1>
      Details of <s:property value="order.id" />
    </h1>
    <h2>
      Ordered at <s:property value="order.date" />
    </h2>
    <h2>
      Status <s:property value="order.status" />
    </h2>
    ...
  </body>
</html>
```




```
public setOrderId(String id) {
  this.id = id
}
```

```
public String execute() {
  order = GetDB().getOrderById(id)
  if (order != null) return SUCCESS;
}
```

```
public Order getOrder() {
  return order;
}
```

```
<body>
  <h1>
    Details of <s:property value="order.id" />
  </h1>
  <h2>
    Ordered at <s:property value="order.date" />
  </h2>
  <h2>
    Status <s:property value="order.status" />
  </h2>
  ...
</body>
```

Zoom: 5x

What is this **magic**?

Request

```
/order?orderId=64757061
```

Struts

Magic?

Action

```
action.setOrderId("64757061")
```

Zoom: 6x

OGNL: Object Graph Navigation Library

OGNL	Java
`foo`	`object.getFoo()`
`foo=bar`	`object.setFoo("bar")`
`list[2]`	`object.getList()[2]`
`foo.bar`	`object.getFoo().getBar()`
`foo.bar=7`	`object.getFoo().setBar(7)`

Zoom: 6x

Processing request with OGNL

Request

```
/order?orderId=64757061
```

OGNL

```
orderId; 64757061
```

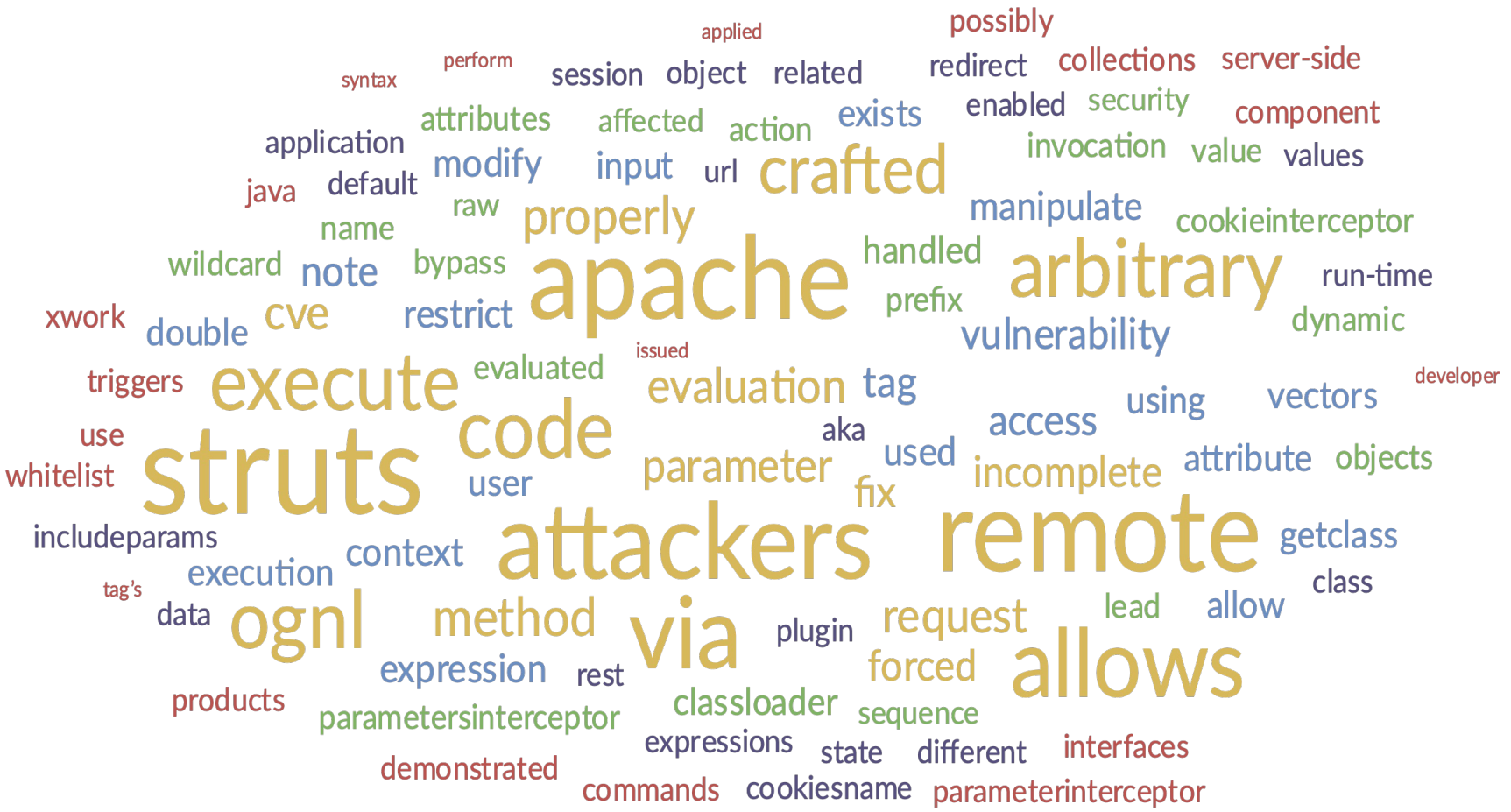
Action

```
action.setOrderId("64757061")
```

Vulnerabilities

CVE-2008-6504
CVE-2010-1870
CVE-2011-3923
CVE-2011-5057
CVE-2012-0392
CVE-2012-0393
CVE-2012-0838
CVE-2012-4387
CVE-2013-1965
CVE-2013-1966
CVE-2013-211
CVE-2013-213
CVE-2013-2135
CVE-2013-2251
CVE-2013
CVE-2
CVE-2014-0116
CVE-2016-0785
CVE-2016-3081
CVE-2016-3087
CVE-2016-3090
VE-2016-4438
VE-2016-4461
VE-2019-0230
CVE-2020-17530
CVE-2021-31805

Vulnerabilities



Tampering with OGNL

Zoom: 6x

Method execution

Request

```
/order?%23today%3Dnew%20java.util.Date%28%29
%2C%23today.toString%28%29%2CorderId=64757061
```

OGNL

```
#today=new java.util.Date(),
#today.toString(),
orderId; 64757061
```

Action

```
today = new java.util.Date()
today.toString()
action.setOrderId("64757061")
```

Zoom: 7x

Disable method execution

core/src/main/java/com/opensymphony/xwork2/interceptor/ParametersInterceptor.java

```
139                         ReflectionContextState.setCreatingNullObjects(contextMap, true);
140                         ReflectionContextState.setDenyMethodExecution(contextMap, true);
…
146                         ReflectionContextState.setCreatingNullObjects(contextMap, false);
147                         ReflectionContextState.setDenyMethodExecution(contextMap, false);
```

Java Showing the top two matches Last indexed on Sep 30, 2022

core/src/main/java/com/opensymphony/xwork2/util/reflection/ReflectionContextState.java

```
53                  return getBooleanProperty(GETTING_BY_KEY_PROPERTY, context);
54          }
55
56          public static void setDenyMethodExecution(Map<String, Object> context, boolean
    denyMethodExecution) {
```

Java Showing the top match Last indexed on Mar 24, 2021

Access session via context variable (no CVE)

Request

```
/order?%23session.user.admin=true
```

OGNL

```
#session.user.admin; true
```

Action

```
ActionContext.getSession().get("user").setAdmin(true)
```

Accepted parameters

```
if (name.indexOf('=') != -1 ||
    name.indexOf(',') != -1 ||
    name.indexOf('#') != -1) ...
```

Accepted parameters

```
if (name.indexOf('=') != -1 ||
    name.indexOf(',') != -1 ||
    name.indexOf('#') != -1 ||
    name.indexOf(':') != -1) ...
```

Accepted parameters

```
if (name.indexOf('=') != -1 ||
    name.indexOf(',') != -1 ||
    name.indexOf('#') != -1 ||
    name.indexOf(':') != -1 ||
    name.indexOf(\u0023) != -1) ...
```

Accepted parameters

```
/[\p{Graph}&&[^,#:=]]*/
```

Zoom: 6x

Context again (CVE-2010-1870)

Request

```
/order?%23context%5B%22xwork.MethodAccessor.denyMethodExecution%22%5D
%3Dtrue%40java.lang.Runtime%40getRuntime%28%29.exec%28%27mkdir%20%2F
tmp%2FPWNAGE%27%29%29%28meh%29&z%5B%28foo%29%28%27meh%27%29%5D=true
```

OGNL

```
(#context["xwork.MethodAccessor.denyMethodExecution"]=true
@java.lang.Runtime@getRuntime().exec('mkdir /tmp/PWNAGE'))(meh)
z[(foo)('meh')]; true
```

Action

```
context.set("xwork.MethodAccessor.denyMethodExecution", true)
java.lang.Runtime.getRuntime().exec('mkdir /tmp/PWNAGE')
```

Accepted parameters

```
/[a-zA-Z0-9\.\]\[\(\)_'\s]+/
```

Zoom: 6x

But wait, there's more!

WW-2160

```
@java.lang.System@exit(0).foo // Static methods
```

CVE-2012-0393

```
(new java.io.FileWriter('/etc/passwd')) // Public constructors
```

CVE-2013-2134

```
${#session.admin.true} // OGNL inside OGNL
```

Accepted parameters

```
/\w+((\.\w+)|(\[\d+\])|(\(\d+\))
|(\['\w+'\])|(\('\w+'\)))*/
```

Accepted parameters

```
/\w+((\.\w+)|(\[\d+\])|(\(\d+\))
|(\['\w+'\])|(\('\w+'\)))*[100]/
```

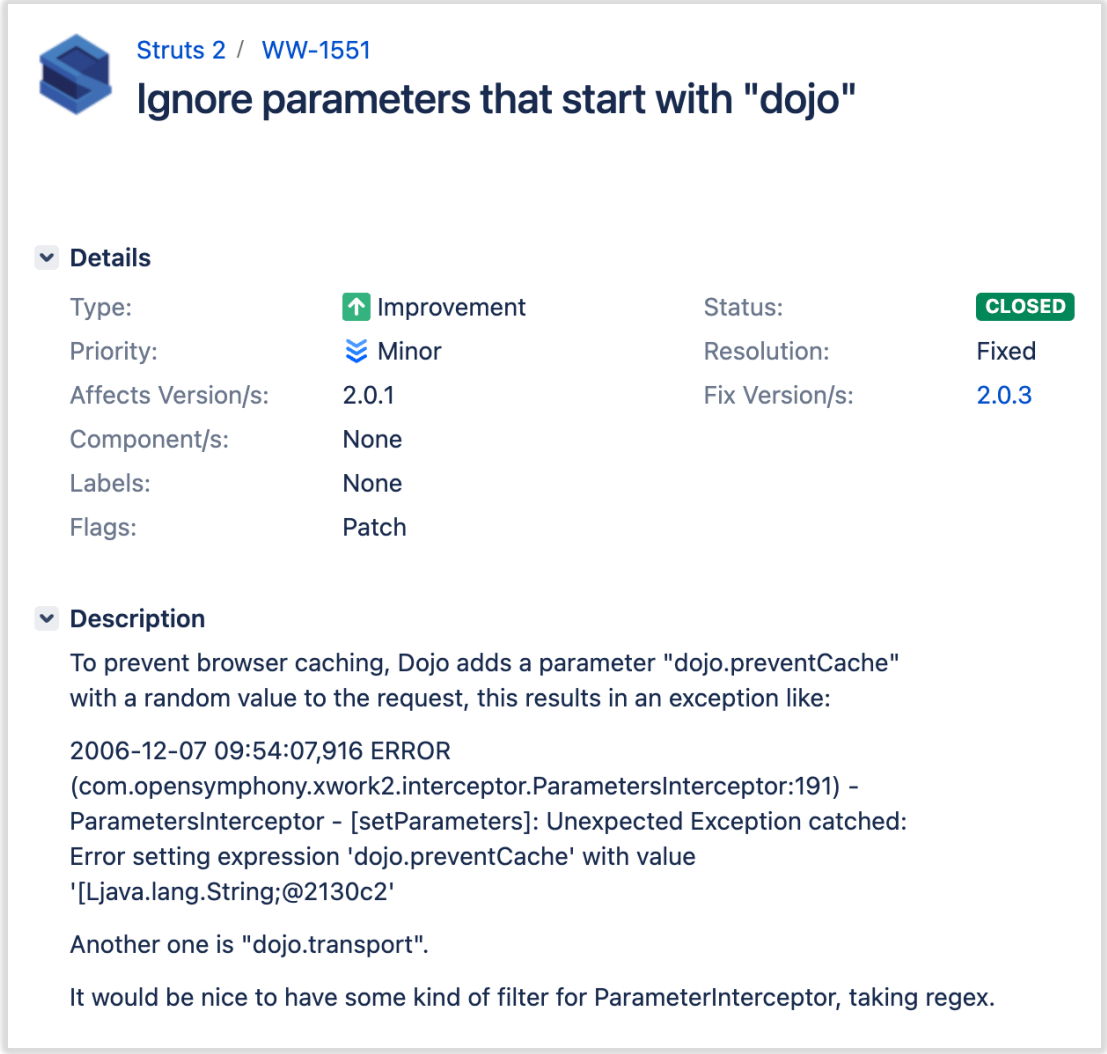
Accepted parameters

```
/\w+((\.\w+)|(\[\d+\])|(\(\d+\))|
(\['(\w|[\u4e00-\u9fa5])+'\])|
(\('(\w|[\u4e00-\u9fa5])+'\)))*/
```

Excluding patterns…

Zoom: 7x

Setting the stage



Struts 2 / WW-1551

Ignore parameters that start with "dojo"

Details

Type:	Improvement	Status:	CLOSED
Priority:	Minor	Resolution:	Fixed
Affects Version/s:	2.0.1	Fix Version/s:	2.0.3
Component/s:	None		
Labels:	None		
Flags:	Patch		

Description

To prevent browser caching, Dojo adds a parameter "dojo.preventCache" with a random value to the request, this results in an exception like:

2006-12-07 09:54:07,916 ERROR (com.opensymphony.xwork2.interceptor.ParametersInterceptor:191) - ParametersInterceptor - [setParameters]: Unexpected Exception catched: Error setting expression 'dojo.preventCache' with value '[Ljava.lang.String;@2130c2'

Another one is "dojo.transport".

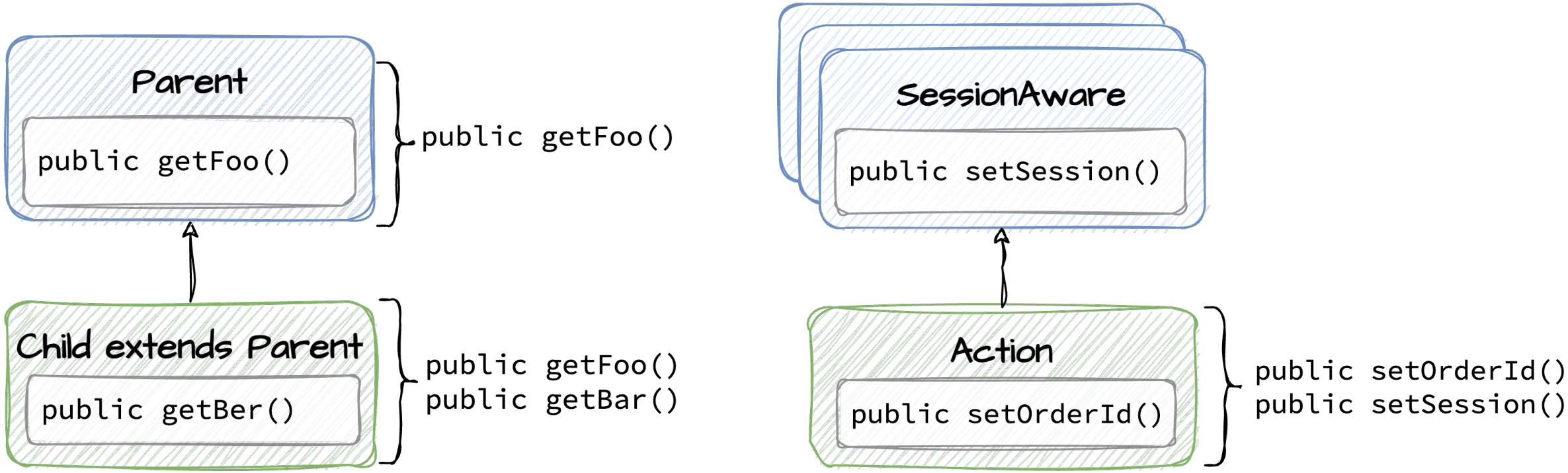
It would be nice to have some kind of filter for ParameterInterceptor, taking regex.

Exclude patterns

```
/dojo\..*/
/^struts\..*/
```

Zoom: 7x

Inheritance, competing features



Zoom: 7x

Session access, inheritance (CVE-2011-5057)

Request

```
/order?session.user.admin=true
```

OGNL

```
session.user.admin; true
```

Action

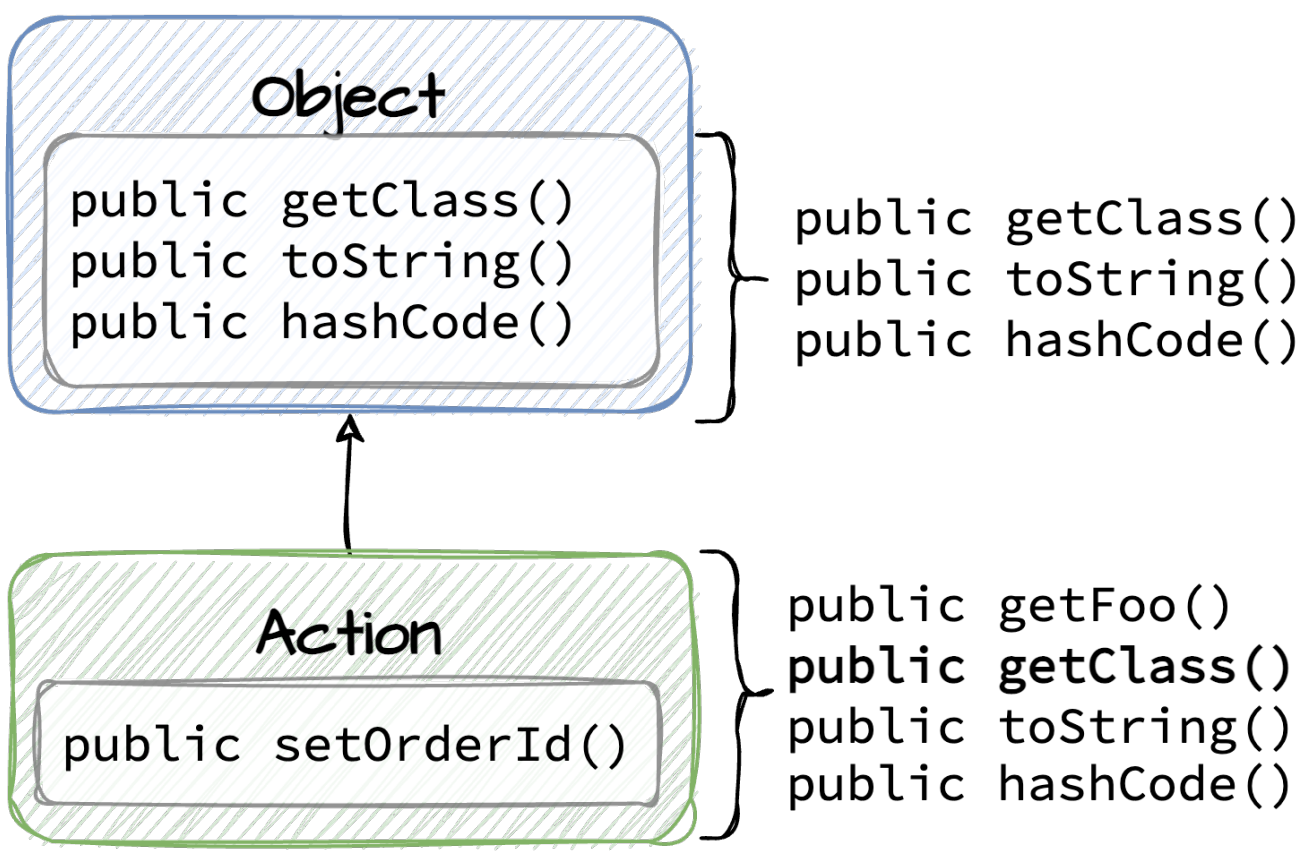
```
action.session.getUser().setAdmin(true)
```

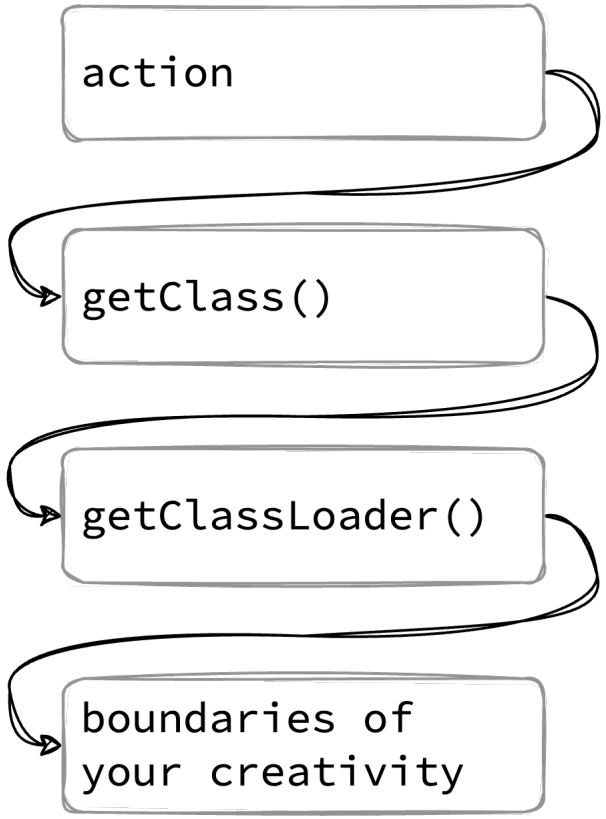
Exclude patterns to the rescue

```
/dojo\..*/
/^struts\..*/
/^session\..*/
/^request\..*/
/^application\..*/
/^servlet(Request|Response)\..*/
/parameters\...*/
```

Zoom: 8x

getClass (CVE-2014-0112)





Exclude patterns strikes back

```
/(.*\.|^|.*|\[('|"))(c|C)lass(\.|('|")]|\[).*/
/dojo\..*/
/^struts\..*/
/^session\..*/
/^request\..*/
/^application\..*/
/^servlet(Request|Response)\..*/
/parameters\...*/
```

Exclude patterns

```
/(.*\.|^|.*|\[('|"))\bclass(\.|('|")]|\[).*/
/(^|.*#)dojo(\.|\[).*/
/(^|.*#)struts(\.|\[).*/
/(^|.*#)session(\.|\[).*/
/(^|.*#)request(\.|\[).*/
/(^|.*#)application(\.|\[).*/
/(^|.*#)servlet(Request|Response)(\.|\[).*/
/(^|.*#)parameters(\.|\[).*/
/(^|.*#)context(\.|\[).*/
/(^|.*#)_memberAccess(\.|\[).*/
```

Exclude patterns

```
/(^|.*#)(dojo|struts|session
|request|application|servlet
(Request|Response)|parameters
|context|_memberAccess)(\.|\[).*"/
```

Exclude patterns

```
/(^|\%\{)((#?)(top(\.|\['|\["))|\[\d\]\.)?)
(dojo|struts|session|request|response|appli
cation|servlet(Request|Response|Context)|
parameters|context|_memberAccess)(\.|\[).*/
```

Are we done?

org.apache.struts2.interceptor

Class CookieInterceptor

java.lang.Object
    com.opensymphony.xwork2.interceptor.AbstractInterceptor
        org.apache.struts2.interceptor.CookieInterceptor

All Implemented Interfaces:

`Interceptor, Serializable`

```
public class CookieInterceptor
extends AbstractInterceptor
```

The aim of this interceptor is to set values in the stack/action based on cookie name/value of interest.

Exclude Patterns: CookieInterceptor

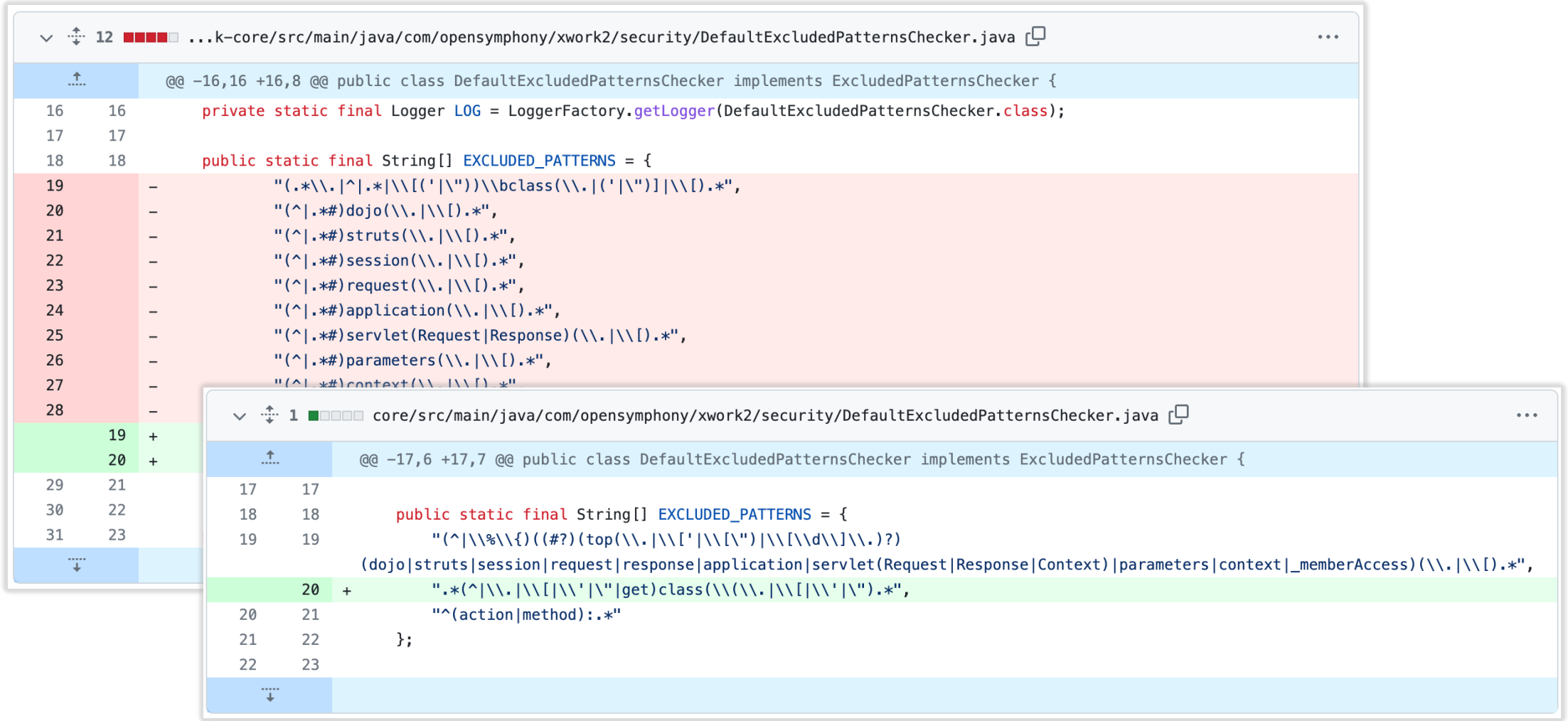
```
/[a-zA-Z0-9\.\]\[\(\)_'\s]+/
```

What went wrong?

And what we can learn?

Zoom: 4x

String matching and regexps



12 ...k-core/src/main/java/com/opensymphony/xwork2/security/DefaultExcludedPatternsChecker.java

```
@@ -16,16 +16,8 @@ public class DefaultExcludedPatternsChecker implements ExcludedPatternsChecker {
16 16      private static final Logger LOG = LoggerFactory.getLogger(DefaultExcludedPatternsChecker.class);
17 17
18 18      public static final String[] EXCLUDED_PATTERNS = {
19    -             "(.*\\.|^|.*|\\[('|\"))\\bclass(\\.|('|\")]|\\[).*",
20    -             "(^|.*#)dojo(\\.|\\[).*",
21    -             "(^|.*#)struts(\\.|\\[).*",
22    -             "(^|.*#)session(\\.|\\[).*",
23    -             "(^|.*#)request(\\.|\\[).*",
24    -             "(^|.*#)application(\\.|\\[).*",
25    -             "(^|.*#)servlet(Request|Response)(\\.|\\[).*",
26    -             "(^|.*#)parameters(\\.|\\[).*",
27    -             "(^|.*#)context(\\.|\\[).*"
28    -         };
19 +         "(^|\\%\\{)((#?)(top(\\.|\\['|\\[\")|\\[\\d\\]\\.)?)(dojo|struts|session|request|response|application|servlet(Request|Response|Context)|parameters|context|_memberAccess)(\\.|\\[).*",
20 +         ".*(^|\\.|\\[|\\'|\"|get)class(\\(\\.|\\[|\\'|\").*",
21 21          "^(action|method):.*"
22 22      };
23 23
```

1 core/src/main/java/com/opensymphony/xwork2/security/DefaultExcludedPatternsChecker.java

```
@@ -17,6 +17,7 @@ public class DefaultExcludedPatternsChecker implements ExcludedPatternsChecker {
17 17
18 18      public static final String[] EXCLUDED_PATTERNS = {
19 19          "(^|\\%\\{)((#?)(top(\\.|\\['|\\[\")|\\[\\d\\]\\.)?)(dojo|struts|session|request|response|application|servlet(Request|Response|Context)|parameters|context|_memberAccess)(\\.|\\[).*",
20 +         ".*(^|\\.|\\[|\\'|\"|get)class(\\(\\.|\\[|\\'|\").*",
21 20          "^(action|method):.*"
22 21      };
23 22
```

Slippery slope

Date	Accepted parameters	Accepted cookies	Excluded patterns
Jan 2004	[empty]	n/a	n/a
Dec 2004	excluded: '=', ',' ,'#'	n/a	n/a
Feb 2007	excluded: '=', ',' ,'#', **':'**	n/a	**dojo\..***
Jul 2008	'=', ',' ,'#', ':', **\u0023**	[empty]	dojo\..*
Oct 2008	**[\p{Graph}&&[^,#:=]]***	[empty]	dojo\..*
Aug 2010	**[a-zA-Z0-9\.\]\[\(\)_'\s]+**	[empty]	dojo\..*, ^struts\..*
Dec 2011	[a-zA-Z0-9\.\]\[\(\)_'\s]+	**[a-zA-Z0-9\.\]\[_'\s]+**	dojo\..*, ^struts\..*
Dec 2011	**[a-zA-Z0-9\.\]\[\(\)_']+**	[a-zA-Z0-9\.\]\[_'\s]+	dojo\..*, ^struts\..*
Jan 2012	**\w+((\.\w+)|(\[\d+\])|(\(\d+\))|(\['\w+'\])|(\('\w+'\)))***	[a-zA-Z0-9\.\]\[_'\s]+	dojo\..*, ^struts\..*
Apr 2012	\w+((\.\w+)|(\[\d+\])|(\(\d+\))|(\['\w+'\])|(\('\w+'\)))*	[a-zA-Z0-9\.\]\[_'\s]+	dojo\..*, ^struts\..***, ^session\..*, ^request\..*, ^application\.., ^servlet(Request|Response)\..*, parameters\...***
Aug 2012	\w+((\.\w+)|(\[\d+\])|(\(\d+\))|(\['\w+'\])|(\('\w+'\)))* **[100]**	[a-zA-Z0-9\.\]\[_'\s]+	dojo\..*, ^struts\..*, ^session\..*, ^request\..*, ^application\..*, ^servlet(Request|Response)\..*, parameters\...*
Mar 2014	\w+((\.\w+)|(\[\d+\])|(\(\d+\))|(\['\w+'\])|(\('\w+'\)))* [100]	[a-zA-Z0-9\.\]\[_'\s]+	**^class\..***, ^dojo\..*, ^struts\..*, ^session\..*, ^request\..*, ^application\..*, ^servlet(Rrequest|Response)\..*, ^parameters\..*, ^action:.*, ^method:.*
Apr 2014	\w+((\.\w+)|(\[\d+\])|(\(\d+\))|(\['\w+'\])|(\('\w+'\)))* [100]	[a-zA-Z0-9\.\]\[_'\s]+	**(.*\.|^|.*|\[('|"))(c|C)lass(\.|('|")]|\[\).***, ^dojo\..*, ^struts\..*, ^session\..*, ^request\..*, ^application\..*, ^servlet(Request|Response)\..*, ^parameters\..*
Apr 2014	\w+((\.\w+)|(\[\d+\])|(\(\d+\))|(\['\w+'\])|(\('\w+'\)))* [100]	[a-zA-Z0-9\.\]\[_'\s]+	*default*: **(.*\.|^|.*|\[('|"))class(\.|('|")]|\[\).*; params:** ^dojo\..*, ^struts\..*, ^session\..*, ^request\..*, ^application\..*, ^servlet(Request|Response)\..*, ^parameters\..*
May 2014	\w+((\.\w+)|(\[\d+\])|(\(\d+\))|(\['\w+'\])|(\('\w+'\)))* [100]	[a-zA-Z0-9\.\]\[_'\s]+	*default:* **(.*\.|^|.*|\[('|"))class(\.|('|")]|\[\).*, ^dojo\..*, ^struts\..*, ^session\..*, ^request\..*, ^application\..*, ^servlet(Request|Response)\..*, ^parameters\..*,** *params:* ^action:.*, ^method:.*
Dec 2014	**\w+((\.\w+)|(\[\d+\])|(\(\d+\))|(\['(\w|[\u4e00-\u9fa5])+'\])|(\('(\w|[\u4e00-\u9fa5])+'\)))***	[a-zA-Z0-9\.\]\[_'\s]+	*default:* **(.*\.|^|.*|\[('|"))\bclass(\.|('|")]|\[\).*, (^|.*#),dojo(\.|\[\).*, (^|.*#)struts(\.|\[\).*, (^|.*#)session(\.|\[\).*, (^|.*#)request(\.|\[\).*, (^|.*#)application(\.|\[\).*, (^|.*#)servlet (Request|Response)(\.|\[\).*, (^|.*#)parameters(\.|\[\).*, (^|.*#)context(\.|\[\).*, (^|.*#)_memberAccess(\.|\[\).*** *params:* ^action:.*,^method:.*
May 2015	\w+((\.\w+)|(\[\d+\])|(\(\d+\))|(\['(\w|[\u4e00-\u9fa5])+'\])|(\('(\w|[\u4e00-\u9fa5])+'\)))*	[a-zA-Z0-9\.\]\[_'\s]+	*default:* **(^|.*#)(dojo|struts|session|request|application|servlet(Request|Response)|parameters|context|_memberAccess)(\.|\[\).*,** *params:* ^(action|method):.*
Sep 2015	\w+((\.\w+)|(\[\d+\])|(\(\d+\))|(\['(\w|[\u4e00-\u9fa5])+'\])|(\('(\w|[\u4e00-\u9fa5])+'\)))*	[a-zA-Z0-9\.\]\[_'\s]+	*default:* **(^|\%\{((#?)(top(\.|\['|\["))|\[\d\]\.)?)**(dojo|struts|session|request|response|application|servlet(Request|Response|Context)|parameters|context|_memberAccess)(\.|\[).*, ^(action|method):.*

Unexpected use of existing conventions

Java field access modifiers

Plain Java objects as actions

Parameters set by matching fields

Any public getter/setter accessible by HTTP request

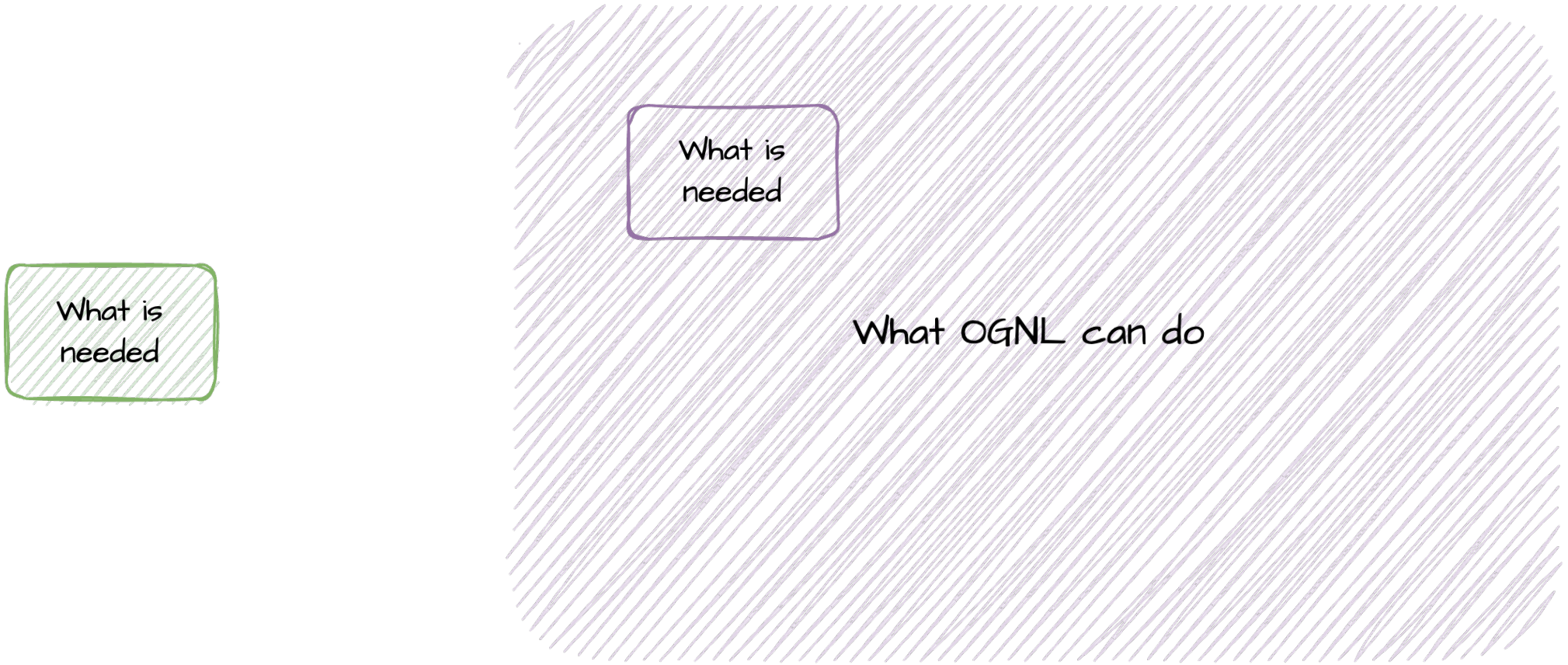
A better way

```
@Controller
class Orders {

  @GetMapping("/order")
  public Order getOrder(@RequestParam String orderId) {
    ...
  }
```

Struts Jira, 19/October/2007: *"A new feature we could add would be a new annotation so that a user could annotate which setters/getters can be accessed, which is probably a good idea regardless".*

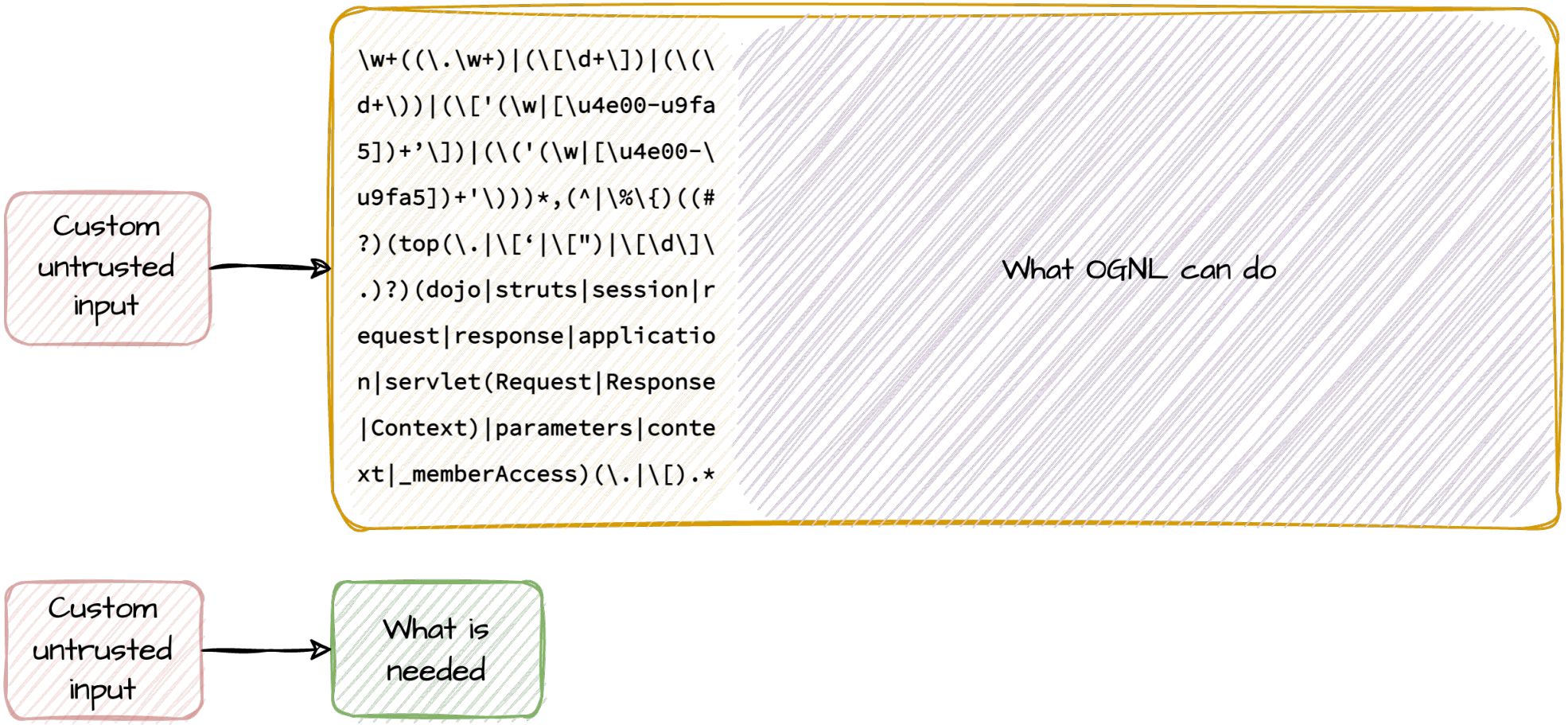
Zoom: 0.7x

Dangers of powerful tools...



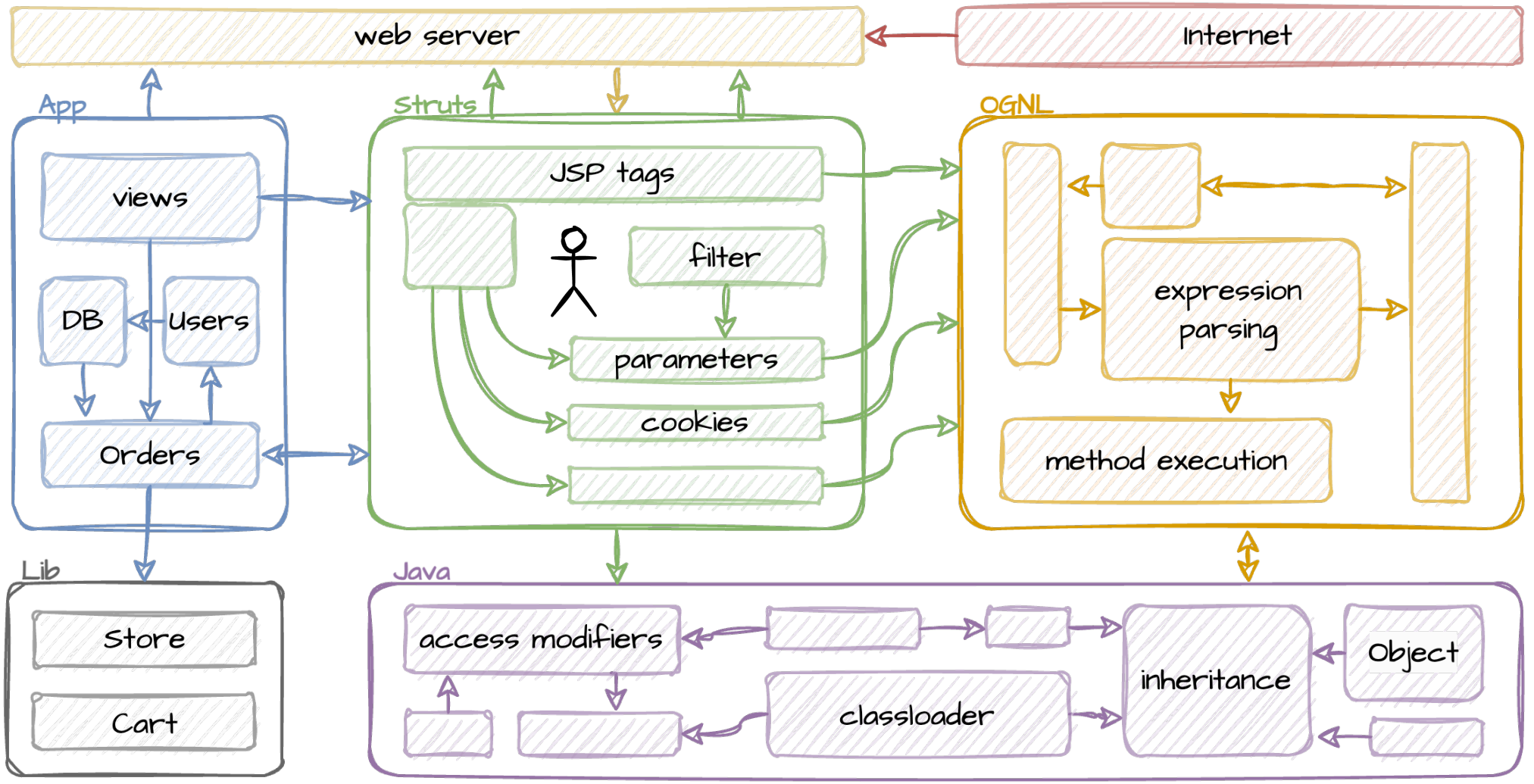
Zoom: 0.7x

… and attempts to contain them



Zoom: 0.5x

Complex interactions



Bug reports may mislead you

Source	Details
Report	Arbitrary File Overwrite in Struts <= 2.3.1 (ParametersInterceptor) Given Test.java has an uninitialized property "name" of type String: String name; /The following request will create/overwrite the file "C:/sec-consult.txt"
CVE-2012-0393	The ParameterInterceptor component in Apache Struts before 2.3.1.1 does not prevent access to public constructors, which allows remote attackers to create or overwrite arbitrary files via a crafted parameter that triggers the creation of a Java object.
Struts Advisory	Arbitrary File Overwrite in Struts <= 2.3.1 (ParameterInterceptor) While accessing the flag allowStaticMethodAccess within parameters is prohibited since Struts 2.2.3.1 an attacker can still access public constructors with only one parameter of type String.

So may vulnerability records

CVE	Details	Score
CVE-2008-6504	ParametersInterceptor [...] does not properly restrict # (pound sign) references to context objects, which allows remote attackers to execute OGNL statements and **modify server-side context objects**, as demonstrated by use of a \u0023 representation for the # character.	5.0
CVE-2010-1870	The OGNL extensive expression evaluation [...] uses a permissive whitelist, which allows remote attackers to **modify server-side context objects** and bypass the "#" protection mechanism in ParameterInterceptors via the (1) #context, (2) #_memberAccess, (3) #root, (4) #this, [...]	5.0
CVE-2011-3923	Apache Struts before 2.3.1.2 allows remote attackers to bypass security protections in the ParameterInterceptor class and **execute arbitrary commands**.	9.8

Zoom: 0.2x

Assumptions about others

“We can't do enough to emphasize the importance of being careful what values you expose via public methods on an action.”

“If the action was really worried about security, it would implement the ParameterNameAware interface.”

Closing highlights

- Struts legacy
- No magic
  - Clean abstraction
  - Understanding
- Powerful tools
  - No way to contain
  - Strings
- Design change timing
- Bug reports
  - May be misleading
  - Or inaccurate

Pieczul, Foley. “The evolution of a security control”, In Security Protocols XXIV

Thank you!

Questions?