

开源大数据平台的安全实践

刘杰
百度(中国)有限公司, 北京 100085

摘要

开源大数据平台的安全机制目前并不特别完善, 特别是用户认证、日志审计等方案还存在不少问题。分析了开源大数据平台存在的安全隐患, 在开源Hadoop平台的基础上提出了基于Giahoo的大数据平台用户认证机制和基于高效的日志审计方案, 并结合百度公司的实际应用阐述所提技术方案的有效性。该技术方案具有一定的普适性, 可根据企业当前技术环境灵活使用。

关键词

大数据安全; 用户认证; 日志审计; 数字安全

中图分类号: TP309 文献标识码: A doi: 10.11959/j.issn.2096-0271.2017050

Practice on security of big data platform

LIU Jie
Baidu China Co., Ltd., Beijing 100085, China

Abstract

The security mechanism of open source platform is not sophisticated, especially on user authentication and log audit. The potential safety problems of open source platform were analyzed. Based on the open source Hadoop platform, the user authentication based on Giano, and the efficient log audit based on rules were proposed. Combined with the practical application of Baidu, the effectiveness of the technical scheme was expounded. The technical scheme has certain universality, and it can be flexibly used according to the current technological environment of the enterprise.

Key words

big data security, user authentication, log audit, digit security

1 引言

伴随大数据技术的发展,数据资源逐渐成为最重要的资源。国家“十三五”规划纲要中指出,实施国家大数据战略,把大数据作为基础性战略资源,促进大数据发展。世界已经进入由数据主导的“大时代”^①。大数据是一场革命,将改变人们的工作、生活和思维方式。但是大数据是一把双刃剑,使人们在享受大数据带来的巨大价值的同时,也带来了大数据的安全问题。大数据在数据采集、数据处理、数据存储和数据内容安全等方面均存在着极大的风险。

Hadoop等开源软件提供了适用于数据采集、数据处理、数据存储和数据挖掘等各环节的大数据工具,包含完整的数据存储和数据处理方案,已经成为大数据的工业标准。然而Hadoop等开源软件的安全机制并不是非常完善,在用户认证、安全审计等方面还有不少的问题。随着企业对大数据平台的依赖逐步加强,如何构建安全的大数据平台成为企业用户关注的重要问题。本文主要介绍了百度公司在开源大数据平台的安全加固实践,重点介绍Hadoop用户认证、安全审计等方面的技术实现。

2 大数据平台安全实现

大数据平台是企业的核心数字资产,需要进行多层次、全方案安全加固和安全防护。开源大数据平台各子系统的安全定义和安全接口是不同的,目前尚没有统一的安全技术标准,各公司根据自身业务需求提供不同的安全加固方案。本文介绍百度公司针对开源大数据平台的安全加固

案例,开源大数据平台安全加固主要包括边界隔离、访问控制、日志审计和数据保护4个方面。

边界隔离是通过软件定义安全技术,将大数据平台划分到特定的数据域,以便统一实施特定的安全策略;访问控制涉及的主要技术是用户认证和用户授权,本文将介绍大数据平台与百度Giano用户认证系统集成方案;日志审计是“事后”安全分析的重要手段,通常可通过分析系统日志实现,本文将介绍Hadoop平台日志审计方案;数据保护用到的技术包括加密解密、数字水印、同态运算等。

①

<http://www.miiit.gov.cn/n973401/n5329862/n5329912/c5583935/part/5583944.pdf>

2.1 大数据平台用户认证

用户认证是大数据平台安全的重要安全环节,要求系统需明确回答以下两个问题:当前用户是谁;当前用户是不是仿冒的。为了解决用户认证的问题,很多公司在开源大数据平台中启用Kerberos认证,然而,在开源大数据平台中启用Kerberos认证存在以下问题。

- 配置复杂。Kerberos生成证书和配置步骤十分复杂,调试也十分困难,没有一定经验很难操作实施。
- 延展性不佳。机器的扩容和减少都会造成证书的重新生成和分发,给运维造成很大困难。
- 存在单点失败风险。Kerberos中心服务器存储所有用户的密钥,中心服务器故障会导致整个大数据平台不可用。同时,Kerberos要求严格的时钟同步,不然会导致认证失败。

考虑到Kerberos的复杂性,很多公司放弃了Kerberos认证。在没有启用Kerberos的情况下开源大数据平台的用户可随意被篡改。例如Hadoop平台的客户端程序默认从当前服务器的环境变量中读取

HADOOP_USER_NAME作为用户名,如果HADOOP_USER_NAME为空,则获取当前操作系统用户作为Hadoop的用户。因此仅需在客户端修改HADOOP_USER_NAME环境变量就可伪造用户进行非法操作。

百度公司大数据团队意识到开源大数据平台安全方面的问题后,深入研究Hadoop等开源系统的内部原理,自主研发了基于Giano的大数据平台用户认证机制,并将Giano作为开源大数据平台安全加固的核心技术方案,在内部推广使用。

2.1.1 百度Giano简介

Giano是百度公司开发的企业级安全平台,面向互联网数据中心(internet data center, IDC)环境提供一站式安全服务,包括安全登录与权限管理、统一身份管理、认证鉴权审计、访问控制、安全配置及变更管理、数据统计挖掘以及策略合规等。解决了从入口处的IDC安全登录、进程启用,到所有与服务及资源访问相关的一揽子安全问题,满足了应用环境隔离、跨域安全访问、托管代理认证、权限管理实施、细粒度资源分配以及服务状态全程监控分析等一系列典型实际需求。

Giano的核心设计源于先进的访问控制安全理论,其实现过程中亦广泛采用了高效的开发工具和严格的质量控制管理,使得Giano在安全性、可用性以及实际效

率等方面均取得了满意的结果。例如通常情况下,其典型中心服务器和客户端每秒查询率(query per second, QPS)可达到十万甚至百万级别。

2.1.2 Hadoop与Giano集成

由于开源大数据平台本身组件比较多,各组件的集成比较琐碎。本文以Hadoop分布式文件系统(Hadoop distributed file system, HDFS)为例,说明Hadoop和Giano的集成方案。

Hadoop系统主要由NameNode、DataNode和Client 3部分组成。其中NameNode负责管理元数据,而DataNode负责具体文件的存储。NameNode的元数据信息主要包括目录结构(目录下有哪些子目录和文件)、权限信息(目录或文件的所有者、同组的用户及其他用户的读、写、进入权限)、文件分块存储信息(文件分成几个数据块以及数据块存储在哪些服务器上)。DataNode负责具体数据块的存储。Client是数据的访问客户端,通过客户端可以把数据写入集群,或者从集群读取数据。

Hadoop系统与Giano集成后的操作流程如图1所示。Client进行任何文件操作都需连接NameNode获取文件元数据信息,因此可在NameNode上进行权限判断。Client连接NameNode之前,可通过Giano系统获得当前用户的身份信息,根据用户名向Giano申请访问令牌,即安全认证凭证。拿到令牌信息后,客户端将用户名和令牌一起发送给NameNode, NameNode询问Giano验证用户是否合法。通过Giano验证后的用户名可作为Hadoop用户名,并以此用户名判断用户是否有权限。

集成Giano后的业务情景分析和特点分析如下。

- 用户名和登录的操作系统账号剥离。如Wang Ming和Li Qiang两名员工同时

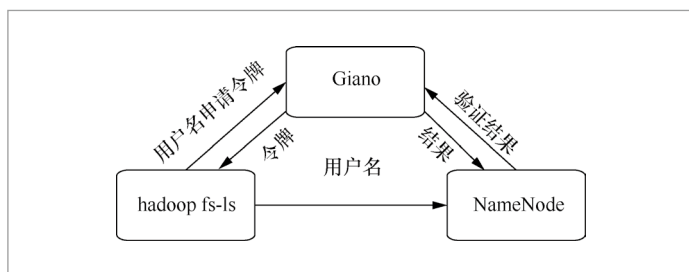


图1 Hadoop 与 Giano 集成后的操作流程

以工作账号登录同一台服务器访问Hadoop集群,集成Giano后的Hadoop系统仍然可以区分出来到底是哪名员工正在访问。

- 身份信息不可伪造。Hadoop与Giano集成后,服务端NameNode对客户端传递过来的用户名和令牌要进行验证,客户端无法仿冒其他用户。

2.2 大数据平台日志审计

2.2.1 日志审计方案

大数据平台日志审计是大数据平台安全的重要组成部分。日志审计需要记录用户在什么时间从哪台服务器对文件做出了什么操作。对于Hadoop系统而言,可审计的文件操作如下。

- Rename: 文件改名。
- Open: 打开文件。
- Delete: 删除文件。
- listStatus: 列出子目录。
- create: 创建文件。
- setPermission: 设置文件权限。
- GetFileInfo: 得到文件信息,包括路径、文件长度、是否是目录、数据块副本数、数据块大小、最后修改时间、最后获取时间、文件权限等信息。
- mkdirs: 创建目录。

Hadoop日志审计功能开启之后,所有的文件操作都会被记录下来。在大型Hadoop集群中NameNode每秒需要处理数万个文件操作,如果任何操作都记录下来会严重影响NameNode的处理速度。另外,日志审计需要频繁比对审计规则,有可能导致NameNode内存碎片化,进而引发大量Java垃圾收集(garbage collection, GC)操作,也会影响NameNode的处理速度。为了实现Hadoop日志审计功能,同时又能保证

NameNode处理速度,本文提出了一种基于规则的高效日志审计方案,可为Hadoop提供按需审计服务。实际应用场景中,可利用本方案实现敏感数据的日志审计。

本技术方案可通过规则配置开启任意目录的若干操作审计,例如用户A非常关心其业务数据什么时间被用户删除掉,他可以仅对目录启用删除审计;用户B需要知道其业务数据被哪些人访问过,他可以对目录开启读取审计。管理员可根据业务需求随时调整审计规则,包括增加审计目录、调整需要审计的操作指令等。规则调整后定时生效,不需要重启集群。

如果hdfs://ns1/user/user1/a.db需要审计改名和打开操作,而hdfs://ns2/user/user1/b.db需要审计创建和删除操作,则具体的配置规则如下。

```
<audit-config>
  <audit-directory>
    <path>hdfs://ns1/user/user1/
a.db</path>
    <commands>rename,open</
commands>
  </audit-directory>
  <audit-directory>
    <path>hdfs://ns2/user/user1/
b.db</path>
    <commands>create,delete</
commands>
  </audit-directory>
</audit-config>
```

2.2.2 目录匹配算法

本文提出的技术方案是通过NameNode插件机制实现的,核心原理是扩展HdfsAuditLogger接口,增加高效的目录匹配算法实现目录匹配。NameNode

的目录匹配算法是采用字符树的判断方法,把所有需要判断的目录组成一个字符树。字符树的定义如下:

```
Class CharacterTree {
    Char c; // 当前节点字符
    CharacterTree[] subTree; //子树
    数组,按数组元素CharacterTree的字符c
    有序,可以加快查找速度
    Boolean leaf; //是否是叶子节点,
    即最后一个字符
    Set<String> commands;// 仅 leaf
    = true时才不为空,保存目录的审计命令的
    集合
}
```

本技术方案核心内容包括动态加载、目录匹配、汇总输出等模块。动态加载模块的主要功能是从配置文件中读取目录判定规则,并对规则配置项进行必要的过滤和容错处理,生成目录判断规则树。该模块定期读取配置文件,并判断该配置文件是否发生了修改,如果发生了修改,则重新生成一个判断规则树,并用新生成的树替换之前的树。其具体实现方案是hdfs-site.xml增加配置项用于设置审计规则,示例代码如下。

```
<property>
    <name>dfs.namenode.audit.
file</name>
    <value>/etc/hadoop/auditfile.
xml</value>
</property>
<property>
    <name>dfs.namenode.audit.file.
check.interval</name>
    <value>60000</value>
</property>
<property>
    <name>dfs.namenode.audit.
loggers</name>
```

```
<value>org.apache.hadoop.hdfs.
namenode.CharacterTreeAuditLogger</
value>
</property>
```

动态加载程序根据配置文件修改时间和上一次配置生效时间等因素判断是否需要重新加载应用规则,当且仅当配置文件发生更改后才会重新加载,以减少系统消耗。动态加载机制确保配置规则修改后,不需要重新启动集群即可生效。

当NameNode通过配置文件后,可以根据配置文件生成判断规则树,用于判定当前访问行为是否需要审计,判定规则树的生成流程如下。

步骤1 初始化字符树,treeRoot作为字符树的根节点,其字符是’/’。

步骤2 把所有的审计目录配置项读到一个新的列表中。

步骤3 依次对比列表中的目录配置项。如果这个配置项的Namespace和当前Namespace不相同,则忽略,否则转到步骤4。

步骤4 取出配置的路径部分,如“/user/user1/a.db”,如果以字符’/’结尾,则把最后’/’符号去掉。

步骤5 调用treeRoot.addPath(path, 1, commands),将当前路径的字符依次添加到字符树里。

步骤6 所有配置项处理完后,判断规则树生成完毕。

上述步骤4中需要多次调用addPath指令构建字符树,例如需要将path为“/user/user1/a.db”的配置项添加到字符树的流程如下。

步骤1 从path取出一个字符,假如当前字符为“u”,执行步骤2。

步骤2 判断子树节点是否等于当前字符。如果相等,则执行步骤4,否则执行步骤3。

步骤3 初始化一个新的字符子树, 它的根节点为当前字符, 并将该子树添加到当前字符树中, 执行步骤4。

步骤4 把index的值增加1, 如果当前index的值小于path的长度, 重新执行步骤1。如果当前index的值等于path的长度, 执行步骤5。

步骤5 设置当前字符树的leaf的值为“true”, 代表当前字符为最后一个字符, 整个path处理完毕。

当目录字符树生成以后, 系统会根据字符数判定当前访问行为是否需要进行审计, 如果需要审计, 则输出审计日志。审计结果输出是通过调用logAuditEvent审计方法实现的, 其接口定义如下。

```
public void logAuditEvent(boolean succeeded,
                           String userName,
                           InetAddress addr,
                           String cmd,
                           String src,
                           String dst,
                           FileStatus status,
                           UserGroupInformation ugi,
                           DelegationToken
                           SecretManager dtSecretManager) {
    // print information
}
```

当NameNode调用logAuditEvent方法输出审计日志时, 需要快速判断当前访问路径是否需要审计, 如果判定速度慢, 会拖慢NameNode处理速度, 进而影响整个集群的性能。本方案利用字符树进行快速判定, 其主要特点如下。

- 在审计过程中, 一次只判断一个字符, 任何字符匹配不成功就立即返回。单线程测试每秒可执行1 000万次判定, 不会影响NameNode的处理速度。

- 在判断过程中没有锁, 多线程访问没有竞争。

- 在审计过程中, 一次只取一个字符, 不产出临时字符串, 可减少NameNode GC压力。

2.3 实际应用

该技术方案已经在百度公司安全管理部的大数据集群稳定运行了半年。该集群有1 100台服务器、38 000多个CPU核心、30 PB存储空间、日运行作业近2万个, 存有安全软件开发工具包 (software development kit, SDK)、威胁情报、网址安全、电脑安全、内网安全等数据。这些数据分为3类: 不涉及隐私的数据; 用户Cookie等虽然不直接包含隐私数据, 但可以通过关联找到用户行为的半隐私数据; 用户手机号、访问记录等用户隐私数据。

通过Hadoop本身的文件权限控制和Giano的集成, 可以防止数据被非法访问。通过审计程序可以跟踪敏感数据的访问情况。访问分为两种, 第一种是非法拒绝访问的记录, 记录了哪位用户尝试访问未授权的文件。第二种是有权限用户的访问, 此信息对于数据泄露的追踪有关键作用。

审计程序先把数据写到本地硬盘上, 然后通过消息传输系统上传到Hadoop平台进行分析。这个数据仍然是隐私数据, 需要授权才能访问, 并且需要审计。

3 结束语

本文总结归纳了在生产实践过程中针对开源大数据平台的安全加固方案, 重点介绍了关于Hadoop平台的用户认证和日志审计方案技术实现。其中, Hadoop用户认证是集成百度Giano系统实现的, Hadoop

日志审计是通过扩展NameNode提供高效目录匹配算法实现的。这些技术方案具备一定的普适性,可根据企业当前技术环境灵活使用。

参考文献:

- [1] 冯登国, 张敏, 李昊. 大数据安全与隐私保护[J]. 计算机学报, 2014, 37(1): 246-258.
- FENG D G, ZHANG M, LI H. Big data security and privacy protection[J]. Chinese Journal of Computers, 2014, 37(1): 246-258.
- [2] 杨曦, GUL J, 罗平. 云时代下的大数据安全
技术[J]. 中兴通讯技术, 2016, 22(1): 14-18.
- YANG X, GUL J, LUO P. Security technology of big data in the cloud era[J]. ZTE Technology Journal, 2016, 22(1): 14-18.

作者简介



刘杰 (1979-), 男, 百度(中国)有限公司商业安全研发部主任、研发架构师, 主要研究方向为分布式计算、Linux内核文件系统、海量数据存储与挖掘、大数据架构及数据安全防护技术。

收稿日期: 2017-08-01