# Part1 by阳隆荣

（蓝色字体标出的代表专业词汇）

摘要：

As one of the most popular platforms for processing big data, Hadoop has low costs, convenience, and fast speed. However, it is also a significant target of data leakage attacks, as a growing number of businesses and individuals store and process their private data in it. How to investigate data leakage attacks in Hadoop is an important but long-neglected issue. This article first presents some possible data leakage attacks in Hadoop. Then an investigation framework is proposed and tested based on some simulated cases.

Hadoop（主语提前）作为处理大数据最流行的平台之一，具有低成本、方便快捷的特点（增词）。然而，随着越来越多的企业和个人在其中存储和处理他们的私人数据，它也是数据泄漏攻击的一个重要目标。如何研究Hadoop中的数据泄漏攻击是一个重要但长期被忽视的问题。本文首先介绍了Hadoop中一些可能的数据泄漏攻击。在此基础上，提出了一个调查框架，并通过模拟案例进行了验证（被动句式转换为主动句式|拆句）。

简介：

Hadoop is one of the most popular platforms for big data storage and analysis. It is widely used in many fields, such as manufacturing, healthcare, insurance, and retail because of its powerful processing capacity, huge storage capacity, scalability, and relatively low cost. Nowadays, a growing number of individuals and businesses store and process their private data in Hadoop, and this valuable data has become an important target of hackers.

Hadoop是最流行的大型数据存储和分析（修饰成分提前作定语）平台之一。它以其强大的处理能力、巨大的存储容量、可扩展性和相对低廉的成本（汉语中先因后果），在制造业、医疗保健、保险、零售等领域得到了广泛的应用。如今，越来越多的个人和企业在Hadoop中存储和处理他们的私有数据，这些有价值的数据已经成为黑客的重要攻击（增词）目标。

In order to prevent these types of attacks, investigating them and reconstructing the entire scenario is very important. Based on the forensic results [5] [6], the vulnerability of Hadoop can be found and the attackers can be accused. Although the Hadoop investigator often faces many challenges, there is still little research work in this area. Current challenges faced by the Hadoop investigator include, but are not limited to, how to locate the data leakage node among thousands of Hadoop nodes, how to obtain reliable evidence in a complex and rapidly changing Hadoop environment, and how to investigate attacks based on Hadoop audit logs, which usually contain a large quantity of redundant and multi-user data.

为了防止这些类型的攻击，对它们进行调查并重建整个场景是非常重要的。根据电子取证结果[5][6]，可以发现Hadoop的漏洞，并对攻击者进行指控（被动句式转换为主动句式）。尽管Hadoop研究者经常面临许多挑战，但在这方面的研究工作仍然很少。Hadoop调查员目前面临的挑战（被动句式转换为主动句式|主语提前）包括但不限于如何在数千个Hadoop节点中定位数据泄漏节点，如何在复杂且快速变化的Hadoop环境中获取可靠证据，以及如何基于Hadoop审核日志调查攻击，它通常包含大量的冗余和多用户数据。

Considering these challenges, in this article, we first present some possible data leakage attacks in Hadoop, then analyze the difficulties of investigation. After that, an investigating framework is proposed and tested based on simulated cases. The framework is composed of the data collector and data analyzer. The data collector collects Hadoop logs, Fsimage files, our own monitor logs, and other information from each node actively or on demand, and transmits them to the data analyzer. Then, the data analyzer analyzes the data with automatic methods to find the stolen data, find the attacker who stole this data, and reconstruct the crime scenario.

考虑到这些挑战，本文首先介绍了Hadoop中可能存在的数据泄漏攻击，然后分析了研究的难点。在此基础上，提出了一个基于模拟案例的调查框架并进行了测试（被动句式转换为主动句式）。该框架由数据采集器和数据分析器组成。数据采集器从每个节点主动或按需收集（修饰成分提前）Hadoop日志、Fsimage文件、我们自己的监视器日志和其他信息，并将它们传输到数据分析器。然后，数据分析器使用自动方法对数据进行分析，以查找被盗数据，找到窃取此数据的攻击者，并重建犯罪场景。

相关工作：

Current Hadoop security research mainly includes trusted audit mechanisms, access control, data encryption, and so on. In [1], an architecture was proposed to fight against advanced persistent threats (APT) targeted against data stored in the HDFS (Hadoop distributed file system). This architecture is based on the trusted platform group (TPM) and trusted computing group (TCG). With the help of this architecture, all of the operations triggered by users can be audited. In this way, suspicious actions can be discovered and evidence can be retrieved for future investigation. However, it results in a serious impact on performance and a huge size of logs, which make it impractical.

目前Hadoop安全研究主要包括可信审计机制、访问控制、数据加密等。在[1]中，我们（增词）提出了一种体系结构（被动句式转换为主动句式）来对抗针对HDFS（Hadoop分布式文件系统）中存储的数据的高级持久性威胁（APT）。该体系结构基于可信平台组（TPM）和可信计算组（TCG）。在这种体系结构的帮助下，用户触发的所有操作（被动句式转换为主动句式）都可以被审计。这样，就可以发现可疑的行为，检索证据（被动句式转换为主动句式）以备将来调查之用。但是，它会对性能造成严重影响，并且会导致日志的巨大规模，这使得它不切实际。

In [2], a complicated access control mechanism was adopted to ensure the security of Hadoop. It protects data in Hadoop from unauthorized access, accidental leakage and loss, and breach of tenant confidentiality. “ACL Access Control” and “Kerberos” are two of them that have been adopted by the new versions of Hadoop. Although this can make Hadoop more secure, it is not able to prevent attacks if criminals use legal user accounts, and nothing can be done for direct data access in the operating system layer because these mechanisms are only for the application layer.

在文[2]中，为了保证Hadoop的安全性，采用了一种复杂的访问控制机制（被动句式转换为主动句式）。它保护Hadoop中的数据免受未经授权的访问、意外泄漏和丢失，以及违反租户机密性的操作（增词）。“ACL访问控制”和“Kerberos”是Hadoop新版本所采用的（修饰成分（定语从句）提前翻译）两种方法。尽管这可以使Hadoop更加安全，但（增词）如果罪犯使用合法的用户帐户，它就无法防止攻击，而且无法在操作系统层直接访问数据（被动句式转换为主动句式|否定转移），因为这些机制仅适用于应用程序层。

In [3], a secure Hadoop architecture was proposed that adds encryption and decryption functions in the HDFS. Through this method, data in the HDFS will not be readable even if it is stolen because the attacker does not have the secret key. Data is protected in this way. Although it is a fundamental solution for securing Hadoop, what cannot be ignored is the impact on performance and the fact that this approach does not guard against attackers who use a legal account.

在[3]中，提出了一种安全的Hadoop体系结构（被动句式转换为主动句式），在HDFS中增加了加密和解密功能。通过此方法，因为（汉语中先因后果）攻击者没有密钥，所以即使HDFS中的数据被窃取，也无法读取。以这种方式保护数据。尽管它是保护Hadoop安全的基本解决方案，但（增词）不可忽视的是对性能的影响，以及这种方法无法防范使用合法帐户的攻击者。

Unlike [1], our work concentrates on investigating (including evidence collection and analysis) data leakage attacks in Hadoop. Although [2] and [3] present some good solutions to prevent data leakage, they are far from silver bullets. This type of attack still takes places frequently, so research on how to investigate them after they happen is necessary and important. We have not found any other work on such an issue.

与[1]不同，我们的工作集中在调查（包括证据收集和分析）Hadoop中的（修饰成分提前）数据泄漏攻击。尽管[2]和[3]提出了一些很好的防止数据泄漏的（修饰成分提前）解决方案，但它们远不是（表示否定意义的词）灵丹妙药。这种类型的攻击仍然频繁发生，因此研究如何在攻击发生后对其进行调查是必要的（删词）。在这个问题上，我们还没有找到其他方法。

Hadoop中的数据泄漏攻击：

The data leakage attacks in Hadoop mainly include but are not limited to the following categories.

Hadoop中的数据泄漏攻击主要包括但不限于以下几类。

Application layer data leakage. This means attackers can obtain private data by application-layer vulnerability or malware. For example, a vulnerability in the current Hadoop audit mechanism is that it only records the operation type, time, and content, but no information on who did this operation. Suppose in a company, Alice, Bob and Cindy belong to a group named Hadoop, which is responsible for managing their company’s Hadoop system. One day, Bob stored a file named star-project.txt in the HDFS, and set the permission of this file to group readable. Soon after that, Bob found the content of his file was known by his rival, yet the content was not leaked by him, so who is the traitor, Alice or Cindy? We cannot find the answer just from Hadoop logs because they do not record who did this. The only clue we have is that Bob knows which file was stolen. In addition, the Hadoop audit logs might have been tampered

应用层数据泄漏。这意味着攻击者可以通过应用层漏洞或恶意软件（修饰成分提前）获取私有数据。例如，当前Hadoop审计机制中的（修饰成分提前）一个漏洞是，它只记录操作类型、时间和内容，而不记录谁执行了此操作（否定转移）。假设在一家公司中，Alice、Bob和Cindy属于一个名为Hadoop的组，该组负责管理他们公司的Hadoop系统（拆句）。一天，Bob在HDFS中存储了一个名为star-project.txt的文件，并将该文件的权限设置为组可读。不久之后，鲍勃发现他的档案内容被他的对手知道了，但他并没有泄露任何内容（被动句式转换为主动句式），那么谁是叛徒，爱丽丝还是辛迪？我们无法仅仅从Hadoop日志中找到答案，因为它们没有记录是谁做的。我们唯一的线索是鲍勃知道哪个文件被偷了。此外，Hadoop审计日志可能已被篡改。

Operating system layer data leakage. This means if attackers have the permission to log-in to the host operating system of a Hadoop node, they can bypass the monitor of Hadoop and steal data (not only the HDFS file block but also the temporary result of the MapReduce task) directly in the OS layer. For example, Bob stored a file named companyA.7z in the HDFS, and this file is larger than 64 Mb, so it was divided into several blocks, each block possibly being saved in different machines. Both the Hadoop administrator and the root user have write permission to all the blocks. Hence, if the attacker wants to steal the file, they do not have to get permission in the HDFS, they just need to get the locations of every block with the help of the name node logs (or through other ways such as the network monitor and so on), then steal the block directly from the physical machine if they have root permission of the host OS. The only clue we have is which file was stolen. The Hadoop audit logs might also have been tampered with by the attackers.

操作系统层数据泄漏。这意味着，如果攻击者有权限登录到Hadoop节点的主机操作系统，他们可以绕过Hadoop的监控，直接在OS层窃取数据（不仅是HDFS文件块，也是MapReduce任务的临时结果）。例如，Bob在HDFS中（修饰成分提前）存储了一个名为companyA.7z的文件，该文件的大小超过64mb，因此它被分成几个块，每个块可能保存在不同的机器中。Hadoop管理员和根用户都拥有对所有块的写权限。因此，如果攻击者想要窃取文件，他们不需要在HDFS中获得权限，（删词）只需要借助名称节点日志（或通过网络监视器等其他方式）（修饰成分提前）获取每个块的位置，然后如果拥有主机操作系统的根权限，他们（改变主语位置）能直接从物理机上窃取块。我们唯一的线索是哪个文件被偷了。Hadoop审计日志也可能被攻击者篡改。

To investigate the cases above, we are faced with the following challenges. First, Hadoop clusters often contain hundreds or thousands of nodes. When attacks happen, it is almost impossible to（固定句式，it is无需翻译） investigate all the nodes because either it requires too much time or the evidence has been changed during the investigations. Most attacks involve only a few nodes, and we need to find an efficient way to locate the nodes being attacked and concentrate our resources on them. Second, the Hadoop environment is complex and fast changing. The evidence may be tainted by new data or tampered with by attackers. Hence, how can we obtain reliable evidence from such an environment? Third, current Hadoop audit logs are far from sufficient in investigations as they lack the information on who did the crimes. Furthermore, these audit logs contain a large quantity of redundant and multi-user data, making evidence analysis extremely difficulty.

为调查上述案件，我们面临以下挑战。首先，Hadoop集群通常包含数百或数千个节点。当攻击发生时，几乎不可能对所有节点进行调查，因为这需要太多的时间，或者在调查过程中（修饰成分提前）证据会发生变化。大多数攻击只涉及几个节点，我们需要找到一种有效的方法来定位被攻击的节点，并将我们的资源集中在这些节点上。其次，Hadoop环境复杂且变化迅速。证据可能被新数据污染或被攻击者篡改。因此，我们如何从这样的环境中获得可靠的证据？第三，目前的Hadoop审计日志在调查中远远不够，因为它们缺乏犯罪者的信息。此外，这些审计日志包含大量的冗余和多用户数据，使得证据分析极为困难。

**总结：**

①专业词汇较多

例子：

Application layer data leakage应用层数据泄漏

trusted audit mechanisms 可信审计机制

forensic results 电子取证结果

②英语中的被动句式在汉语中多转换为主动句式

例子：

A complicated access control mechanism was adopted to ensure the security of Hadoop.

为了保证Hadoop的安全性，采用了一种复杂的访问控制机制。

An investigating framework is proposed and tested based on simulated cases.

提出了一个基于模拟案例的调查框架并进行了测试。

In [1], an architecture was proposed to fight against advanced persistent threats (APT) targeted against data stored in the HDFS (Hadoop distributed file system).

在[1]中，我们提出了一种结构来对抗针对HDFS（Hadoop分布式文件系统）中存储的数据的高级持久性威胁（APT）。

In this way, suspicious actions can be discovered and evidence can be retrieved for future investigation.

这样，就可以发现可疑的行为，并检索证据以备将来调查之用

③英语中常常果前因后，但在汉语中常常因前果后

例子：

It is widely used in many fields, such as manufacturing, healthcare, insurance, and retail because of its powerful processing capacity, huge storage capacity, scalability, and relatively low cost.

它以其强大的处理能力、巨大的存储容量、可扩展性和相对低廉的成本（汉语中先因后果），在制造业、医疗保健、保险、零售等领域得到了广泛的应用。

Data in the HDFS will not be readable even if it is stolen because the attacker does not have the secret key.

因为攻击者没有密钥，所以即使HDFS中的数据被窃取，也无法读取。

④修饰成分多数情况下需要翻译在被修饰词之前，英语中的修饰成分多用在被修饰词之后，而汉语中的修饰成分多用在被修饰词之前

例子：

This is a vulnerability in the current Hadoop audit mechanism.

这是当前Hadoop审计机制中的一个漏洞。

The evidence has been changed during the investigations.

在调查过程中证据会发生变化。

⑤否定转移，英语中的否定多用在主语或宾语中，汉语中的否定多用在谓语中

例子：

No information on who did this operation.

不记录执行了此操作的人。

Nothing can be done for direct data access in the operating system layer.

无法在操作系统层直接访问数据。

⑥需要改变主语位置，有时需要将主语提前至句子的开头，否则会“头重脚轻”，有时也需要将主语提后；

例子：

As one of the most popular platforms for processing big data, Hadoop has low costs, convenience, and fast speed.

Hadoop作为处理大数据最流行的平台之一，具有低成本、方便快捷的特点。

Hence, if the attacker wants to steal the file, they do not have to get permission in the HDFS, they just need to get the locations of every block with the help of the name node logs (or through other ways such as the network monitor and so on), then steal the block directly from the physical machine if they have root permission of the host OS.

因此，如果攻击者想要窃取文件，他们不需要在HDFS中获得权限，只需要借助名称节点日志（或通过网络监视器等其他方式）获取每个块的位置，然后如果拥有主机操作系统的根权限，他们能直接从物理机上窃取块

⑦并列句或主从句可以拆句翻译

例子：

Suppose in a company, Alice, Bob and Cindy belong to a group named Hadoop, which is responsible for managing their company’s Hadoop system.

假设在一家公司中，Alice、Bob和Cindy属于一个名为Hadoop的组，该组负责管理他们公司的Hadoop系统。

Then an investigation framework is proposed and tested based on some simulated cases.

在此基础上，提出了一个调查框架，并通过模拟案例进行了验证。

⑧科技论文中的用词非常精炼，大部分情况下无需删词，需要增词的情况可能出现得更多

例子：

This valuable data has become an important target of hackers.

这些有价值的数据已经成为黑客的重要攻击目标。

⑨科技论文中需要翻译的复杂句式不多，要力求逻辑的简洁明了

# Part2 by张昊宇

An Investigation Framework for Hadoop Forensics

Hadoop取证调查框架

Considering the above challenges, an investigation framework aimed at data leakage attacks in Hadoop was proposed. In fact, it can also be extended to investigate other kinds of attacks in Hadoop. This framework is composed of （固定结构，被动语态）many data collectors and one data analyzer. The data collector is located in the kernel of the host operating system in Hadoop nodes (mainly data nodes). It is designed based on a demand collection policy. First, it actively monitors the accesses to important data on each node and transmits these behavior logs to the data analyzer. Then it collects the required evidence (such as disk images and network traffic logs) from specific nodes according to the analysis results. The data analyzer is located in a specific forensic server. It analyzes the data with automatic algorithms to find the stolen data, find the attackers who stole this data, and reconstruct the entire scenario. It also sends commands to specific collectors to obtain more evidence.

考虑到上述挑战，我们（补充主语）提出了针对Hadoop中数据泄漏攻击的研究框架（被动转主动）。实际上，它也可以用作（词义扩展）研究Hadoop中的其他类型的攻击。该框架由许多数据收集器和一个数据分析器组成。数据收集器位于Hadoop节点（主要是数据节点）中的主机操作系统的内核中。它是根据需求收集策略设计的。数据收集器的工作流程如下：（添加语句使中文语意连贯）首先，它主动监视每个节点上对重要数据的访问，并将这些行为日志传输到数据分析器。然后根据分析结果从特定节点收集所需的证据（先因后果）（例如磁盘映像和网络流量日志）。数据分析器位于特定的取证服务器中。它使用自动算法对数据进行分析，以找到被盗的数据和（合译）偷窃该数据的攻击者，并重建整个场景。它还将命令发送给特定的收集器以获得更多证据。

There are serval advantages in the above design. Through the cooperation between the data collectors and the data analyzer, the data leakage attacks can be detected immediately by the live monitor and the automatic analysis algorithm, and the node where it happened can be located quickly according to the information in the collected behavior logs. The demand collection policy can reduce cost because the large sized evidence (e.g., disk images) is only collected on several nodes based on analysis results. The live kernel-level monitors ensure the evidence is collected before contamination, and the collector itself is transparent to normal Hadoop users and attackers. Moreover, all evidence is stored in an independent and carefully protected forensic server. This makes the evidence more reliable.

上述框架（异译便于理解）有不少优势。通过数据采集器和数据分析器的协作，可以通过实时监控器和自动分析算法立即检测到数据泄漏攻击，并根据收集到的行为日志中的信息快速定位发生泄漏的节点（被动转主动）。仅根据分析结果就可以在在多个节点上收集大型证据（例如磁盘映像）可以为需求收集策略降低成本（先因后果）。实时的内核级监控器确保在复写（专业术语）之前收集证据，并且收集器本身对一般的Hadoop用户和攻击者透明。此外，所有证据都存储在独立且经过精心保护的电子取证（专业术语）服务器中。这些设计（增词）使证据更加可靠。

Data Collector

数据收集器

The architecture of the data collector is shown in Fig. 1. The data it collects includes Hadoop logs, the Fsimage file, our own monitor (i.e., HProgger) logs and images or logs of files, processes, networks, and system. In the live monitoring stage, data collectors on each node obtain Hadoop logs, HProgger logs, and the fsimage first. Then, the log filter changes them to a unified format. Finally, before transmitting them to the server, the log encryptor encrypts them to keep their credibility. In the demand collecting stage, the data collector and even some other forensic tools will collect file logs, network logs, process logs, and system images, and then transmit them to the server for analyzing. This data can also be encrypted before transmission. The two-stage design gives our framework the ability to collect data according to what the investigation needs, with reduced consumption and improved efficiency.

图1展示了数据收集器的体系结构（被动转主动）。它收集的数据包括Hadoop日志，Fsimage文件，我们自己的监视器（即HProgger）日志以及文件，进程，网络和系统的映像或日志。在实时监控阶段，每个节点上的数据收集器首先获取Hadoop日志，HProgger日志和fsimage。然后，日志过滤器将它们编译（专业术语）为统一格式。接着日志加密器会对它们进行加密以保持其可信度，最后再将它们传输到服务器中（语序调整）。在需求收集阶段，数据收集器甚至其他一些电子取证工具将收集文件日志，网络日志，过程日志和系统映像，然后将它们传输到服务器进行分析。也可以在传输之前对该数据进行加密（被动转主动）。这种两阶段设计使我们的框架能够根据调查需要收集数据，从而（递进）减少了消耗并提高了效率。

Hadoop logs include name node logs, job tracker logs, data node logs, and task tracker logs. Our framework will mainly focus on name node logs and data node logs because data leakage attacks are mainly recorded in them.

The Fsimage is the image file of the Hadoop system. Unlike general system image files, the fsimage only records the data directory and data distribution. In simple words, the fsimage records the meta information of files in the file system and the distribution of blocks.

Hadoop日志包括名称节点日志，作业跟踪程序日志，数据节点日志和任务跟踪程序日志。我们的框架将主要关注名称节点日志和数据节点日志，因为数据泄漏攻击主要发生（异译）在其中。

Fsimage是Hadoop系统的映像文件。与常规系统映像文件不同，fsimage仅记录数据目录和数据分布（专业术语）。简单来说，fsimage记录的是文件系统中文件的元信息和文件块的分布。

HProgger is our own monitor tool which is implemented based on Progger [4]. Progger is an open source data monitor tool running in the kernel of Linux. It modifies the addresses of certain system calls in the system call table, allowing it to record the invocations of these calls. Progger is able to help us acquire more reliable evidence because data leakage attacks in Hadoop will inevitably invoke some system call to access certain files and directories. Although Progger is powerful in auditing, it causes a serious downgrade in performance and delays every operation. Additionally, the amount of logs that Progger generates is unacceptable, i.e. it generates hundreds of megabytes of logs in just a few minutes. To make it applicable in Hadoop forensics, performance must be improved and the number of logs must be reduced as much as possible. Thus, we design HProgger (i.e. Hadoop-Progger). It isimproved in several aspects:

HProgger是我们自己的基于Progger [4]实现的监视工具（状语从句合译）。Progger是在Linux内核中运行的开源数据监视工具（语序调整）。 它修改了系统调用表中某些系统调用的地址，从而（递进）使其可以记录这些调用的调用。因为Hadoop中的数据泄漏攻击将不可避免地调用某些系统调用来访问某些特定文件和目录。所以Progger能够帮助我们获得更加可靠的证据。（先因后果）尽管Progger的审核功能强大，但它会严重降低性能并延迟每次操作。另外，Progger它在短短几分钟内生成了数百兆的日志，这种巨大的日志数数据不可接受的（语序调整）。为了使其适用于Hadoop电子取证，必须提高性能并且必须尽可能减少日志数量（被动转主动）。因此，我们设计了HProgger（即Hadoop-Progger）。相较于Progger（增句），在几个方面进行了改进：

1. Only three system calls (i.e. OPEN, COPY, MOVE) will be recorded.

2. The log format is minimized; only necessary information such as operation type, time, operator, and so on remains.

3. Instead of monitoring all the directories, only those associated with the Hadoop system, such as the directory where Hadoop stores HDFS files, are retained. Hence, compared to Progger, HProgger is lighter and more applicable for Hadoop forensics.

1.仅记录三个系统操作（即OPEN，COPY，MOVE）（被动转主动）。  
2.将日志格式最小化；仅保留必要的信息，例如操作类型，时间，操作员等（被动转主动）。  
3.保留所有与Hadoop系统关联的目录（例如Hadoop存储HDFS文件的目录），而不是监视所有目录。因此，与Progger相比，HProgger更智能（一词多义），更适用于Hadoop电子取证。

Data Analyzer

数据分析仪

The data analyzer is composed of a decryptor, a log database, a fsimage manager, an automatic detector, and the management and presentation module (as shown in Fig.2). When data arrives at the analyzer, it is first processed by the decryptor. Then Hadoop logs and HProgger logs are stored in the log database and the fsimage files are stored in the fsimage manager. Users can query and manage the logs and the fsimage in the management and presentation module. This data is formatted before presentation. The automatic detector analyzes the logs automatically to find the data that was stolen, the attacker who stole the data, and the process of the attacks. The results will also be presented in the management and presentation module.

数据分析器由解密器，日志数据库，fsimage管理器，自动检测器以及管理和显示模块组成（如图2所示）。数据到达分析器后，首先由解密器处理。然后将Hadoop日志和HProgger日志存储在日志数据库中，并将fsimage文件存储在fsimage管理器中。用户可以在管理和显示模块中查询和管理日志和fsimage。此数据在演示前已经被格式化。自动检测器会自动分析日志，以查找被盗的数据，偷走数据的攻击者以及攻击过程（合译）。管理和显示模块中也将显示结果（被动转主动）。

In order to detect data leakage attacks as soon as possible, some automatic detection algorithms are designed. For an OS-layer data leakage attack, the detection algorithm works in four dimensions: abnormal directory, abnormal user, suspicious block proportion, and abnormal operation. The algorithm detects abnormal directories and abnormal users in the file datasets to find out if suspicious blocks exist, and it calculates the suspicious block proportion. In the meantime, the algorithm monitors the trend of operations in the block datasets to detect abnormal values, and also calculates the suspicious block proportion. If any of these four dimensions gives a warning, then attacks may have happened and the investigators can continue their investigation according to the warning information.

为了尽快检测数据泄漏攻击，设计了一些自动检测算法（被动转主动）。对于操作系统层（专业术语）的数据泄露攻击，检测算法的工作范围包括四个方面：目录异常，用户异常，可疑块比例和操作异常。该算法检测文件数据集中的异常目录和异常用户，以找出是否存在可疑块，并计算出可疑块比例（合译）。同时，该算法监视块数据集中的操作趋势以检测异常值，并计算可疑块比例。如果这四个维度中的任何一个发出了警告，则可能发生了数据泄漏（补充）攻击，调查人员可以根据警告信息继续进行调查。

Abnormal Directory (AD): Normal Hadoop system file operations will involve fixed directories because these directories are configured by Hadoop system profiles. Hence, in collected logs, if any directory that is out of this range is found, an attack may have happened. The suspicious blocks can thus be found by analyzing records that contain these abnormal directories.

异常目录（AD）：正常的Hadoop系统文件操作将涉及固定目录，因为这些目录是由Hadoop系统文件配置的。因此，在收集的日志过程中，如果找到了超出此范围的目录（被动转主动），则可能发生了攻击。因此，可以通过分析包含这些异常目录的记录来找到可疑块。（被动转主动）

Abnormal User (AU): By studying the HProgger logs, it can be concluded that all of the Hadoop system file operations except copying files from the HDFS to a local machine will involve only one operator, i.e. the Hadoop super user.Hence, if any other user instead of the Hadoop super user is found in HProgger logs, an attack may have happened. The suspicious blocks can thus be found by analyzing records that contain these abnormal directories. On the other hand, if an attacker wants to steal data directly from the operating system, they must have root permission or Hadoop super user permission. However, because more than one user can switch to the root, how can the real user who conducted the operations in root permission be found? Fortunately, HProgger is able to keep track of all users’data as long as they log in to the system, so the HProgger logs can be analyzed to find the real user.

异常用户（AU）：通过研究HProgger日志，可以得出结论，除了将文件从HDFS复制到本地计算机外，所有Hadoop系统文件操作将只涉及一个操作员，将其定义为Hadoop超级用户（改译）。在HProgger日志中找到其他用户而不是Hadoop超级用户（被动转主动），则可能发生了攻击。因此，可以通过分析包含这些异常目录的记录来找到可疑块被动转主动。另一方面，如果攻击者想直接从操作系统窃取数据，则他们必须具有root权限或Hadoop超级用户权限。但是，由于有多个用户可以切换到root用户，如何才能找到以root用户权限进行操作的真实用户？幸运的是，HProgger能够跟踪所有用户登录到系统的数据，因此可以分析HProgger日志以找到实际用户（被动转主动）。

Abnormal Operation (AO): Every Hadoop system operation in the HDFS is composed of reading files from the HDFS and writing files into the HDFS. By studying the HProgger logs generated by all kinds of Hadoop system operations, it can be concluded that every block is written only once and read at least twice during its life cycle (other backups of this block are not considered).However, if an attacker wants to steal a block from the physical machine, they need to copy (move) it to another directory and sometimes may rename the block, so if any of these operations are found, the attack may have happened. Moreover, the attack scenario can be reconstructed based on these operations. The attacker and the lost files may also be identified based on them.

异常操作（AO）：HDFS中的每个Hadoop系统操作都由从HDFS读取文件并将文件写入HDFS组成。通过研究由各种Hadoop系统操作生成的HProgger日志，可以得出结论，每个块在其操作周期（专业术语）中仅写入一次并至少读取两次（不考虑该块的其他备份被动转主动）。 要从物理机上窃取一个块，他们需要将其复制（移动）到另一个目录，有时可能会重命名该块，因此，如果找到这些操作中的任何一个，（被动转主动）则可能发生了攻击。此外，可以基于这些操作来重建攻击场景（被动转主动）。攻击者和丢失的文件也可以基于它们来识别。

Block Proportion (BP): Let SusBlockNum (File-ID) represent the number of suspicious blocks in a file dataset and AllBlockNumber (FileID) represent the total number of blocks in a file dataset.We define block proportion (BP) as the result of dividing SusBlockNum (FileID) by AllBlockNumber (FileID). Under normal circumstances, the value of BP should be 0 as no suspicious block should be found. The larger BP is, the greater the probability there has been an attack, so if BP is larger than 0, the algorithm will give a warning.

块比例（BP）：让SusBlockNum（文件ID）代表文件数据集中可疑块的数量，而AllBlockNumber（FileID）代表文件数据集中的总块数。我们将块比例（BP）定义为用SusBlockNum（FileID）除以AllBlockNumber（FileID）。在正常情况下，BP的值应为0，因为不应找到可疑块。 BP越大，发生攻击的可能性越大（从句简译），因此，如果BP大于0，则算法将发出警告。

The above describes the automatic detection algorithm for an OS-layer data leakage attack. Detection of an application-layer attack is much easier based on the HProgger log. For example, regarding stealing behavior using a legal user account, we are not able to find the thief based on Hadoop audit logs because necessary information is missing. However, based on HProgger, the thief can be identifi ed according to the following steps:

上面介绍了OS层数据泄漏攻击的自动检测算法，基于HProgger日志检测应用层攻击要容易得多。例如，如果（介词结构）使用合法用户帐户进行的窃取行为，由于缺少必要的信息，我们无法根据Hadoop审核日志找到窃取者（先因后果）。但是，基于HProgger，可以根据以下步骤识别窃取者：

Step 1: Find the HDFS blocks according to the fi le name and obtain their block ID from Hadoop logs or fsimage fi les.

Step 2: Filter out HProgger logs that contain“copyToLocal” or other necessary key operation types.

Step 3: Filter out the Hadoop logs that contain the block ID found in Step 1, then map these logs to the HProgger logs obtained in Step 2 based on their operation time. If such mapping records are found, the attacker name can be found in the relevant HProgger logs, and these mapping records can support each other.

步骤1：根据文件名称查找HDFS块，并从Hadoop日志或fsimage文件中获取其块ID。  
步骤2：过滤出包含“ copyToLocal”或其他必要键操作类型的HProgger日志。  
步骤3：过滤出包含在步骤1中找到的块ID的Hadoop日志，然后根据它们的运行时间将这些日志映射（专业术语）到在步骤2中获得的HProgger日志。 如果找到了此类映射记录，则可以在相关的HProgger日志中找到攻击者的名称，并且这些映射记录可以相互支持。

**总结：**

1. 专业词汇较多

如：images 映像 ；forensic 电子取证 ；map 映射

1. 多被动转为主动

如：To make it applicable in Hadoop forensics, performance must be improved and the number of logs must be reduced as much as possible.

为了使其适用于Hadoop电子取证，必须提高性能并且必须尽可能减少日志数量。

1. 英语为先果后因，汉语为先因后果

如：we are not able to find the thief based on Hadoop audit logs because necessary information is missing.

由于缺少必要的信息，我们无法根据Hadoop审核日志找到窃取者。

1. 较少长难句从句（长句但不难）

# Part3 by王煜文

# 案例研究

In order to test our method, a small Hadoop cluster using Virtual Box including one master node and 15 slave nodes is set up. The node environment is an i5-4590 3.3 GHz 2G Ubuntu 12.04 and Hadoop Version is 1.2.1.

为了测试我们提出的方案，在Virtual Box虚拟机上安装了有一个主节点和15个从节点的小型Hadoop群。（被动句转换为主动）这些节点配置的环境为：i5-4590 3.3GHz的CPU，2G内存版本为12.04的Ubuntu系统，版本1.2.1的Hadoop。

## 应用层面的数据泄露研究

We first conduct an investigation at the Hadoop application level. Suppose attackers stole a secret file named star-project.txt using a legal account. As we mentioned before, the Hadoop audit log cannot record the operator and there might be CSP employees who are authorized to modify the Hadoop logs and the Fsimage, so both the Hadoop logs and Fsimage may be contaminated. However, this case can be investigated based on the HProgger log, the Hadoop log, and the Fsimage according to following steps.

我们首先在Hadoop的应用层开始研究。假设攻击者使用合法帐户窃取了一个名为star-project.txt的秘密文件。像之前讲到的，由于Hadoop的审计日志无法记录操作者，并且有CSP工作人员被授权修改Hadoop日志和Fsimage的可能，这将可能会导致Hadoop日志和Fsimage被污染。（逻辑）但是，可以根据以下基于HProgger日志、Hadoop日志和Fsimage的步骤来调查这种情况。

**Step 1.** Find corresponding block IDs by file name in the Hadoop logs that come from the name node. The records show that block 8241916282566986668 belongs to the file star-project.txt.

**步骤1.** 在来自名称节点的Hadoop日志中按文件名查找相应的块id。记录显示块8241916282566986668属于文件star-project.txt。

**Step 2.** In application-level attacks, criminals access data through Hadoop commands, e.g. some Hadoop shell commands for file operations, such as CopyToLocal, so we can filter out this type of record from the HProgger logs. In this case, we can know from the filtered logs that in 21:04:28, User Alice used the CopyToLocal command to copy a file to a local machine, renamed it to abc.txt, and put it to /home/Alice. However, we cannot confirm that abc.txt is star-project.txt.

**步骤 2.** 在应用层面的攻击中，罪犯者通过Hadoop命令访问数据，比如一些用于文件操作的Hadoop shell命令，CopyToLocal就为其中之一，因此我们可以从HProgger日志中筛选出这种类型（命令）的记录。在这种情况下，我们可以从筛选的日志中得知，在21:04:28，用户Alice使用copy to local命令将文件复制到本地计算机，并将其重命名为abc.txt，然后将其放到/home/Alice目录下。但是，我们不能确定abc.txt是否为star-txt。

**Step 3.** Find the records containing the investigated file name or block IDs around the key records found in Step 2 in the HProgger logs, then obtain the timestamp of this operation. For example, we found a record showing that block 8241916282566986668 was opened in 21:04:28, and this record is just the former one of the key records we found in Step 2.

**步骤 3.** 在HProgger日志的步骤2中关键记录的周边进行搜索，找到包含所调查文件名或块id的记录，然后获取此操作的时间戳。例如，我们发现一条记录，显示块8241916282566986668在21:04:28打开，而这条记录只是我们在步骤2中找到的前一条关键记录。

**Step 4.** Search the records in the Hadoop logs according to the timestamps and block ID we found in the above steps. The results show that in 21:04:28, a HDFS\_READ was executed on block 8241916282566986668 and there were no other records at the same time. This record and the two records we found in HProgger can prove to each other that the criminal is User Alice. She copied star-project.txt to local by CopyToLocal and renamed it abc.txt.

**步骤 4.** 根据上面步骤中找到的时间戳和块ID搜索Hadoop日志中的记录。结果表明，在21:04:28，在块8241916282566986668上执行HDFS\_READ，同时没有其它（操作）记录。这个记录和我们在HProgger找到的两个记录可以相互证明罪犯是用户Alice。她通过CopyToLocal命令将star-project.txt复制到本地，并将其重命名为abc.txt。

In conclusion, Alice is the attacker. We can conduct second-stage forensics and find more information about this attack, for example, obtaining the network traffic in that node to detect whether Alice transmitted this file to another location. However, if attackers tampered with the Hadoop logs from the data node or the name node, how do we investigate such a case?

综上所述，Alice是攻击者。我们可以进行第二阶段的取证，以找到更多关于此攻击的信息，例如，通过获取该节点的网络流量去检测Alice是否将此文件传送到另一个位置。但是，如果在攻击者篡改了数据节点或名称节点的Hadoop日志的情况下我们该如何调查？

Hadoop logs from the data node are tampered with: In this case, we cannot obtain the relevant record from the Hadoop logs as was done in Step 4. However, any changes to the file or directory can be detected by HProgger. If we set HProgger to monitor /usr/local/hadoop/logs/, which is the directory where Hadoop logs are stored, we can find the illegal tampering from attackers.

数据节点中的Hadoop日志被篡改：在这种情况下，我们无法像步骤4中所做的那样从Hadoop日志中获取相关记录。但是，HProgger可以检测到对文件或目录的任何更改。如果我们设置HProgger作为监视器去监控/usr/local/hadoop/logs/这个存储hadoop日志的目录，我们就可以找到来自攻击者的非法篡改。

Hadoop logs from the name node are tampered with: In this case, we cannot obtain the block ID based on file name as was done in Step 1. Fortunately, we can obtain this relation from the Fsimage. That is why the data collector collects the Fsimage from each node.

名称节点中的Hadoop日志被篡改：在这种情况下，我们不能像在步骤1中那样基于文件名获取块ID。幸运的是，我们可以从Fsimage中获得这种关系。这就是数据采集器从每个节点收集Fsimage的原因。

The Fsimage is broken: If the Hadoop logs and the Fsimage both cannot be trusted, how do we investigate such a case? In this situation, we can first obtain the Fsimage from the secondary name node, which is the backup for the name node and synchronizes with it periodically in most Hadoop environments. Second, we can use the old version of the Fsimage that was saved by the Fsimage file manager.

Fsimage已损坏：如果Hadoop日志和Fsimage都不可信，我们如何调查这种情况？在这种情况下，我们可以首先从次级名称节点获取Fsimage，它是名称节点的备份，并周期性地在大多数的Hadoop环境中同步。其次，我们可以使用Fsimage文件管理器保存的Fsimage的旧版本。

In conclusion, even if the Hadoop logs and Fsimage files are tampered with, the investigator can still find the attackers based on our method.

总之，即使Hadoop日志和Fsimage文件被篡改，调查者仍然可以根据我们的方法找到攻击者。

## 操作系统层面的数据泄露研究

In this section, we conduct an investigation at the operating system level. Suppose an attacker knew the location of a secret file companyA.7z from the name node logs or some other way. Then they could read this file directly from these data nodes by logging in to the host OS. Such an operation cannot be monitored by the Hadoop audit mechanism, but we can find this attacker based on the HProgger logs, the Hadoop logs, and the Fsimage according to the following steps.

在本节中，我们将在操作系统的层次进行研究。假设攻击者从名称节点日志或者通过其他方式知道机密文件companyA.7z的位置。然后他们可以通过登录主机操作系统直接从这些数据节点读取该文件。因为Hadoop的审查机制无法监控此类操作，所以我们需要通过基于HProgger日志、Hadoop日志和Fsimage的方法去找到此攻击者。（逻辑）

**Step 1.** Find the corresponding block IDs by file name in the Hadoop logs that are from the name node. Different from the first experiment, we obtained three block IDs, i.e., 4067922487609870000, 6299596445748830000 and 7132029012670650000. That is because companyA.7z is larger than 64 Mb, so it is divided into three blocks by Hadoop.

**步骤 1.** 在来自名称节点的Hadoop日志中按文件名查找相应的块id。与第一次实验不同，我们得到了4067922487609870000、6299596445748830000和7132029012670650000三个块id。这是因为companyA.7z大于64Mb，所以它被Hadoop分成三个块。

**Step 2.** Find the HProgger records by block ID. These records show that User Alice copied block 5255825402 465249432 from the Hadoop system directory to home/alice at 11:41:44. We find Alice did the same thing to two other blocks that belonged to companyA.7z, but in different data nodes. That is to say, Alice copied all the blocks of companyA.7z directly from the operating system in different machines. We can conclude that Alice stole companyA.7z. We can conduct a second-stage forensic investigation and find more information about this attack, e.g., obtain the network traffic in that node to detect whether Alice transmitted this file to another location.

**步骤 2.** 按块ID查找HProgger记录。这些记录显示，用户Alice在11:41:44将块5255825402465249432从Hadoop系统目录复制到home/Alice目录下。我们发现Alice对另外两个属于companyA.7z的块做了相同的操作，但是是在不同的数据节点中。也就是说，Alice直接从不同机器上的操作系统中复制了companyA.7z的所有块。我们可以得出结论Alice窃取了companyA.7z。我们可以进行第二阶段的调查，找到有关此攻击的更多信息，例如，通过获取该节点中的网络流量去检测Alice是否将此文件发送到另一个位置。

Similar to the first experiment, if the Hadoop logs cannot be trusted, we can obtain the mapping between the file and the block through the Fsimage, and if the Fsimage is broken, we can obtain the old version from the secondary name node or the Fsimage file manager.

与第一个实验类似，如果Hadoop日志不可信，我们可以通过Fsimage获取文件和块之间的映射，如果Fsimage被破坏，我们可以从次级名称节点或Fsimage文件管理器获取旧版本。

We have shown how to find attackers when the only clue is the name of the file that was stolen, but how do we find which file was stolen? Earlier an automatic detection algorithm was proposed. In order to test its efficiency, we run the algorithm every five minutes in our cluster (the frequency can be customized). Figure 3 and Figure 4 show the monitoring results of companyA.7z during a period of 30 minutes.

我们已经展示了如何在唯一线索是被盗文件名的情况下找到攻击者，但是如何寻找被盗的文件？早期提出了一种自动检测算法。为了测试它的效率，我们每五分钟在我们的集群中运行一次算法（频率可以定制）。图3和图4显示了companyA.7z在30分钟内的监控结果。

Figure 3 shows that all three dimensions of the file companyA.7z changed from 21:20 to 21:55, and all values of the three dimensions are greater than their normal value. For other files, the values of all dimensions remained unchanged the entire time.

图3显示从21:2至21:55，文件companyA.7z的所有的三个维度都改变，并且三个维度的所有的值都大于其正常值。对于其他文件，所有维度的值始终保持不变。

Figure 4 shows the AO values of all three blocks of the file companyA.7z growing to 7 or 8, both greater than the normal value 6. For the blocks of the other files, the values remained at 6 all the time. Hence, suspicious operations must have happened to companyA.7z during 21:20 to 21:55. So the detailed suspicious HProgger logs were found, which showed that Alice copied block 6299596445748830000 and block 713202901267065000 from /app/hadoop/tmp/ dfs/data/ current to /home/abc/, and she copied block 40679224 87609870000 to home/Alice and then to home/abc. All the blocks of companyA.7z were stolen by Alice.

图4显示了文件companyA.7z中所有三个块的AO值都增长到7或8，都大于正常值6。对于其他文件的块，值始终保持在6。因此，在21:20到21:55期间，companyA.7z中一定发生了可疑的操作。对此，我们发现了可疑的HProgger日志，其中显示Alice从/app/hadoop/tmp/dfs/data/current目录向/home/abc/目录复制了6299596445748830000块和713202901267065000块，并将40679224 87609870000块复制到home/Alice，之后将其复制到home/abc。至此companyA.7z所有的块都被爱丽丝窃取了。

# 结论

This article presents a forensic framework including an on-demand data collection method and an automatic analysis method for data leakage attacks in Hadoop. It collects data from the machines in the Hadoop cluster to our forensic server and then analyzes them. With the automatic detection algorithm, it can find out whether there exist suspicious data leakage behaviors and give warnings and evidence to users. This collected evidence can be used to find the attackers and reconstruct the attack scenarios. Some simulated investigating cases have shown its efficiency.

本文提出了一种基于Hadoop的检测框架，包括按需数据的采集方法和数据泄漏攻击的自动分析方法。它从Hadoop集群中的机器中收集数据到我们的判别服务器，然后对它们进行分析。利用自动检测算法，可以发现是否存在可疑数据泄露行为，并向用户发出警告和证据。收集到的证据可用于寻找攻击者和攻击场景的重建。一些仿真调查案例显示了其有效性。

# 翻译总结

1. 专业词汇、专有名词多

例如：Fsimage（Hadoop的日志镜像文件），Ubuntu（操作系统）

1. 论文句式简单，无复杂的长难句，但是句子间内涵逻辑联系。

As we mentioned before, the Hadoop audit log cannot record the operator and there might be CSP employees who are authorized to modify the Hadoop logs and the Fsimage, so both the Hadoop logs and Fsimage may be contaminated.

像之前讲到的，由于Hadoop的审计日志无法记录操作者，并且有CSP工作人员被授权修改Hadoop日志和Fsimage的可能，这将可能会导致Hadoop日志和Fsimage被污染。

（因为审计日志无法记录操作者，因此工作人员的修改操作导致日志被污染。）

In application-level attacks, criminals access data through Hadoop commands, e.g. some Hadoop shell commands for file operations, such as CopyToLocal, so we can filter out this type of record from the HProgger logs.

在应用层面的攻击中，罪犯者通过Hadoop命令访问数据，比如一些用于文件操作的Hadoop shell命令，CopyToLocal就为其中之一，因此我们可以从HProgger日志中筛选出这种类型（命令）的记录。

（e.g. 与such as 均为例如，但是两者之间存在从属关系。）

1. 主句从句之间合并翻译。

However, if attackers tampered with the Hadoop logs from the data node or the name node, how do we investigate such a case?

但是，如果在攻击者篡改了数据节点或名称节点的Hadoop日志的情况下我们该如何调查？

（将从句整合进主句）

1. 主被动转换

In order to test our method, a small Hadoop cluster using Virtual Box including one master node and 15 slave nodes is set up.

为了测试我们提出的方案，在Virtual Box虚拟机上安装了有一个主节点和15个从节点的小型Hadoop群。（被动句转换为主动）