Apache Hadoop Security: Keeping out the Lookie Loos

Owen O'Malley
owen@yahoo-inc.com
Yahoo's Hadoop Team

ApacheCon 2010

Who am I

- An architect working on Hadoop full time
  - Mainly focused on MapReduce
- Tech-lead on adding security to Hadoop
- Before Hadoop – Yahoo Search WebMap
- Before Yahoo – NASA, Sun
- PhD from UC Irvine

What is Hadoop?

- A framework for storing and processing big data on lots of commodity machines.
  - Up to 4,000 machines
  - Up to 20 PB
- Open Source Apache project
- High reliability done in software
  - Automated failover for data and computation
- Implemented in Java

What is Hadoop?

- HDFS – Distributed File System
  - Combines cluster's local storage into a single namespace.
  - All data is replicated to multiple machines.
  - Provides locality information to clients
- MapReduce
  - Batch computation framework
  - Jobs divided into tasks. Tasks re-executed on failure
  - User code wrapped around a distributed sort
  - Optimizes for data locality of input

Hadoop as an Internal Service

- Yahoo! uses Hadoop a lot
  - 38,000 computers in ~20 Hadoop clusters.
  - Clusters run as shared service for yahoos.
  - Hundreds of yahoos use Hadoop every month
  - Run more than 1 million jobs every month
- All Hadoop clusters behind firewalls
- Clusters only used by yahoos
  - Why does Hadoop security matter?

Problem

- Different yahoos need different data.
  - PII versus financial
  - Need assurance that only the right people can see data.
  - Need to log who looked at the data.
- Yahoo! has more yahoos than clusters.
  - Requires isolation or trust.
  - Security improves ability to share clusters between groups

History

- Originally, Hadoop had no security.
  - Only used by small teams who trusted each other
  - On data all of them had access to
- Users and groups were added in 0.16
  - Prevented accidents, but easy to bypass
  - hadoop fs –Dhadoop.job.ugi=joe –rmr /user/joe
- We needed more…

Why is Security Hard?

- Hadoop is ***Distributed***
  - runs on a **cluster** of computers.
- Can't determine the user on client computer.
  - OS doesn't tell you, must be done by application
- Client needs to authenticate to each computer
  - Using password database like SQL servers won't work
- Client needs to protect against fake servers

Need Delegation

- Not just client-server, the servers access other services on behalf of others.
- MapReduce need to have user's permissions
  - Even if the user logs out
- MapReduce jobs need to:
  - Get and keep the necessary credentials
  - Renew them while the job is running
  - Destroy them when the job finishes

Another Rejected Solution

- One possible solution is to run client and server as root.
  - This is what NFS does.
  - Both client and server use restricted ports.
- Java has good sandboxing, but terrible support for running as root.
- Would require that the client code go through kernel.

Solution

- Prevent unauthorized HDFS access
  - All HDFS clients **must** be authenticated.
  - Including tasks running as part of MapReduce jobs
  - And jobs submitted through Oozie.
- Users must also authenticate servers
  - Otherwise fraudulent servers could steal credentials
- Integrate Hadoop with Kerberos
  - Provides well tested open source distributed authentication system.

Requirements

- Security must be optional.
  - Not all clusters are shared between users.
- Hadoop must not prompt for passwords
  - Makes it easy to make trojan horse versions.
  - Must have single sign on.
- Must handle the launch of a MapReduce job on 4,000 Nodes

Security Definitions

- **Authentication** – Determining the user
  - Hadoop 0.20 completely trusted the user
    - Sent user and groups over wire
  - We need it on both RPC and Web UI.
- **Authorization** – What can that user do?
  - HDFS had owners and permissions since 0.16.
  - Map/Reduce had minimal authorization in 0.20.
- **Auditing** – Who did that?

Authentication

- Changes low-level transport
- RPC authentication using SASL
  - Kerberos (GSSAPI)
  - Token
  - Simple
- Browser HTTP secured via plugin
- Configurable translation from Kerberos principals to user names

Authorization

- HDFS
  - Command line and semantics unchanged
- MapReduce added Access Control Lists
  - Lists of users and groups that have access.
  - mapreduce.job.acl-view-job – view job
  - mapreduce.job.acl-modify-job – kill or modify job
- Code for determining group membership is pluggable.
  - Checked on the masters.

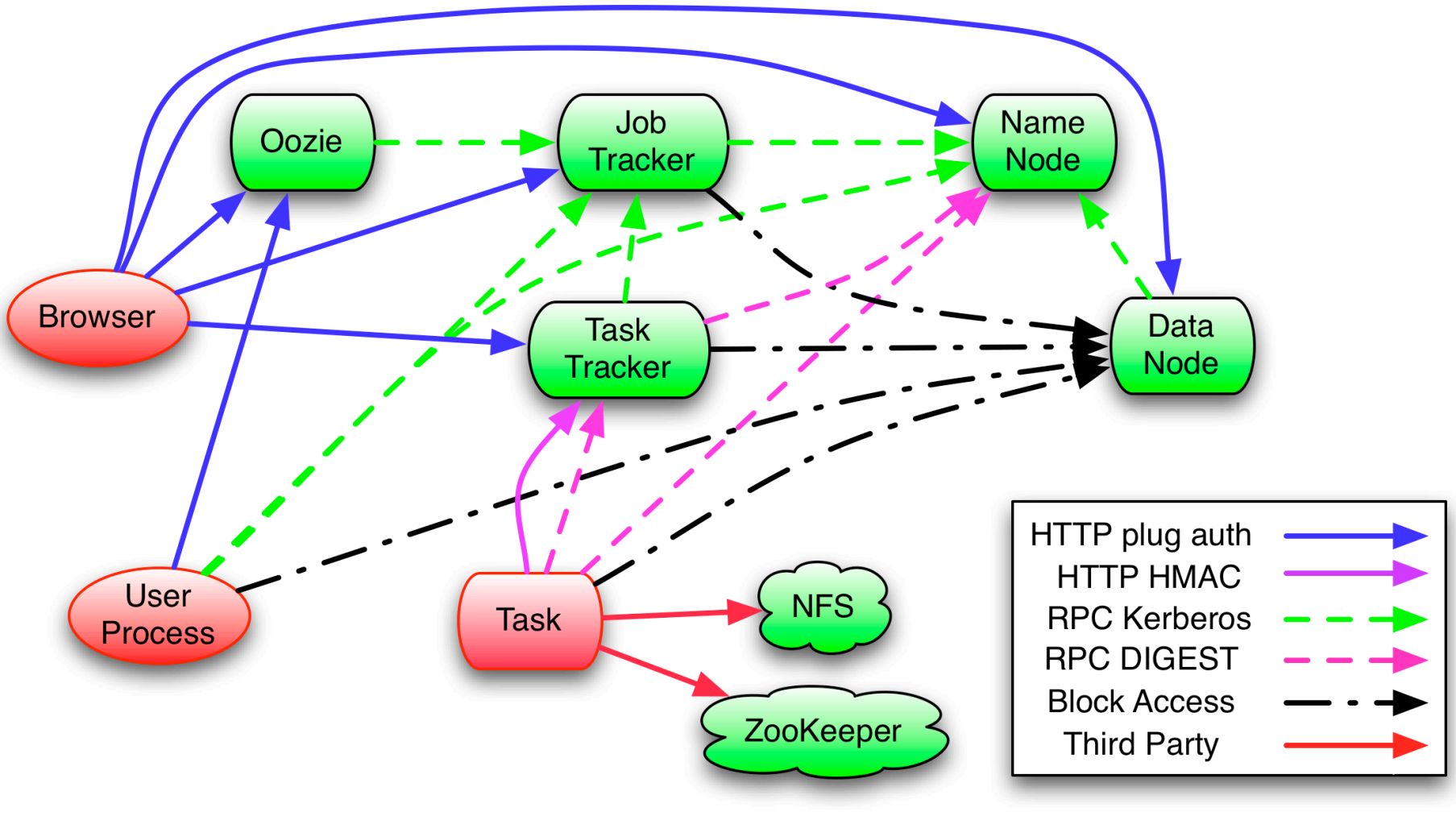
Auditing

- HDFS can track access to files
- MapReduce can track who ran each job
- Provides fine grain logs of who did what
- Configured by log4j
  - org.apache.hadoop.fs.FSNameSystem.audit
  - org.apache.hadoop.mapred.AuditLogger
- With strong authentication, logs provide audit trails

Delegation Tokens

- To prevent authentication flood at the start of a job, NameNode creates delegation tokens.
- Allows user to authenticate once and pass credentials to all tasks of a job.
- JobTracker automatically renews tokens while job is running.
  - Max lifetime of delegation tokens is 7 days.
- Cancels tokens when job finishes.

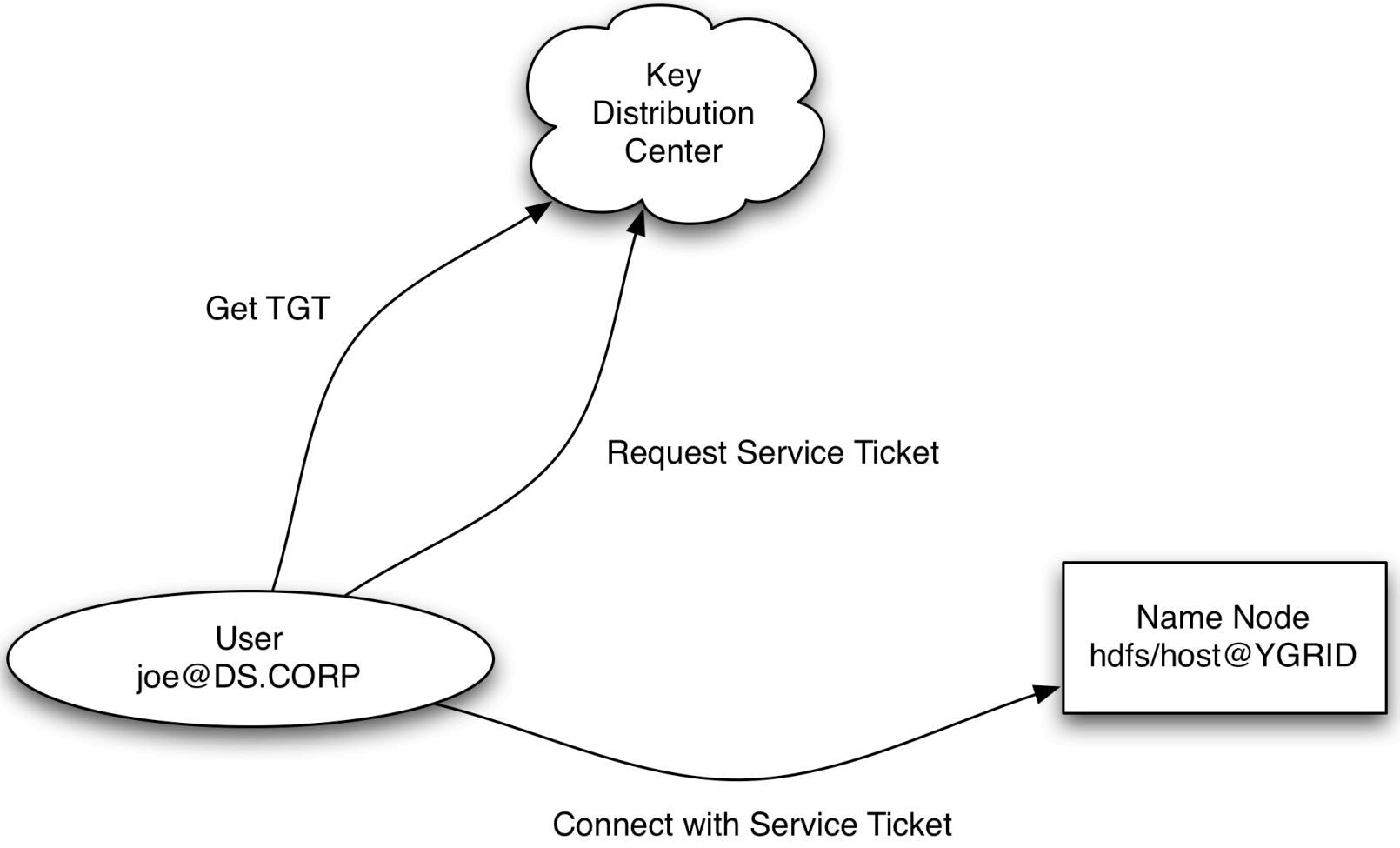
Primary Communication Paths



Kerberos and Single Sign-on

- Kerberos allows user to sign in once
  - Obtains Ticket Granting Ticket (TGT)
    - kinit – get a new Kerberos ticket
    - klist – list your Kerberos tickets
    - kdestroy – destroy your Kerberos ticket
    - TGT's last for 10 hours, renewable for 7 days by default
  - Once you have a TGT, Hadoop commands just work
    - hadoop fs –ls /
    - hadoop jar wordcount.jar in-dir out-dir

Kerberos Dataflow



API Changes

- Very Minimal API Changes
- MapReduce added secret credentials
  - Available from JobConf and JobContext
  - Never displayed via Web UI
- Automatically get tokens for HDFS
  - Primary HDFS, File{In,Out}putFormat, and DistCp
  - Can set mapreduce.job.hdfs-servers

Task Isolation

- Tasks now run as the user.
  - Via a small setuid program
  - Can't signal other user's tasks or TaskTracker
  - Can't read other tasks jobconf, files, outputs, or logs
- Distributed cache
  - Public files shared between jobs and users
  - Private files shared between jobs

Web UIs

- Hadoop relies on the Web UIs.
  - These need to be authenticated also…
- Web UI authentication is pluggable.
  - Yahoo uses an internal package
  - We have written a very simple static auth plug-in
  - SPNEGO plugin being developed
- All servlets enforce permissions.

Upgrading to Security

- Need a KDC with all of the user accounts.
- Need service principals for all of the servers.
  - hdfs/_HOST and mapred/_HOST
  - _HOST is replaced by the hostname.
  - Put the principals for each host into a keytab.
- Need user accounts on all of the slaves
- If you use the default group mapping, you need user accounts on the masters too.

Upgrading Tips

- Need to install the support for stronger encryption for Java
  - http://bit.ly/dhM6qW
- Can turn on Kerberos debugging
  - -Dsun.security.krb5.debug=true
- Hadoop will complain and fail if the permissions are incorrectly set.

Mapping to Usernames

- Kerberos principals need to be mapped to usernames
  - omalley@APACHE.ORG -> omalley
  - mapred/jt1.apache.org@APACHE.ORG -> mapred
- Default rule maps all names in default realm to first component.
- Operator can define translation.

Higher Level Services

- Oozie is a workflow engine where users submit jobs to run later
- Can be compared to
  - Make – run job when data is available
  - Cron – run job every 15 minutes
- Must run jobs as individual users.
- Can't store delegation tokens, since the job may be delayed indefinitely.

Proxy-Users

- Oozie (and other services) run as headless user.
- Configure HDFS and MapReduce with the oozie user as a proxy:
  - Group of users that the proxy can impersonate
  - Which hosts they can impersonate from
- Provides control without over burdening operations.

Out of Scope

- Encryption
  - RPC transport
  - Block transport protocol
  - On disk
- File Access Control Lists
  - Still use Unix-style owner, group, other permissions
- Non-Kerberos Authentication
  - Much easier now that framework is available

Schedule

- The security team worked hard to get security added to Hadoop on schedule.
  - Roughly 6 months of calendar time.
- Security Development team:
  - Devaraj Das, Ravi Gummadi, Jakob Homan, Owen O'Malley, Jitendra Pandey, Boris Shkolnik, Vinod Vavilapalli, Kan Zhang
- Currently on production clusters

Questions?

- Questions should be sent to:
  - common/hdfs/mapreduce-user@hadoop.apache.org
- Security holes should be sent to:
  - security@hadoop.apache.org
- Available from
  - http://developer.yahoo.com/hadoop/distribution/
  - Also a VM with Hadoop cluster with security
- Thanks!