Strata+ Hadoop WORLD

PRESENTED BY

O'REILLY cloudera

strataconf.com

#StrataHadoop

Data Security in Hadoop

Speaker: Ajit Gaddam

Hadoop and Bigdata represent a greenfield opportunity for security practitioners. It provides a chance to get ahead of the curve, test and deploy your tools, processes, patterns, and techniques before big data becomes a big problem.

We will walk through control frameworks developed to support data security, compliance, cryptographic protection, and effective risk management for sensitive data.

About Speaker

- Day job as Chief Security Architect @ VISA Inc.
- Before – Enterprise Architect at Progressive. Deployed a $xxM Enterprise Data Protection program
- Co-founder of 2 startups
- SABSA, CISSP, GSEC, GPEN, TOGAF, SANS
- Ping me @ajitgaddam if you want a copy of this deck

Disclaimer

If I shared something here that you find helpful as you are trying to secure your enterprise, that's wonderful. But when push comes to shove, this is my personal presentation. The views expressed in this presentation are mine alone and not those of my current employer.

- Q: Heh. Did you get a talking to?

  A: No, I haven't. Hopefully I never will.

- Q: Why are you doing this now?

  A: Just in case. If I say something stupid in the future, it's better to be able to point out that the stupidity is mine, and mine alone. My stupidity! You can't have it!

Agenda

- What is Hadoop Security & why do we need it?
- Hadoop Data Security Framework – 5 pillars:
  - Data Management
  - Identity & Access Management
  - Data Protection at Rest
  - Data Protection in Transit
  - Data Leakage/Exfiltration Prevention
- Successes, Failures, and Best Practices

Three Reasons for Securing Hadoop

1

Hadoop contains **Sensitive Data**

- Various business units and dev groups quickly go from a POC to deploying a production cluster, and with it moving petabytes of data.
- This data contains sensitive cardholder and other customer or corporate data that must be protected

2

Hadoop is subject to **regulatory adherence**

- With #1 comes compliance to PCI DSS, FISMA, HIPAA, EU laws to protect PII
- Hadoop and bigdata represent a greenfield opportunity and provides a chance to get ahead of the curve and deploy your tools, processes, and techniques before big data becomes a big problem.

3

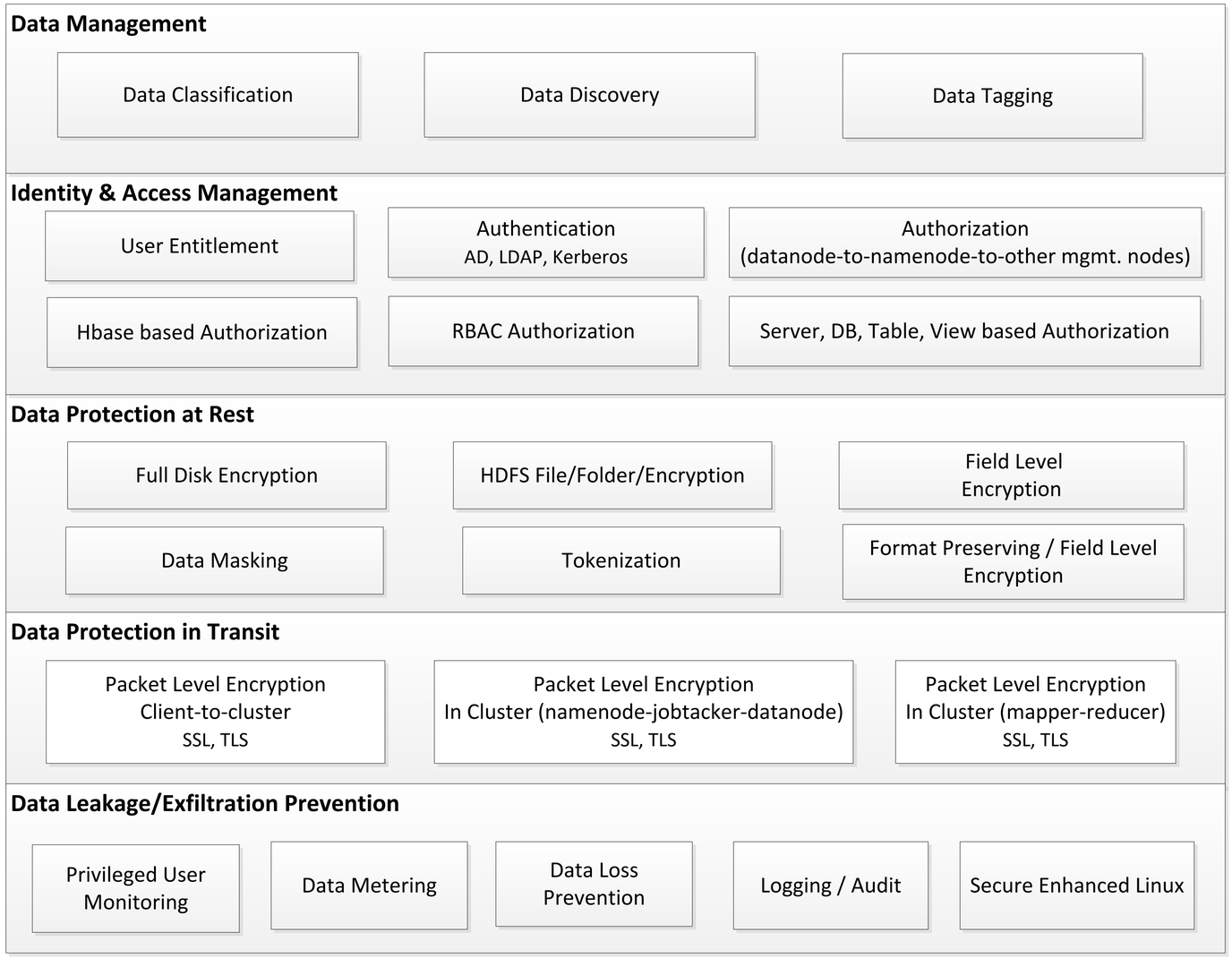
Hadoop security can **enable your business**

- Before Hadoop usage was broad and possibly restricted to non-sensitive data.
- With data security in Hadoop, you can allow for sensitive workloads on restricted datasets

Current State of Hadoop Security Solutions

- Traditional RDBMS platforms have undergone many decades of security evaluations and assessments. Hadoop Security solutions do not have the same security rigor
- Hadoop security is fragmented. OSS vendors and distribution vendors (e.g. Cloudera, Hortonworks, MapR) in many cases force fit security into Apache Hadoop framework. This is changing but much more is needed.
- No standardization or portability of data security between different OSS projects and vendors, even when they implement the same feature for the same Hadoop component.
- RBAC policy files, MR ACLs, are frequently configured via cleartext files – editable by root and priv accounts
- Hive is vulnerable to SQL injection attacks for example
- The CVE database only shows 3 reported and fixed Hadoop vulnerabilities over past 3 years. Software, even Hadoop, is far from perfect

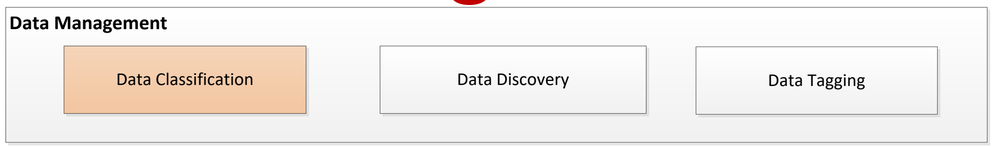
Hadoop Data Security Framework



5 pillars of Hadoop Data Security:

- Data Management
- Identity & Access Management
- Data Protection at Rest
- Data Protection in Transit
- Data Leakage/Exfiltration Prevention

Data Management: Data Classification & Prioritization



1. Work with your legal, privacy office, IP, finance and determine all **distinct** data fields
2. Do a control assessment exercise
   - Breach reporting requirements when SSN is breached
   - Location of data (e.g. in restricted zone, exposed to internet access)
   - Number of users/systems with access
   - Current security controls (e.g. can it be protected cryptographically etc.)
3. Determine value to an attacker
   - Easy to resell on the black market
   - Value from Intellectual Property (e.g. a nation state looking for defense blueprints)
4. Impact due to fines (e.g. breach) – hard
5. Impact from other soft categories (e.g. customer impact, brand loss etc.)

Data Management: Data Discovery (1/3)

Data Discovery is ground zero for all data protection activities.

1. Search for, identity and classify on presence of your sensitive data.
2. Define and validate the data structure and schema. This is all useful prep work for data protection activities later
3. Collect metrics (e.g. volume counts, unique counts etc.)
   - ❑ For example, if a file has 1M records but it is duplicate of a single person, very useful for compliance but more importantly risk management
4. Share this insight with your Data Science teams for them to build threat models, profiles which will be useful in data exfiltration prevention scenarios.
5. Provide your executives & stakeholders reports & sensitive data heatmaps

Data Management: Data Discovery (2/3)

Challenges

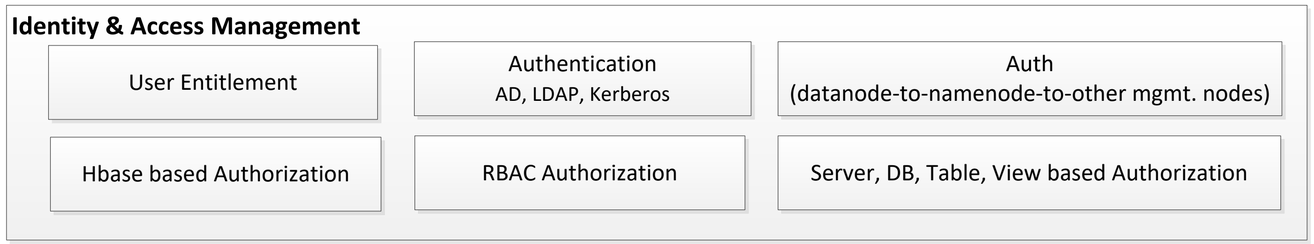
1. Data structure issues in your Hadoop cluster
   - ❑ Lack of structure definition for data (e.g. 1 sequence file with SSN in 1st column and another file with SSN in column 134)
   - ❑ Lack of consistency on column delimiters and other edge cases (e.g. some space, none, ^G delimiter; directory names starting with a period or underscore)
2. Compression of data
   - ❑ Ensure you have a balanced algorithm. For example, don't select an algorithm that has the slowest I/O but max compression. For discovery, need to account for x times additional processing, maps/tasks
3. Infrastructure
   - ❑ Number of nodes available aka number of mappers available to execute discovery operations
4. Results reported
   - ❑ High false positive rate (e.g. a 9 digit number (zipcode or a SSN?)). Imagine a lookup table and comparing against billions of records

Data Management: Data Discovery (3/3)

Recommendations / Guidance:

1. Determine whether structured scans or unstructured scans make more sense in your environment
   - ❑ Run unstructured scans on a small subset (e.g. a directory)
   - ❑ Determine structure and schema of data
   - ❑ Run targeted structured scans and report on it
2. Vendor technologies are still improving
   - ❑ Conditional search (e.g. only report on date of birth if a person's name is found or PAN + CVV or PAN +zip)
   - ❑ False positives, lookup tables with whitelisting; unique counts (in-mem DB solution?)
   - ❑ Once data has been encrypted for example, then what?
3. Opportunity to improve your data architecture
   - ❑ Enforce consistent data architecture & structure before data load into HDFS
   - ❑ Plan for what if, once data has been protected (e.g. encrypted, tokenized etc.)

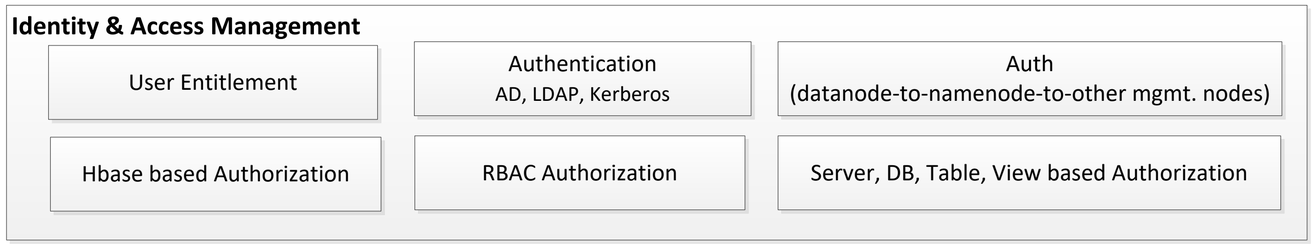
Identity & Access Management (1/3)



Key Observations:

- **User Entitlement:** Provide users access to data; centrally manage policies; tie policy to data and not access method
- **RBAC Authorization:** Deliver fine-grained authorization; manage data access by role (and not user); relationships between users & roles through groups - leverage AD/LDAP group membership and enforce rules across all data access paths
- **Authorization:** Leverage Attribute based access control and protect data based on tags that move with the data through lineage; permissions decisions can leverage the user, environment (e.g. location), and data attributes.

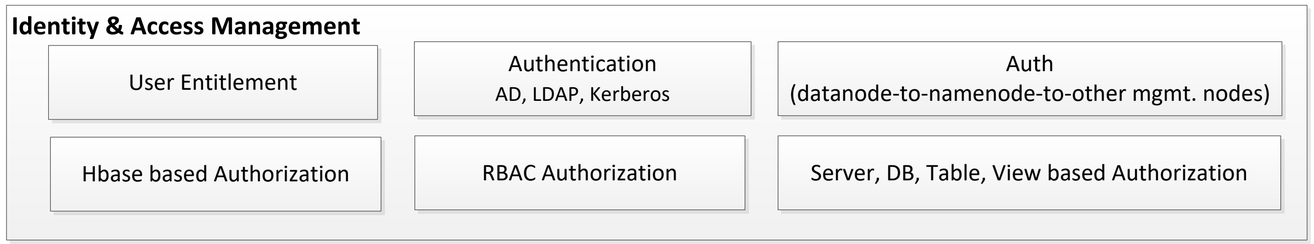
Identity & Access Management (2/3)



Guidance / Recommendations

- Use Kerberos – to validate nodes & client applications before admission into the cluster
- Common Enterprise security deployments like 2FA for admin access and segregation of duties between the different folks with access
- Validate nodes during deployment using virtualization mgmt. or using Chef/puppet
- Validate application requests for MapReduce (MR) and similar functions
- Do not allow your end users to connect to data nodes, but to name nodes only (Apache Knox can help here)
- Apache Sentry is a good platform to leverage to manage RBAC; it can leverage AD to determine user's group assignments automatically and keeping its permissions upto date.

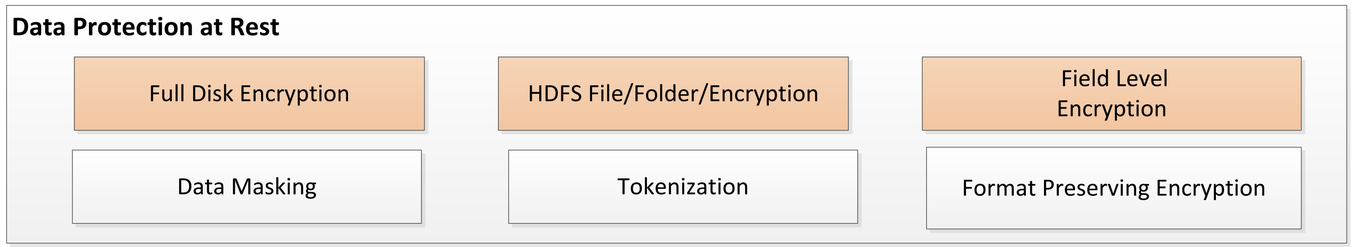
Identity & Access Management (3/3) - Keytab



Recommendations / Guidance:

- #1 guidance – think about your keytab files in a Kerberos cluster.
  - The keytab file (short for key table) contains the principle (service key) used by a service to authenticate to the KDC
  - Threat: It is possible to impersonate root or an application ID by copying its keytab
  - Keytab files are analogous to a user/application password.
    - Store the keytab files on local disk and make them only readable by the root user
    - Have the highest level of monitoring on these files. Never send it over an insecure network
    - Ensure that root account is limited (actions of those people, when authenticated as Root, cannot be tied back to those individuals)

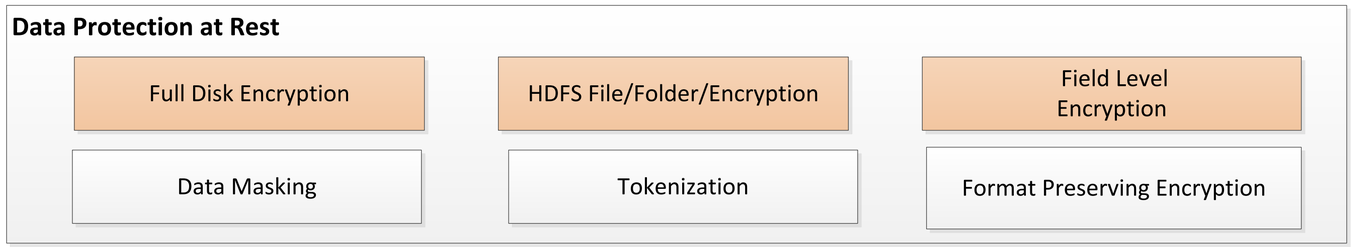
Data Protection at Rest (1/2) – Encryption



Key Observations

- Full Disk Encryption can also be OS native encryption such as dm-crypt
- Protection against different threat models (data center, cloud, external, internal ..)
  - Need to protect all data (structured, unstructured, metadata, files)
  - Many types of users; many types of data; different sensitivity levels; diff tools
- Understand the downstream impact when encrypting (downstream app rules; key exchange; scenario - data exiting Hadoop into a reporting warehouse that is encrypted and keys stored in a different HSM)
- FPE is still evolving (no standard yet; NIST has FFX, BPS, VPFE as finalists)
- Field level encryption can provide security granularity and audit tracking, but comes at expense of manual intervention to determine sensitive fields and where and how to enable authorized decryption

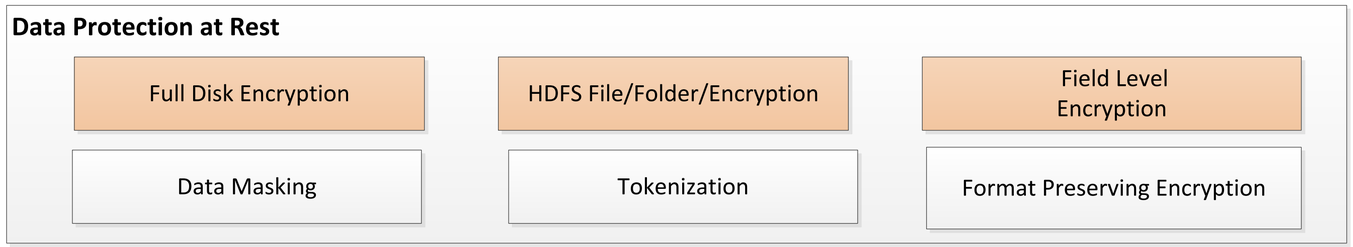
Data Protection at Rest (2/2) – Encryption



Guidance / Recommendations

- Use file/OS level encryption to protect against privileged users or apps with direct access to files. Covers some threat models against unknown/unprotected data
- Use a central key mgmt. server to manage the crypto keys. Also separates keys from data
- Go for standards based technologies where possible (e.g. using KMIP vs. proprietary key mgmt. technologies to avoid vendor lock-in)
- Leverage encryption at hardware level where possible – e.g. AES NI with Intel Optimization
- For encryption, leverage native APIs if possible over WS calls for better performance (crypto done in app memory space vs. WS where it can be chatty; data transported to crypto appliance; encrypted then returned. Also understand what your backend network backbone is (Gigabit backbone vs. 10Gig for example)
- Start with transparent encryption and move up the maturity chain with app-level encryption for some datasets (new); protects against PB of data currently in Hadoop today

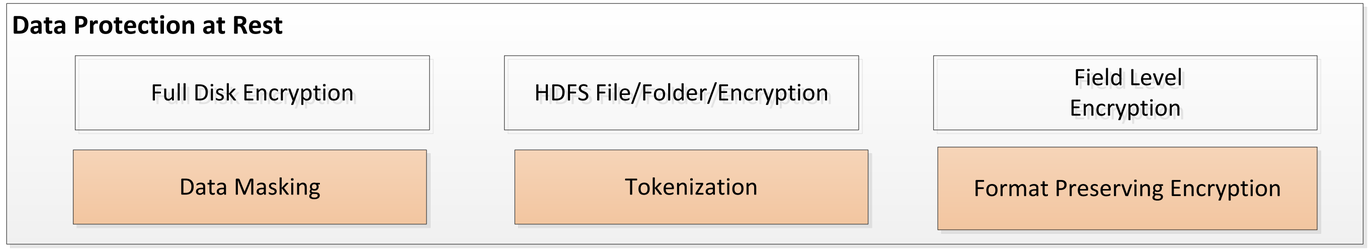
Data Protection at Rest – Key Management



Recommendations / Guidance:

- Always have an external key manager (separate keys from data). Be able to call from inside the cluster to retrieve keys
- Remember, it is not just crypto keys. There are SSL certs, SSH keys, passwords, Kerberos keytab files also.
- Integrate with HSMs from the core players at a minimum (Thales, SafeNet, RSA)
- Ensure that you are pursuing open standards like KMIP for key exchange and not locked in to a vendor proprietary key management implementation
- Account for integration on key transitions, rollover, archival, and backup
- Support individual crypto keys for different types of data (e.g. PCI, PII)

Data Protection at Rest – Tokenization



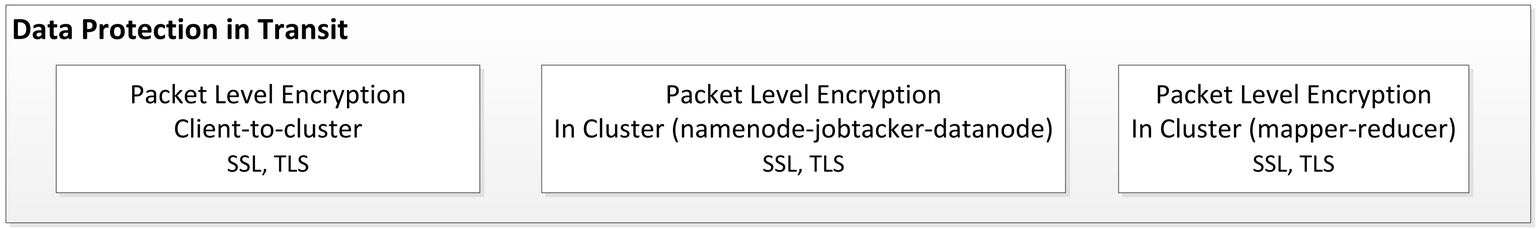
Guidance / Recommendations

- Understand your business usecase of data processing; (e.g. does business processing only need first 6 digits of the card?)
- Explore different formats (fixed, numeric, mixed)
- Tokenize numeric data; data needed in search; data used as a primary key
- Plan for discovery post tokenization (how do you differentiate a format preserving token from original data?)
- Generate bulk tokens for usage in your non-prod environment

Challenges:

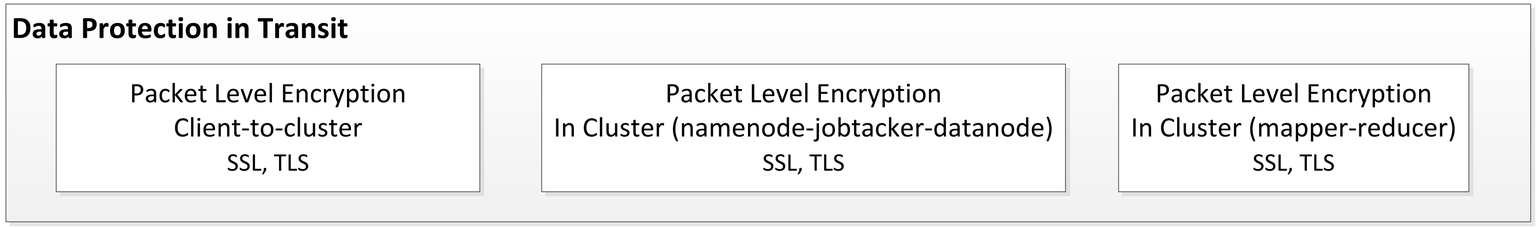
- Watch out for Format Preserving Encryption options (stick with NIST guidance)
- Performance wise – remember, tokenizing 9 digits could result in 10-30% lower performance. Vendors suggest to restrict the token to 7 digits to reduce the number of table lookups.(e.g. first 6 + last 4 of the 16 digit card in the clear)

Data Protection in Transit (1/2)



- Maturity curve – go left to right in the control framework above
- Untrusted mappers can be altered to snoop on requests, alter MapReduce scripts, or alter results.
- Hadoop 2.0 is first a resource mgmt. framework. Although app instances will be launched as close to data as possible, there is a possibility that the scheduler cannot find resources next to the data and may need to read data over the network
- With large data sets, it is nearly impossible to identify malicious mappers that may create significant damage, especially for scientific and financial computations.
- Do Endpoint validation - Ingestion of data from trusted and mutually authenticated sources through secure mechanisms to mitigate the threats from malicious entry and compromise of transmission

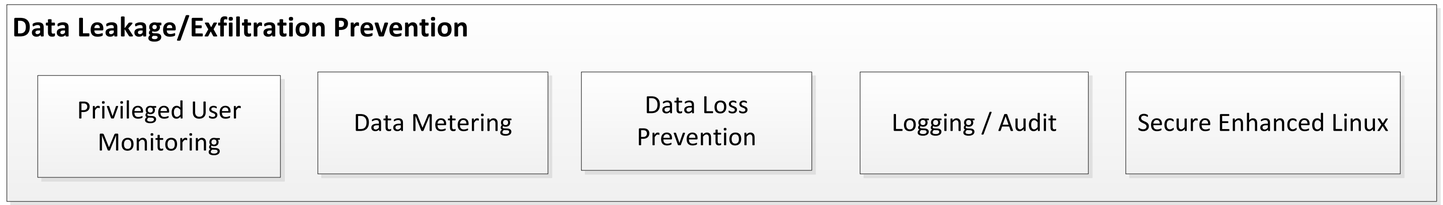
Data Protection in Transit (2/2)



Guidance/Recommendations

- Implement secure communication between components (e.g. consoles to servers)
- Use TLS protocol to authenticate and ensure privacy of communications between nodes, name servers, and applications
- Allow your admins to configure and enable encrypted shuffle and TLS/https for HDFS, MapReduce, YARN, HBase UIs etc.

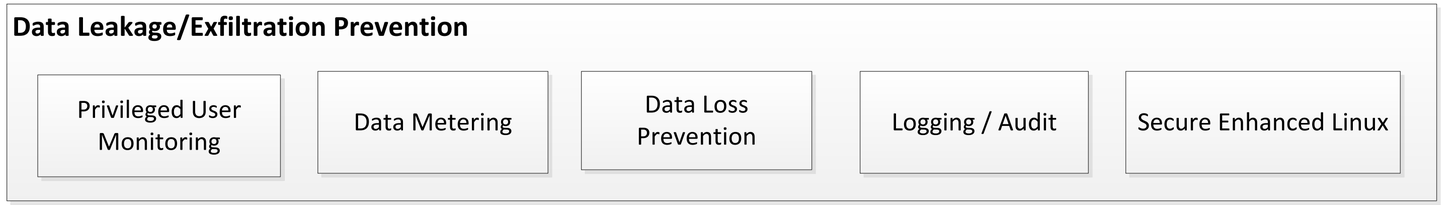
Data Leakage / Exfiltration Prevention (1/2)



Recommendation / Guidance:

- Enable SELinux on your Hadoop Cluster (start with permissive and eventually go to enforcing mode).
  - cat /etc/selinux/config
    - # This file controls the state of SELinux on the system.
    - # SELINUX= can take one of these three values:
    - # enforcing - SELinux security policy is enforced.
    - # permissive - SELinux prints warnings instead of enforcing.
    - # disabled - SELinux is fully disabled.
- Build user models; use discovery data, user/application monitoring data and begin to enforce data metering policies
- Protect against rogue nodes. Leverage pre-shared secrets or certificates and prevent addition of unauthorized cluster nodes

Data Leakage / Exfiltration Prevention – Logging / Audit (2/2)



Recommendations/ Guidance:

- Log transactions, anomalies, and administrative activity through logging tools that leverage the big data cluster itself; log for addition/deletion of data and management node states
- Building data metering capabilities to detect anomalies
- Using a separate SIEM infrastructure away from the Data lakes to ensure integrity of the logs and preventing internal attacks on them
- Data metering capability – build models for your users/app accounts – something that ran 4 jobs a day on weekday is running 10 jobs/weekend – why? alert

Recap

1. Understand your data landscape; don’t be afraid to pick different vendors for different levels of data sensitivity based on your assessment of their security controls
2. Everything starts with Data Discovery; build data hygiene and protection plan from that
3. IAM – use Kerberos and protect keytab files
4. Encryption – start with transparent encryption first; mature to app level and tokenization for data at rest; ensure key mgmt. leverages standards; don’t have vendor lock-in; see if AES-NI can be leveraged for crypto performance
5. Leverage an external key manager; integrate with HSMs (Java Key Store can be short-term); use standards like KMIP for key exchange
6. Enable TLS – start with client to cluster first; mature to Hadoop nodes (namenode-jobtracker-datanode) and then to the MR jobs
7. Data Exfiltration is the final piece of the puzzle; think kill-chain; log & monitor; follow-the-data and build models for your users/apps

Q & A Session