COMMUNITY
THE ASF CONFERENCE
CODE

Insecure By Design

How Not to Build Your Next Data Pipeline

David Handermann, PMC Member of Apache NiFi, Software Engineer at Datavolo

Introduction – David Handermann

PMC Member
nifi.apache.org

Software Engineer
datavolo.io

Software Development Blog
exceptionfactory.com

Summary

Insecure by design: **How does it happen?**

1. Not **Encrypting** Communications
2. Not **Authenticating** Peers
3. Not **Validating** Inputs
4. Not **Enumerating** Outputs
5. Not **Expecting** Errors

Not Encrypting Communications

TLS is hard

Let's go shopping...

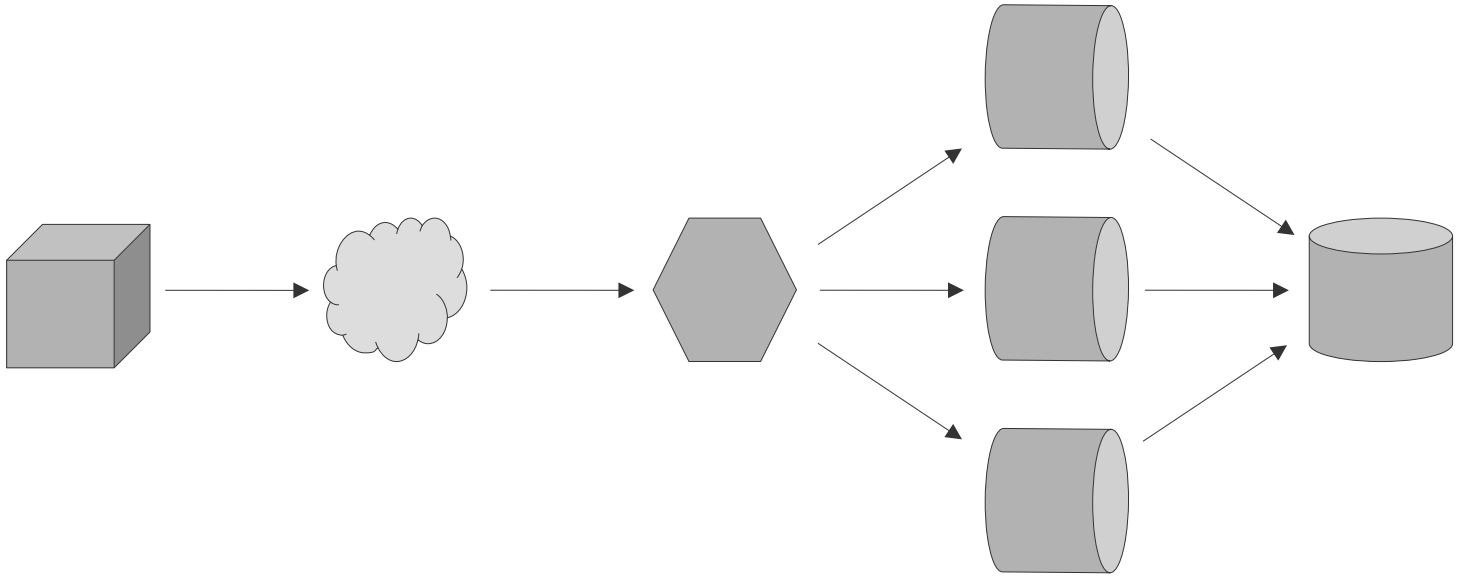
Not Encrypting Communications

TLS Issues: **Unsupported or Misconfigured**

- **HTTP** instead of HTTPS
- Excessive trust in **TLS Termination Gateways**
- **Log Collection** without TLS
- **Database Connections** without TLS
- **Caching** and **Coordination** without TLS

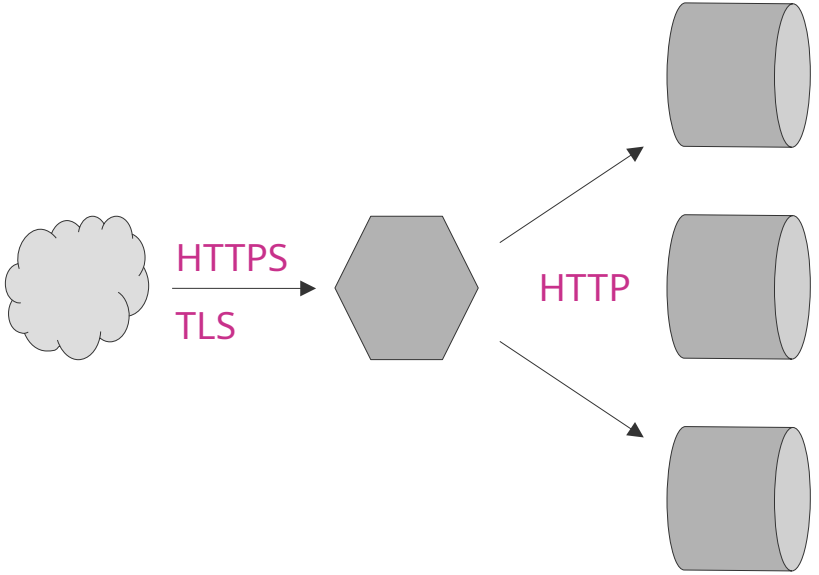
Not Encrypting Communications

TLS Termination: **Where is the boundary?**



Not Encrypting Communications

TLS Gateways: **Not Quite Zero Trust**



Not Encrypting Communications

TLS Configuration: **Is it actually secure?**

- Which Version? **TLS 1.0** and **1.1** are **deprecated**
  - TLS 1.2 supports insecure cipher suites
- Which Cipher Suites? Avoid **unauthenticated** ciphers
  - TLS 1.3 requires **AES-GCM** or **ChaCha20-Poly1305**

Not Encrypting Communications

SFTP is easier

How is it configured?

Not Encrypting Communications

SFTP Configuration: **Which algorithms allowed?**

- Which Cipher Algorithms? Many **legacy** algorithms
  - Prefer **AES-GCM** and **ChaCha20-Poly1305**
- Which Key Exchange Algorithms?
  - Prefer **ssh-ed25519** or **rsa-sha2-256**
- Which Message Authentication Code Algorithms?
  - **MD5** and **SHA-1** should be disabled

Not Authenticating Peers

Are these the droids you're looking for?

Not Authenticating Peers

Pipeline Authentication: **What could go wrong?**

- **Trusted** Networks
- **Custom** Authorities with Mutual TLS
- **Personal** Usernames and Passwords
- **User** Access Tokens
- **Shared** Service Accounts

Not Authenticating Peers

Trusted Networks: **What is the blast radius?**

- What if the **gateway** is misconfigured?
- What if the **processing system** is manipulated?
- Does the **storage service** permit elevated access?

Not Authenticating Peers

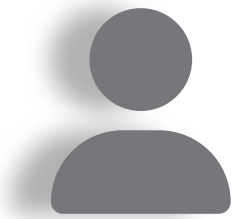
Mutual TLS: **Bidirectional Authentication**

- **Certificate Authorities** approve or deny peers
- **Root** and **intermediate** authorities issue certificates
- **Organization** authority or **service** authority?
- What is the **identification strategy?**
  - Subject Principal or Alternative Names

Not Authenticating Peers

User Credentials: **Not so service-oriented**

- **Ambiguous attribution**: user or machine?
- **Unintended** access privileges
- **Single** point of responsibility



Not Authenticating Peers

Shared Credentials: **Service access surprises**

- Unclear system boundaries
- Credential rotation **chain reactions**
- **Enterprise issues** waiting to happen

Not Validating Inputs

Validation
is a
Security Concern

Not Validating Inputs

CWE-400: Uncontrolled Resource Consumption

Denial of Service

from

Resource Exhaustion

Not Validating Inputs

Basic Validation Rules

1. **Size**
2. **Shape**
3. **Semantics**

Not Validating Inputs

Size Validation: **What is too big to process?**

- Unanticipated **Data Rates**
- Unexpected **File Sizes**
- Unsupportable **Field Lengths**
- Unconstrained **Compression Ratios**

Not Validating Inputs

OutOfMemoryError: **Java heap space**

- Just increase maximum heap size?
- Just increase system memory?
- Just hope it does not recur?

Not Validating Inputs

Size Validation: **How big is too big?**

- A little validation goes a long way
- Apache NiFi **RouteOnAttribute** Processor
  - Property: `size-exceeded`
  - Value: `${fileSize:gt(1048576)}`

Not Validating Inputs

Format Validation: **How big is that value?**

- Apache Avro **Limits** System Properties
- Apache POI **ZipSecureFile** Methods
- Jackson JSON **StreamReadConstraints**
- Jetty Maximum HTTP **Request Header Size**

Not Validating Inputs

Shape Validation: **Paint by Magic Numbers**

- Unexpected Errors from Unexpected Formats
- MIME Type Detection versus Expected Inputs
  - **Apache Tika** provides extensible detection
  - Apache NiFi **IdentifyMimeType** uses Apache Tika

Not Validating Inputs

Semantic Validation: **Some assembly required**

- Field Type Specifications
  - Is it an **int** or a **long**?
- Field Value Ranges
  - TCP port number **65536**?
- Field Requirements versus Extensibility
  - Just what is **optionally required**?

Not Validating Inputs

Semantic Validation: **Schema definitions**

- Common Formats
  - Apache Avro Schema
  - JSON Schema
  - XML Schema
- Lack of **versioning** leads to lack of **validation**

Not Enumerating Outputs

Do you know
where **your data** is going?

Not Enumerating Outputs

Data Transmission: **What and where?**

- How dynamic is the data?
- How flexible are the destinations?
- Who controls the routing?

Not Enumerating Outputs

Data-Driven Routing: **Field-based destination?**

```
{
  "id": 1,
  "action": "STARTED",
  "topic": "events",
  "uri": "https://events.local"
}
```

Not Enumerating Outputs

Data-Driven Routing: **Field-based flow**

- NiFi **EvaluateJsonPath** Processor
  - Property: **`destinationUri`**
  - Value: **`$.uri`**
- NiFi **InvokeHTTP** Processor
  - Property: **`URL`**
  - Value: **`${destinationUri}`**

Not Enumerating Outputs

Data-Driven Routing: **Parameterized flow**

- NiFi **EvaluateJsonPath** Processor
  - Property: **`topic`**
  - Value: **`$.topic`**
- NiFi **InvokeHTTP** Processor
  - Property: **`URL`**
  - Value: `https://events.local/`**`${topic}`**

Not Expecting Errors

No **plan** for **failure**
is
planning to **fail**

Not Expecting Errors

Ignoring Murphy's Law

- Terminated **failure relationships**
- Infinite **retries** without backoff strategies
- Unconstrained **logging**
- Missing or misconfigured **socket timeouts**
- **Custom code** with minimal verification

Not Expecting Errors

Failure Handling: **Availabilty and integrity**

- Ignoring errors leads to **data loss**
- Poor exception handling leads to **performance loss**
- Excessive logging **exhausts resources**

Not Expecting Errors

Socket Timeouts: **How long is too long?**

- Infinite timeouts lead to **blocked threads**
- Short timeouts lead to unexpected **closed streams**
- Uncoordinated timeouts and retries waste cycles

Not Expecting Errors

Custom Code: **Works on my machine?**

- **System.exit()** is not error handling
- **System properties** are not component configuration
- **Reading all bytes** is not stream processing

Conclusion

Security
is a
cross-cutting concern

Conclusion

Secure pipelines by design

1. **Encrypt** Communications **Correctly**
2. **Authenticate** Peers **Properly**
3. **Validate** Inputs **Extensively**
4. **Enumerate** Outputs **Completely**
5. **Expect** Errors **Comprehensively**

Conclusion

Stay connected

exceptionfactory.com