SDC 20

BY Developers FOR Developers

Storage Developer Conference
September 22-23, 2020

OS Level Encryption and Access Control for Superior Data Protection

Peter Scott, Principal Software Engineer
Rajesh Gupta, Sr. Development Manager

THALES

Data is prey

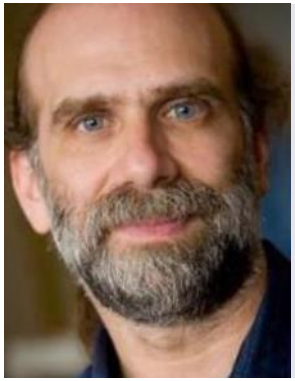
Threats to businesses

- Credential Theft
- Weak encryption
- Infection by malware and Ransomware
- Privilege Escalation

Risks

- Data Breach
- Data Exposures
- Data Exfiltration

Not all encryption is created equal



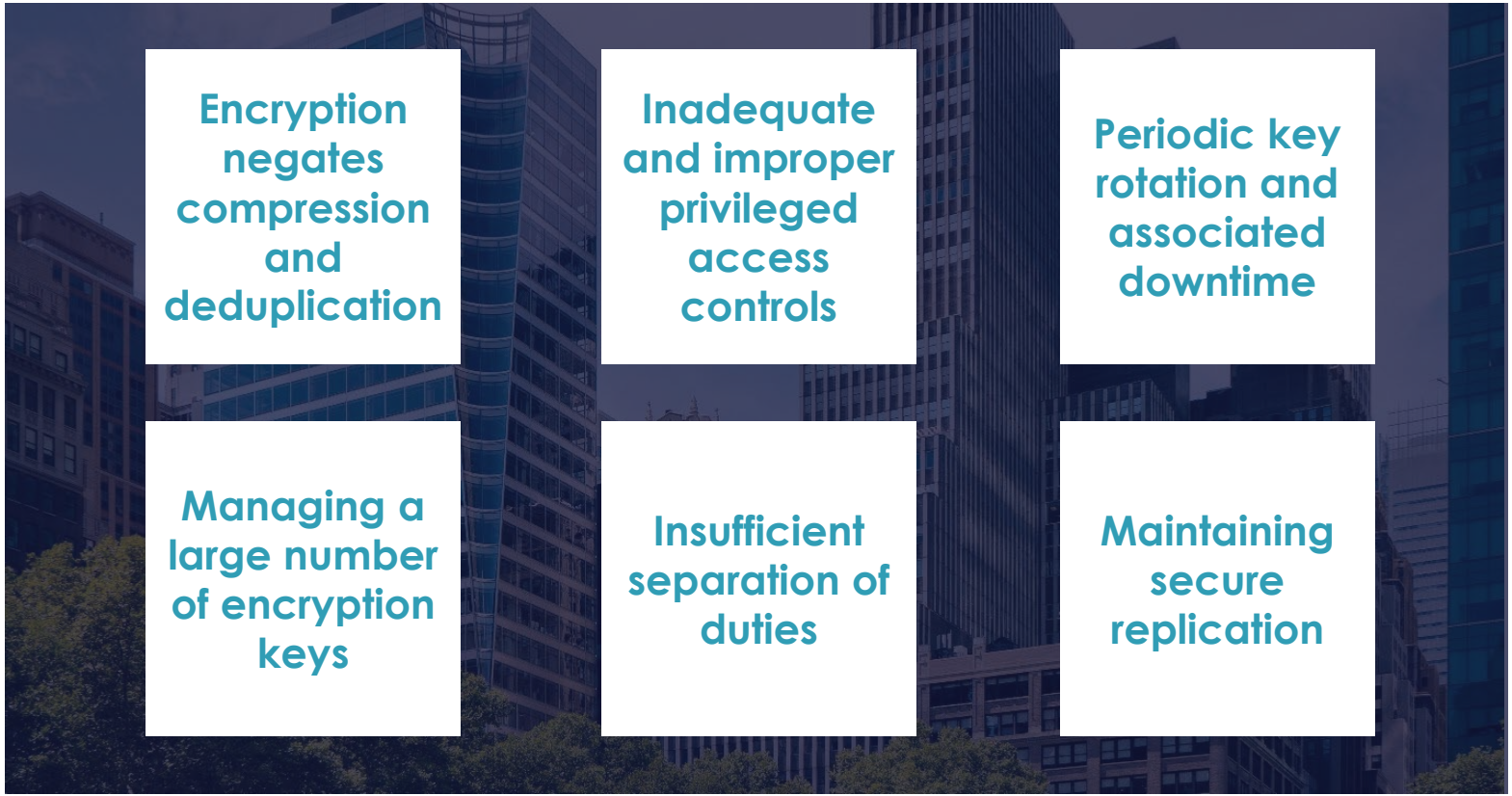
"There are two types of encryption: one that will prevent your sister from reading your diary and one that will prevent your government."

– Bruce Schneier

"Companies spend millions of dollars on firewalls and secure access devices, and it's money wasted because none of these measures address the weakest link in the security chain: the people who use, administer and operate computer systems"

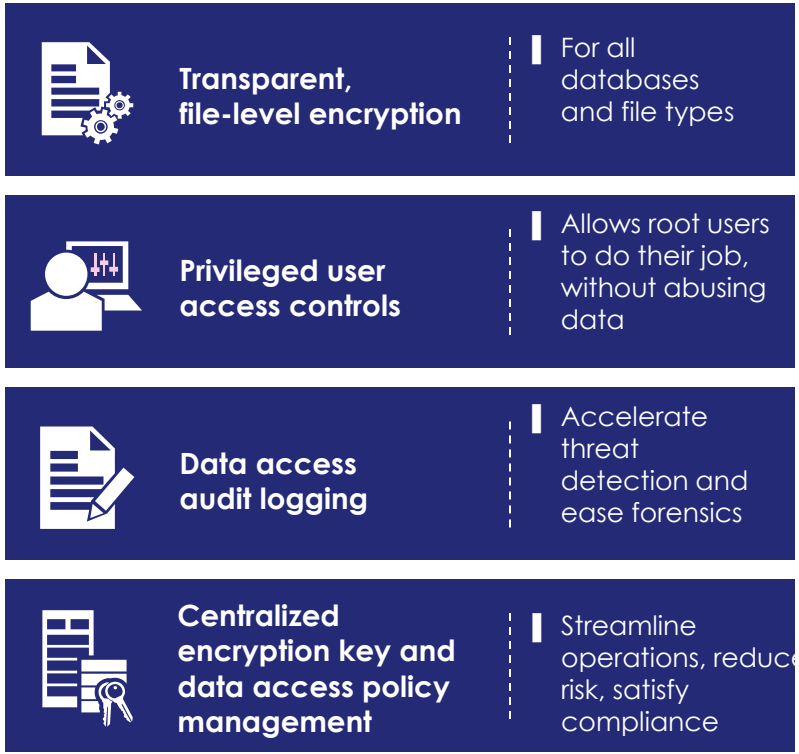
– Kevin Mitnick

Top storage security challenges



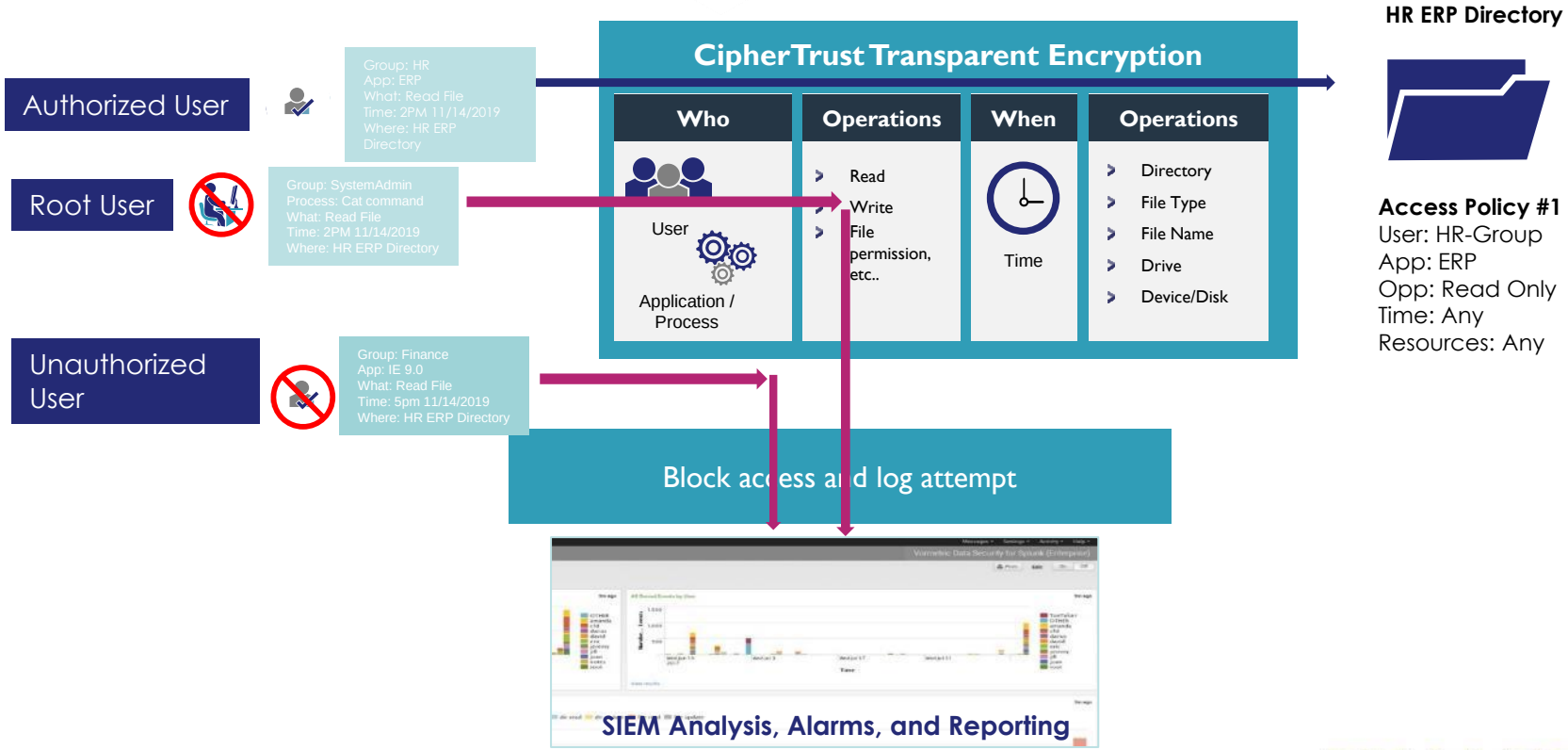
THALES

The layers of protection

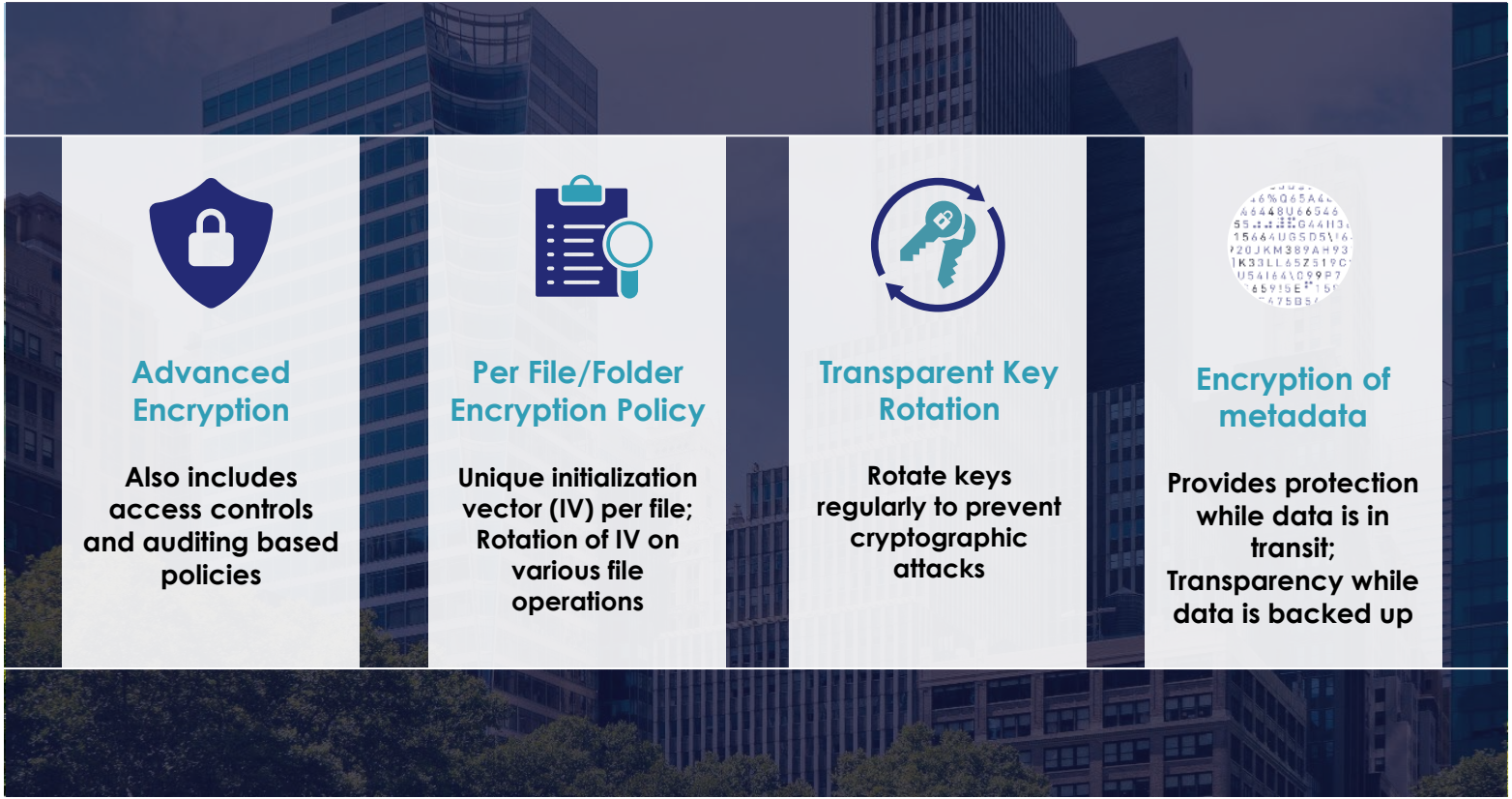


- File level enhanced encryption
- Fine-grain access control
- Device protection from unauthorized access
- Application whitelisting - identify "trusted applications"
- System level "audit logs"

Process and user aware file access policies



Enhanced encryption and protection



Advanced Encryption

Also includes access controls and auditing based policies

Per File/Folder Encryption Policy

Unique initialization vector (IV) per file; Rotation of IV on various file operations

Transparent Key Rotation

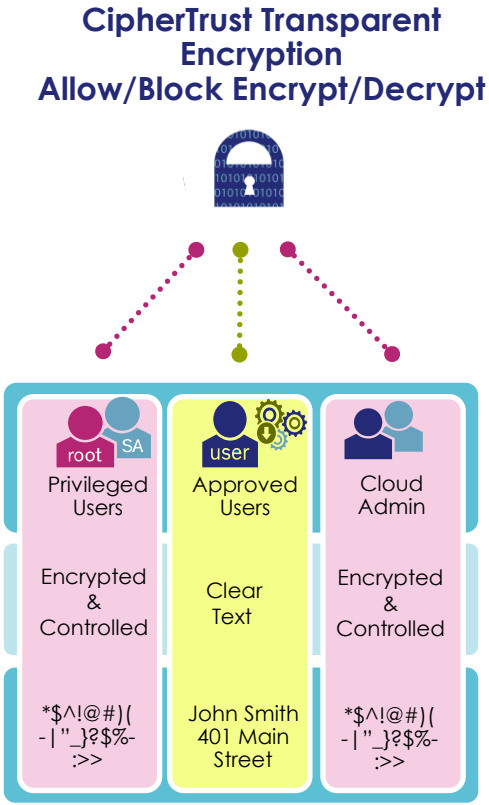
Rotate keys regularly to prevent cryptographic attacks

Encryption of metadata

Provides protection while data is in transit; Transparency while data is backed up

THALES

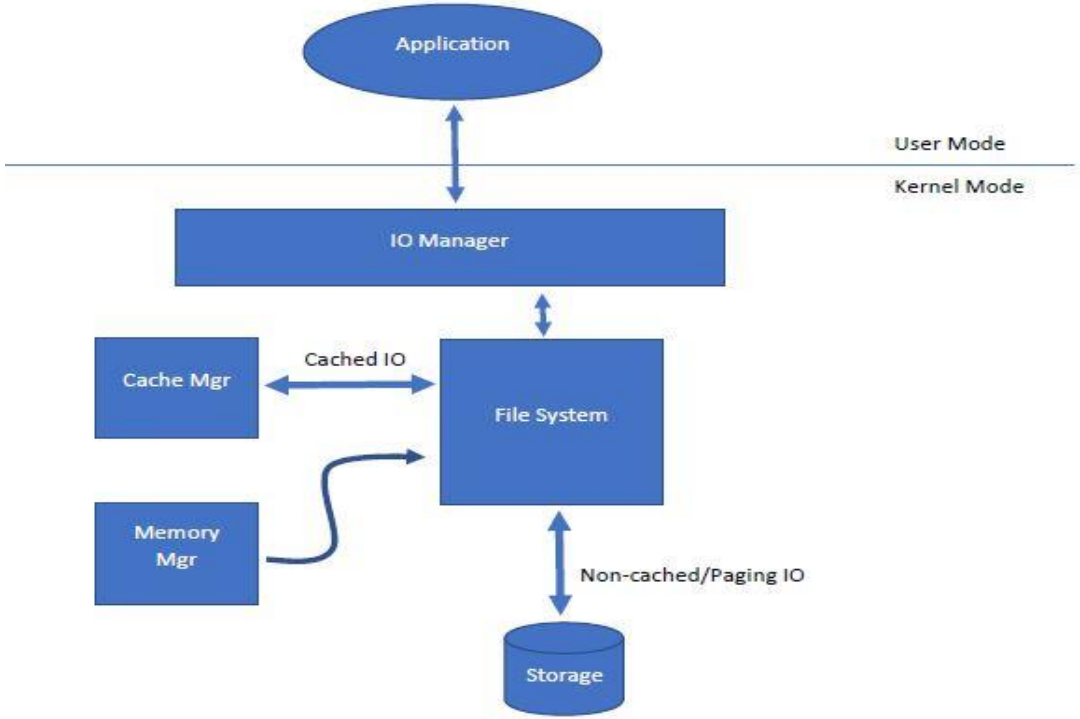
Device protection and app whitelisting



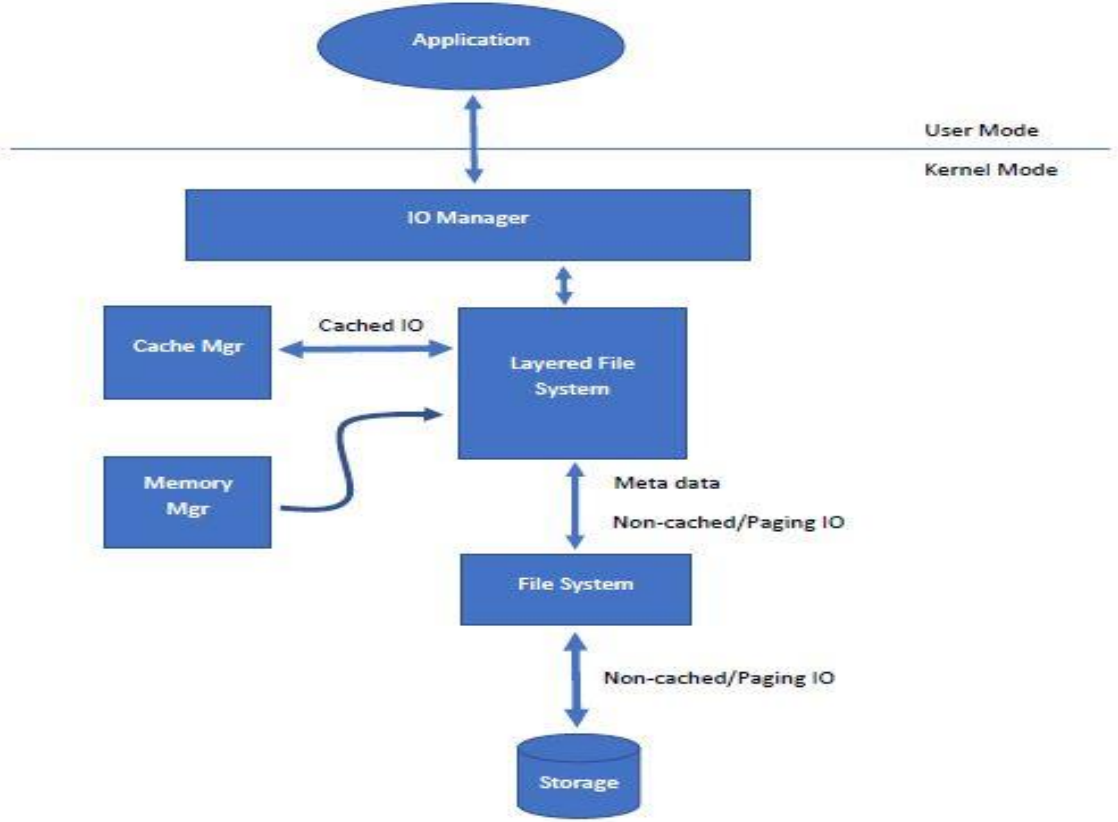
- **Identify "trusted applications"**
  - Allow only trusted applications to access the device
  - Allow only trusted applications to complete
- **Check the integrity of these trusted applications with signatures**
- **Associate the trusted applications with user**
  - Which application can be used to decrypt the data

Let’s look at design

The key to success



The key to success



The key to success

- Deployed a layered file system
  - More robust access controls on file objects
  - Allow more control over metadata management
- Implement the necessary per file integration with the OS
  - With cache and memory manager integration
  - Use underlying file system as a data store
  - Use cross-platform compatible format for metadata
- Leverage what we can
  - Use underlying file system namespace and attributes

Design and implementation

- Apply access controls in the LFS
  - Namespace access controls performed in a known context
  - Can extract caller information and security context
  - Understand and control how the caller wants to access the file
- White-list applications for fine granularity
  - Extract the process context
- Apply object manager integration
  - Prevent remote thread execution exploits
  - Memory address space scraping
- Understand data access for users based on content and tagging
  - Identify the sensitive files and prevent the unauthorized access

Design and implementation cont.

- Apply per file encryption through the LFS
  - Ability to add a *header* on the file to maintain per file state information
  - Easy to hide the header
  - Size adjustments for callers is integrated with the OS
- Expose different data sections to allow different *views* of the data
  - Standard section which offers view to caller of clear text
  - Read-only section which allows view of cipher text for applications such as backup

Design and implementation cont.

- Apply device level access controls
  - Only trusted applications can access device
  - Control and prevent direct device access
  - Access is managed through the LFS
- Device level encryption
  - Per file encryption has a noticeable performance impact. If majority of a device contains encrypted content, go with device level encryption
  - Key management is per device
  - Maintain data efficiency with storage vendor integration

A deep dive into design

A deep dive inside the LFS

- Implemented within the windows filter manager framework
  - Appears as a standard file system filter driver
- Takes ownership of *top-level* file objects and maintains private file object to underlying file
  - This entails control over file object pointers such as the section object pointers control structure and the fscontext and fscontext2 pointers.
  - System cache integration is handled within the LFS so only paging and direct, non-cached IO are passed to the underlying file system
  - Must ensure underlying file system obtains correct locks for handling paging IO

A deep dive inside the LFS

- Hiding the header from the upper layers as well as user mode applications
  - Requires size faking which is nearly impossible outside of a layered file system design
  - Adjusting IO requests accordingly requiring the header to be a sector aligned, fixed length region at the start of the file
  - Header vs footer? Prevents need to open files during directory enumerations to determine true size of file

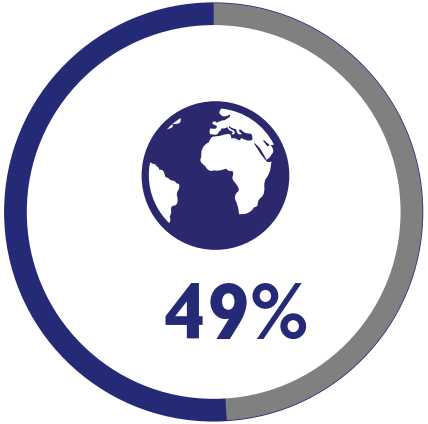
A deep dive inside the LFS

- Support for network shares like CIFS and NFS increases the complexity for everyone
  - A distributed environment must allow for remote server to maintain access rights and share access across multiple clients
  - Not all callers request read and write access; handle cases where header content changes, but current access cannot read or write header; Opportunistic Locks in an LFS
  - Performing key rotation, or initial encryption, on network files requires a central point of management
    - Distributed locking models are as complex as the OS itself
  - Must be able to maintain cache coherency across multiple clients
    - Windows feature to piggyback on opportunistic locks is key

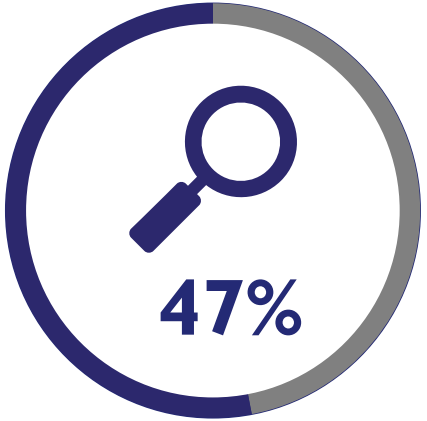
Final Thoughts

Data breaches continue, compliance is difficult and failure is time consuming and costly

of organizations admitted to having been breached in the past year.



of global respondents have experienced a breach at some point in their history.



of organizations report that they have been breached or failed a compliance audit in the past year.

2020 Thales Data Threat Report – Global Edition
Research and analysis by IDC

THALES

Future of security

- Can ransomware attacks and data scraping be stopped with this model?
  - Yes but it comes at a cost – tighter restrictions can cause application incompatibilities
- Disgruntled employees have full access, or do they?
  - Avoid data exfiltration
  - Access control can be quickly modified but there is still a window
- How do we move this to the cloud?
  - Access through client-side redirectors can be managed
    - On windows, azure access integrated within explorer or through web client
  - Raw blob storage is more complicated
    - How to efficiently handle blob updates