STORAGE DEVELOPER CONFERENCE
SDC 21
BY Developers FOR Developers

Virtual Conference
September 28-29, 2021

A SNIA Event

Hardening OpenZFS

To Further Mitigate Ransomware Attacks

Michael Dexter
CTO, Gainframe
Member, SNIA DPPC

Related SDC 2021 Talk

Ransomware!!! – an Analysis of Practical Steps for Mitigation and Recovery

Thomas Rivera – Mounir Elmously

Ransomware according to SNIA

*Software that uses **encryption** to disable a target's access to its data until a ransom is paid.*

*There is **no guarantee** that payment will result in the necessary decryption key or that the decryption key provided will function properly.*

The Open Zettabyte File System

- Vendor-neutral, Open Source, and widely adopted
- Combined File System and Volume Manager
- Cross-platform, supporting:
  - Tier 1: illumos, FreeBSD, GNU/Linux, and macOS
  - Tier 2: NetBSD and Windows
- Continuously validated on write and read with checksums
- Highly scalable, supporting multiple petabytes without penalties
- Highly flexible, supporting “hybrid” spinning and flash media
- Atomic Copy On Write (COW) for efficient snapshotting

ZFS Was Not Possible

- Space could not be wasted on metadata or snapshots
- Before the year 2000, CPUs could not be wasted calculating and validating checksums
  - CPUs now *increase* storage performance via compression
- File system portability was rarely a priority unless proprietary
  - 1:1 relationship of operating systems and file systems
- Flash storage did not adhere to standard interfaces
  - Specialized NAND storage rather than SAS, SATA, or NVMe
- Open Source was the exception rather than the norm

What is *now* possible with OpenZFS

- Affordable snapshotting without performance penalties
  - The perfect foundation for efficient replication
- Complete hardware agnosticity
  - Hardware RAID cards had a good run
- Nearly-complete operating system portability
  - Endian-agnostic, portability limited by enabled features
- Native encryption and enough features to fill several books
- **A robust administrative layer below the POSIX layer**

Operating *below* the POSIX Layer

- POSIX: Files, folders, permissions, ACLs, EAs…
- OpenZFS: Data Management Unit (DMU) object store
  - POSIX available as a *compatibility* layer
  - zvol “volumes” for block storage
  - Lustre supports direct DMU operations
- `zfs set readonly=on <pool>/<dataset>`
- `zfs unmount <pool>/<dataset>`
- *Dynamically* set exec, compression, recordsize, sync, atime…
  - No longer mount-time operations!

OpenZFS Ransomware Mitigation Today Using Snapshots and Replication

Real-world scenarios I see every day

OpenZFS Snapshots in Detail

- Efficient point-in-time deltas/states of a dataset or volume
  - Dataset: A directory with inherited or local properties
  - Volume: A block device with inherited or local properties
- Checksummed on write, validated on read
- Immutable unless “Cloned” for a writable dataset or volume
  - Zero additional space until written to
  - New writes are deltas of the immutable data
- “Sent” to another local or remote dataset for an *identical* copy
- Minimal performance overhead unlike most snapshot options

OpenZFS Snapshots in Practice

- Select a supported NAS or general purpose OS
- Select a quality hardware platform, HBA, and network adapters
- Select quality SAS, SATA, or NVMe storage devices
- Plan for RAID-Z2 distributed parity “virtual devices”
  - RAID 5 N+2 equivalence with N+1 and N+3 available
  - Group devices with less than 10 devices per “vdev” group
  - Stripe additional vdevs for capacity *and* performance growth
- Alternatively use mirrored vdevs but I strongly suggest N+2
  - Excellent IOPS and throughput doing so if Finance agrees

OpenZFS Snapshots in Practice

- Configure for hourly or daily snapshots, even minutes
- Replicate the snapshots to a second, and ideally a third system
- Age-out snapshots at a minimum of your longest holiday, plus.
- Use the storage locally or share with a supported protocol
- File: SMB, NFS, AFP, FTP, WebDAV, SSHFS...
  - Anything that can speak to the POSIX layer internally
- Object: S3-compatible, Lustre on DMU, Gluster on POSIX
- Block: iSCSI, Fibre Channel
- If you can share it from POSIX, you can back it with OpenZFS

Snapshot Retention Policies

- Some organizations comply with mandates for year of retention or longer
- When mitigating Ransomware, aim for a minimum of the longest holiday plus a week
- The perfect encryption-based ransomware storm comes from a bored user prior to a long holiday, watching cat videos or rummaging through their spam box
- Malware can initiate a diverse attack that goes unnoticed for the duration of the holiday, until everyone returns to work

This is a very good start...

- Variations on this strategy help users around the world recover quickly from ransomware attacks
- You can roll back to an uninfected snapshot and do your best to see what recent or in-flight data was lost
- Optionally “clone” the infected state for forensics/recovery
- Note, “hourly, weekly, weeks, months…”
  - Time is always a very important factor
  - Foundation of Restore Time/Point Objectives
  - How many minutes of data changes can you tolerate “losing”?

OpenZFS Ransomware Mitigation Tomorrow and Beyond

Mitigating the current threat is a good start but the threat will evolve

Who is the Threat?

- Have you ever accidentally deleted data?
  - No malicious attacker required
- Has you mail client ever auto-completed the wrong address?
  - Data exfiltration without malice
- Has a storage system failed without chance of recovery?
  - Also no attacker required
- Have you ever lost a data encryption key or credentials?
  - Ransomware with no one standing by to accept payment

The professionals are simply better at causing this same damage

Never Forget the Fundamentals

- Pre-Computer Security Fundamentals
  - Need-to-know basis/Principle of Least Privilege (PoLP)
  - Keep Response time below Penetration/Detection time
  - Pt > Dt + Rt = “Secure” – Safe cracker is arrested in time
- Computer Era Security Fundamentals
  - “Block All” Network and File access and *add* require access
  - Stay up to date (Assuming vendors are acting responsibly)
  - Train you team caution and skepticism. Recognize .pdf.exe
  - Know your data. Inventory it. Validate it. You know it best!

Updating Our Expectations

- We do have the OpenZFS hammer and will look for nails
  - OpenZFS: above-average job with ransomware recovery
  - “It’s only a matter of time” before OpenZFS is targeted
  - Let’s focus on that with beginning with one goal in mind

Let’s first reduce Ransomware attacks to the level of tolerable petty theft and vandalism, rather than the disastrous force that it is

1. Leverage the Management Layer

- Dynamic data immutability, visibility, and executability required expensive administrative operations!
  - Remount “read/only” to achieve immutability
  - Unmount to achieve invisibility
  - Remount to set “noexec” executability (noexec ~/Downloads!)
- OpenZFS management delegation allows for dedicated users
- Let’s embrace and leverage these...

1. Leverage the Management Layer

- Remember: Read/only data cannot be encrypted
- Remember: Invisible data cannot be exfiltrated
- `zfs set readonly=on <pool>/<dataset>`
- `zfs unmount <pool>/<dataset>`
- If your use case and application allows for it:

Automatically tier data immutability and visibility by age

1. Leverage the Management Layer

Proactively set immutability in response to anomalous activities

- The sooner and wider the encryption, the higher the ransom
- The inconvenience of a periodic false positives by users beats company-wide impacts
- Alert and log when alarms are tripped, time is of the essence
- Empower users to initiate lockdown in case of suspicious behavior

A.K.A. Volvation

Image © Melissa McMasters – Creative Commons 2.0 Generic

2. Trust Less

The Castle Wall and Moat are obsolete.
Control access internally.

- NIST SP 800-207 is vague about “ZeroTrust” and storage
- Simplified: Your teenager receives money as needed, not with the proverbial “blank check”
- Leverage Multi-Factor Authentication (MFA)
- Grant least-privileged storage access and escalate as needed
- Human error *alone* is reason to minimize trust and reach/access

2. Trust Less Continued

- I personally am *not* a fan of any biometrics
- Virtual desktops (VDI) may preclude *all* storage access
- Long-term: The POSIX 1003.1e ACL draft could consider Access-Based (File/Directory) Enumeration (ABE)

Either way, repeat after me...

Immutable data cannot be encrypted

Invisible data cannot be exfiltrated

3. Secure the Management Layer

- It's only a matter of time that OpenZFS itself is targeted
- Here and now:
  - Embrace OpenZFS delegation of administration abilities
  - Non-root/administrator, MFA-authenticated users can perform most administrative duties
  - Solves the age-old WAN server backup problem
  - Do the same for other storage administration tasks and roles
  - Administrative binaries can be *removed* from systems

3. Secure the Management Layer

- Attackers may bring their software own tools but…

Operating System diversity is a proven exposure reduction strategy

OpenZFS excels in this regard, beat only by FAT32

3. Secure the Management Layer

- “Open” ZFS is just now gaining momentum
  - Gradually unifying code base on all operating systems
  - First hand experience: OpenZFS on Windows is coming
- Future:
  - OpenZFS contains the counters needed for “volvation”
  - 2FA/MFA requirement for all administrative actions
  - “Confine the omnipotent root” user
    - Consider securelevel requirements for storage actions
    - Goal: Immutability hard-enforced on a running system

4. “Poisoned” Replication Stream

- OpenZFS operates in kernel space for performance
- Introduces an attack vector of a “poisoned” snapshot/replication send stream payload

- Solutions:
  - The pool-wide “checkpoint” can globally undo *any* new writes
  - A virtual machine with an independent kernel can isolate

OpenZFS and Virtual Machines were made for eachother

5. Mitigating “Sleeper” Ransomware

- Sounds exciting, “slow attack” clearly involves time
- Let’s assume significant, automated immutability
  - Assume immutable, “noexec” archival storage
  - Your aging installers, notably Java, are by definition a threat
  - Ransomware execution will always be in the present
  - Snapshot history can be rebuilt to exclude undesired items
  - “Redacted send” is designed to do exactly that
  - Air gap and checksum validate your archives regardless
    - `zfs diff, sha256, md5deep` – watch for anomalies

6. Air-Gapped, Online Archives

- OpenZFS is an “online” storage platform unlike tape
- This provides many opportunities for *validating* backups
  - Clone your database snapshots and verify that they import
  - Clone your virtual machine snapshots and boot them
  - Checksum with external tools to detect tampering
  - Immutable, online archives are conducive to multi-format archiving towards the “100-Year Archive” goal

Hopefully this has broadened your thinking about how OpenZFS can mitigate evolving ransomware threats

Thank You!

Michael Dexter

dexter@gainframe.com
@michaeldexter

Related, Previous SDC Talks

- SDC 2018: “Combating Evolving Ransomware at the Block Level”
- SDC 2017: “Mitigating Ransomware at the Block Level with OpenZFS”
- SDC 2017: “By the Book: Open Source Reference Implementations for Key SNIA Terminology”

Note: Incorporate These Ideas?

Why isn't OpenZFS more popular?
Legal interpretations to date
NIST ZeroTrust concepts (No storage section?)
Lengthy Appendix for the PDF!
"Malice and incompetence are often indistinguishable"
"Cockup before Conspiracy"
Encryption atop encryption
Micro-segmentation preventative (like network vlans) (software, not hardware) - fine-grained (though at the network level)
GDPR issues? KMIP? (SNIA rough time?)
Reporting obligations?

Additional points to integrate

- A “pool” of blocks that is flexibly allocated (pool terminology)
- The merkel tree is kinda-sorta a blockchain (take a drink)
- A paranoid security position before they access any storage resources
  - Do not forget the fundamentals! (An industry unto itself)
  - GeoIP block all WAN access
    - Ideally bock all and selectively allow
  - Minimize WAN exposure by port
  - Rate-limit connection attempts, block for a period, blacklist repeaters
  - Consider VDI over VPN access in place of storage access
  - Plus countless more, all prior to reaching storage resources
- Remember the Immutable data/Executable binary dichotomy
  - *Binaries* and their directories should be read-only and immutable to the greatest extent feasible
  - *Data* and its directories should be non-executable
    - Verify if any storage protocols (SMB? NFS? Do *not* respect this)
  - Tier out data to immutability based on age and access time
    - If it’s infrequently read it is often infrequently written
    - Could aggressively tier out files and directories to read-only, causing some user inconvenience (make a copy from a read-only version), but in exchange for a high-level of protection. While never truly “WORM” on a live computing system, it could greatly reduces exposure
  - Behavior Monitoring
    - Watch for unusual/inhuman file opens with Samba/audit_full/DTrace/etc.
    - Not to be confused with disk activity such as a scrub/maintenance operation
- (Elaborate on pragmatic snapshotting and replication)
- Air gap (but validate!) backups
- Validate backups with virtual machines - Do databases import? VMs boot?

Section Title

Section Subtitle

STORAGE DEVELOPER CONFERENCE

SDC 21

BY Developers FOR Developers

Virtual Conference
September 28-29, 2021

A SNIA Event

Presentation Title

Presentation Subtitle

Presented by

Section Title

Section Subtitle

Dark Slide Title

- Bullets 1
  - Bullets 2
    - Bullets 3
      - Bullets 4
        - Bullets 5

Light Slide Title

- Bullets 1
  - Bullets 2
    - Bullets 3
      - Bullets 4
        - Bullets 5

Important Notes

- In this new template, we have the following layouts
  - Cover slide
  - Light content slide
  - Dark content slide
  - Purple section divider slide
  - White section divider slide

- When copying over slides into this new template, please look out for the following:
  - In some decks, the page number was not part of the template, so an extra page number could get copied over. Make sure to delete the extra page number on the slide. It typically shows up in the middle-right of the slide.
  - The new template has a shallower header and a slightly larger footer. You may need to adjust the content of slides upwards a bit.
  - Any formatting changes you've made to titles, bullets, etc. will copy over. After copying the slide,

Section Title

Section Subtitle

Light Slide Title

- Bullets 1
  - Bullets 2
    - Bullets 3
      - Bullets 4
        - Bullets 5

Dark Slide Title

- Bullets 1
  - Bullets 2
    - Bullets 3
      - Bullets 4
        - Bullets 5

