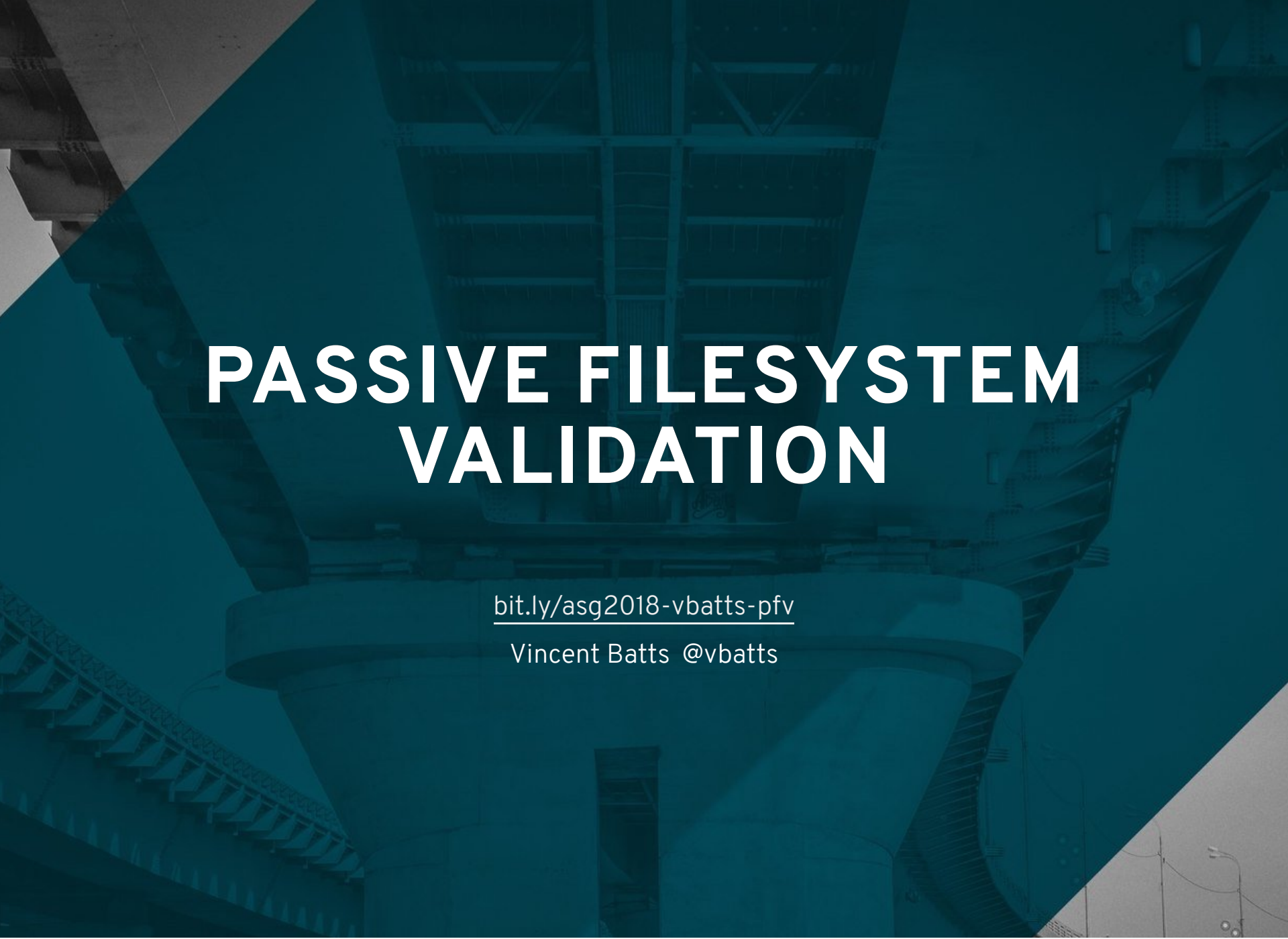


PASSIVE FILESYSTEM VALIDATION

bit.ly/asg2018-vbatts-pfv

Vincent Batts @vbatts

```
$> finger $(whoami)
```

Login: vbatts **Name:** Vincent Batts

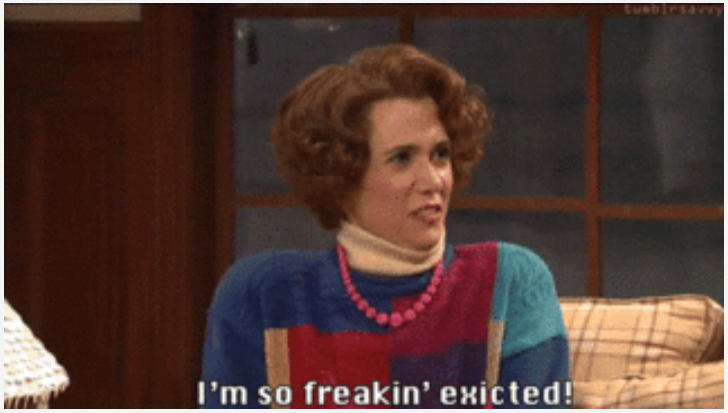
Directory: /home/vbatts **Shell:** /bin/bash

Such mail.

Plan:

OHMAN

```
$> id -Gn
devel opencontainers docker appc redhat golang slackware
```



AGENDA

- Packaging

AGENDA

- Packaging
- Content Addressibility

AGENDA

- Packaging
- Content Addressibility
- Compression!

AGENDA

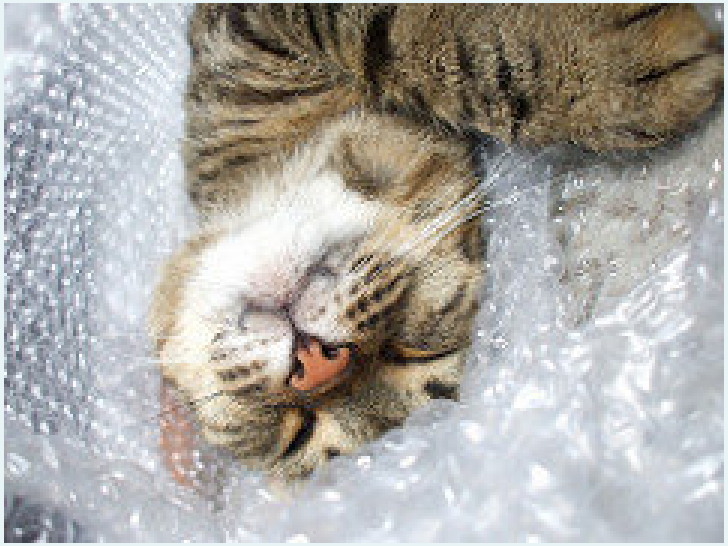
- Packaging
- Content Addressibility
- Compression!
- Reproducible Archives

AGENDA

- Packaging
- Content Addressibility
- Compression!
- Reproducible Archives
- Verify at rest filesystems

AGENDA

- Packaging
- Content Addressibility
- Compression!
- Reproducible Archives
- Verify at rest filesystems



PACKAGING

PACKAGING

PACKAGING

tar archives

PACKAGING

tar archives

Slackware packages (tar(1) archives)

PACKAGING

tar archives

- Slackware packages (tar(1) archives)
  - Debian *.deb (ar(1) archive of tar(1) archives)

PACKAGING

tar archives

- Slackware packages (tar(1) archives)
  - Debian *.deb (ar(1) archive of tar(1) archives)
    - Red Hat *.rpm (custom key/value binary and cpio(1))

PACKAGING

tar archives

- Slackware packages (tar(1) archives)
  - Debian *.deb (ar(1) archive of tar(1) archives)
    - Red Hat *.rpm (custom key/value binary and cpio(1))
      - Java *.jar and *.war (zip(1) archive)

PACKAGING

tar archives

- Slackware packages (tar(1) archives)
  - Debian *.deb (ar(1) archive of tar(1) archives)
    - Red Hat *.rpm (custom key/value binary and cpio(1))
      - Java *.jar and *.war (zip(1) archive)
        - Ruby *.gem (tar(1) archive of tar(1) archives)

PACKAGING

tar archives

Slackware packages (tar(1) archives)

Debian *.deb (ar(1) archive of tar(1) archives)

Red Hat *.rpm (custom key/value binary and cpio(1))

Java *.jar and *.war (zip(1) archive)

Ruby *.gem (tar(1) archive of tar(1) archives)

Container Images (tar(1) archives)

CONTENT ADDRESSIBILITY

CONTENT ADDRESSIBILITY

Opaque Object storage

CONTENT ADDRESSIBILITY

Opaque Object storage
changed object = new object

CONTENT ADDRESSIBILITY

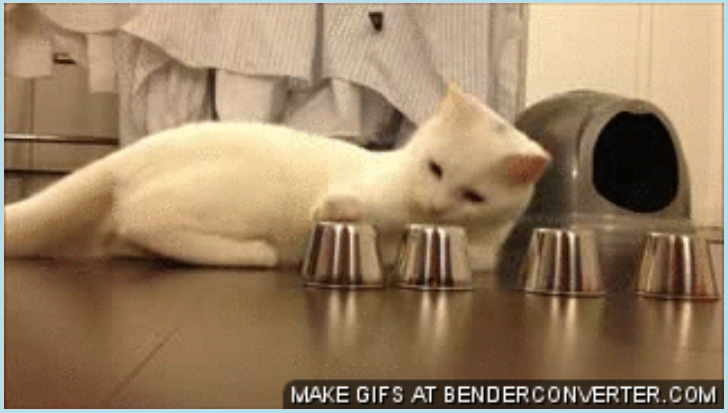
Opaque Object storage

changed object = new object

cryptographic assurance

CONTENT ADDRESSIBILITY

Opaque Object storage
changed object = new object
cryptographic assurance



COMPRESSION!

COMPRESSION!

same objects, but variation in compression

COMPRESSION!

same objects, but variation in compression

inflate/deflate (RFC1951)

COMPRESSION!

same objects, but variation in compression

inflate/deflate (RFC1951)

Gzip (RFC1952)

COMPRESSION!

same objects, but variation in compression

inflate/deflate (RFC1951)

Gzip (RFC1952)

`gzip` vs Golang `compress/gzip` vs Zlib

COMPRESSION!

same objects, but variation in compression

inflate/deflate (RFC1951)
Gzip (RFC1952)
`gzip` vs Golang `compress/gzip` vs Zlib

ideally compress for transfer and storage, but not for identity

COMPRESSION!

```
#!/bin/sh
dd if=/dev/urandom of=rando.img bs=1M count=2
cat rando.img | gzip -n > rando.img.gz
cat rando.img | gzip -n -9 > rando.img.9.gz
cat rando.img | xz > rando.img.xz
cat rando.img | xz -9 > rando.img.9.xz
sha1sum rando.img* > SHA1

cat rando.img | gzip -n > rando.img.gz
cat rando.img | gzip -n -9 > rando.img.9.gz
cat rando.img | xz > rando.img.xz
cat rando.img | xz -9 > rando.img.9.xz
sha1sum -c ./SHA1
```

COMPRESSION!

```
#!/usr/bin/env ruby

require 'zlib'
include Zlib

input = File.open(ARGV.first)
GzipWriter.open(ARGV.first + '.gz', DEFAULT_COMPRESSION, HUFFMAN_ONLY) do |gz|
  gz.write(IO.binread(input))
end
input.flush()
input.close()
```

COMPRESSION!

```
package main

import (
        "compress/gzip"
        "io"
        "os"
)

func main() {
        input, err := os.Open(os.Args[1])
        if err != nil {
                println(err.Error())
                os.Exit(1)
        }
        output, err := os.Create(os.Args[1] + ".gz")
        if err != nil {
                println(err.Error())
                os.Exit(1)
        }
        gz := gzip.NewWriter(output)
        if _, err := io.Copy(gz, input); err != nil {
                println(err.Error())
                os.Exit(1)
        }
```

VERIFY AT REST FILESYSTEMS

dm-verity in select use-cases

VERIFY AT REST FILESYSTEMS

Regardless of transport, ensure resulting filesystem

dm-verity in select use-cases

VERIFY AT REST FILESYSTEMS

Regardless of transport, ensure resulting filesystem
(*.tar archive, rsync, bittorrent, IPFS, etc)

dm-verity in select use-cases

VERIFY AT REST FILESYSTEMS

Regardless of transport, ensure resulting filesystem
(*.tar archive, rsync, bittorrent, IPFS, etc)

`rpm -qV <package>` functionality

dm-verity in select use-cases

VERIFY AT REST FILESYSTEMS

Regardless of transport, ensure resulting filesystem
(*.tar archive, rsync, bittorrent, IPFS, etc)

`rpm -qV <package>` functionality

dm-verity in select use-cases

Future hopes could be IMA/EVM

VERIFY AT REST FILESYSTEMS

Regardless of transport, ensure resulting filesystem
(*.tar archive, rsync, bittorrent, IPFS, etc)

`rpm -qV <package>` functionality

dm-verity in select use-cases

Future hopes could be IMA/EVM

Passive validation of directory hierarchies

VERIFY AT REST FILESYSTEMS

Regardless of transport, ensure resulting filesystem
(*.tar archive, rsync, bittorrent, IPFS, etc)

`rpm -qV <package>` functionality

dm-verity in select use-cases

Future hopes could be IMA/EVM

Passive validation of directory hierarchies
BSD mtree(8)

VERIFY AT REST FILESYSTEMS

FreeBSD mtree(8)

mtree-port (for linux)

go-mtree (golang cli and library)

libarchive-formats(5) (so `bsdtar`)

casync mtree

umoci unpack

VERIFY AT REST FILESYSTEMS

gomtree

```
[root@host /]# gomtree -c -K sha256 -p /usr/ | head -30
#           user: root
#        machine: host
#           tree: /usr
#           date: Wed Sep 26 16:07:53 2018
#       keywords: size,type,uid,gid,mode,link,nlink,time,sha256digest

# .
/set type=file nlink=1 mode=0664 uid=0 gid=0
. size=100 type=dir mode=0755 time=1524747817.000000000
    tmp size=10 type=link mode=0777 link=../var/tmp time=1517996467.000000000

# bin
bin size=5268 type=dir mode=0555 time=1537977678.074646319
    \133 size=48 mode=0555 time=1521637266.000000000 sha256digest=afd97bbd643bfe1473794af167cd
    alias size=29 mode=0755 time=1521122304.000000000 sha256digest=c9e358c5012c2cf9171ec4f7692
    applydeltarpm size=72752 mode=0755 time=1517985721.000000000 sha256digest=359f076a0a259bda
    arch size=51 mode=0555 time=1521637267.000000000 sha256digest=209bae4071910ef54b4a3bd30205
```

VERIFY AT REST FILESYSTEMS

casync mtree

```
[root@host /]# casync mtree /usr/
. type=dir mode=0755 uid=0 gid=0 time=1524747817.000000000
bin type=dir mode=0555 uid=0 gid=0 time=1537977212.955190222
bin/[ type=file mode=0555 size=48 uid=0 gid=0 time=1521637266.000000000 sha512256digest=f02fe2
bin/alias type=file mode=0755 size=29 uid=0 gid=0 time=1521122304.000000000 sha512256digest=c5
bin/applydeltarpm type=file mode=0755 size=72752 uid=0 gid=0 time=1517985721.000000000 sha5122
bin/arch type=file mode=0555 size=51 uid=0 gid=0 time=1521637267.000000000 sha512256digest=551
bin/awk type=link mode=0777 link=gawk uid=0 gid=0 time=1519649348.000000000
bin/b2sum type=file mode=0555 size=52 uid=0 gid=0 time=1521637266.000000000 sha512256digest=42
bin/base32 type=file mode=0555 size=53 uid=0 gid=0 time=1521637266.000000000 sha512256digest=4
bin/base64 type=file mode=0555 size=53 uid=0 gid=0 time=1521637266.000000000 sha512256digest=d
[...]
```

VERIFY AT REST FILESYSTEMS

umoci unpack

```
> skopeo copy docker://docker.io/busybox:latest oci:busybox:latest
> umoci unpack --image ./busybox:latest busybox-bundle
> cat busybox-bundle/sha256_9b9b48e2d92691f344ad1701e7df04f89dd2041f2f05cccfec39af8ac3a62d25.m
#          user: root
#       machine: host
#          tree: busybox-bundle/rootfs
#          date: Wed Sep 26 16:21:46 2018
#      keywords: size,type,uid,gid,mode,link,nlink,tar_time,sha256digest,xattr

# .
/set type=file nlink=1 mode=0664 uid=0 gid=0
. size=52 type=dir mode=0755 tar_time=0.000000000

# bin
bin size=4638 type=dir mode=0755 tar_time=1533068407.000000000
    \133 size=1083720 mode=0755 nlink=393 tar_time=1533068407.000000000 sha256digest=ee8001fee
    \133\133 size=1083720 mode=0755 nlink=393 tar_time=1533068407.000000000 sha256digest=ee800
    acpid size=1083720 mode=0755 nlink=393 tar_time=1533068407.000000000 sha256digest=ee8001fe
    add-shell size=1083720 mode=0755 nlink=393 tar_time=1533068407.000000000 sha256digest=ee80
    addgroup size=1083720 mode=0755 nlink=393 tar_time=1533068407.000000000 sha256digest=ee800
    adduser size=1083720 mode=0755 nlink=393 tar_time=1533068407.000000000 sha256digest=ee8001
```

VERIFY AT REST FILESYSTEMS

Tar Archive Support

```
tar cf /tmp/demo.tar .
gomtree -c -T /tmp/demo.tar -K sha256digest | tee /tmp/demo.mtree

gomtree -f /tmp/demo.mtree -T /tmp/demo.tar
echo $?

read

gomtree -f /tmp/demo.mtree -p ./
echo $?
```

VERIFY AT REST FILESYSTEMS

Directory Path

```
go get -u github.com/vbatts/go-mtree/cmd/gomtree
gomtree -c -p ./ -K sha256digest | tee /tmp/demo.mtree

gomtree -f /tmp/demo.mtree -p ./
echo $?

read

touch $0 # SCANDALOUS
gomtree -f /tmp/demo.mtree -p ./
```

VERIFY AT REST FILESYSTEMS

BSD mtree || mtree-port

```
mtree -c -p ./ -K sha256digest | tee /tmp/demo.mtree

mtree -f /tmp/demo.mtree -p ./
echo $?

read

touch $0 # SCANDALOUS
mtree -f /tmp/demo.mtree -p ./
```

VERIFY AT REST FILESYSTEMS

with packages: libarchive and python-libarchive-c

```
#!/usr/bin/env python

import libarchive

with libarchive.file_writer('../demo.mtree', 'mtree') as a:
    a.add_files('./')
```

NOTICE: libarchive uses older mtree format

VERIFY AT REST FILESYSTEMS

bsdtar (libarchive)

```
[root@host /]# bsdtar --format mtree -cf foo.mtree /usr
bsdtar: Removing leading '/' from member names
[root@fa97e1919c44 /]# more foo.mtree
#mtree
./usr gname=root uname=root time=1524747817.0 mode=755 gid=0 uid=0 type=dir
./usr/tmp gname=root uname=root time=1517996467.0 mode=777 gid=0 uid=0 type=lir
./usr/bin gname=root uname=root time=1537976676.897120032 mode=555 gid=0 uid=0
./usr/bin/[ gname=root uname=root time=1521637266.0 mode=555 gid=0 uid=0 type=f
./usr/bin/alias gname=root uname=root time=1521122304.0 mode=755 gid=0 uid=0 ty
[...]
```

NOTICE: libarchive uses older mtree format

CALL TO ACTION

get familiarized with mtree format

make your implementation compatible

consider your provenance and sharing fs metadata use-cases

github.com/vbatts/go-mtree

VINCENT BATTS

@VBATTS| VBATTS@REDHAT.COM

GITHUB.COM/VBATTS/TALKS

THANK YOU!