Least Privilege:

Security gain without developer pain

Travis McPeak
Sr. Security Engineer - Netflix
@travismcpeak

Least privilege has been around since the 1975.



“The Protection of Information in Computer Systems”

- ***Saltzer and Schroeder***

The idea is simple:

give only permissions that are required to get the job done.

So,
everybody has least privilege systems,

right?



$250+ million impact
40 million Credit Cards

18.4% of incidents involved privilege misuse*.



*2017 Verizon DBIR



143 million SSNs


29.5% of breaches involved web application attacks*.
*2017 Verizon DBIR

200+ thousand classified documents




25% of breaches involved internal actors*
(intentional or not)
*2017 Verizon DBIR

Not knocking these organizations - least privilege is hard!

Why?

We don’t want…
permissions to be frozen in time



We don’t want…
overprovisioned permissions

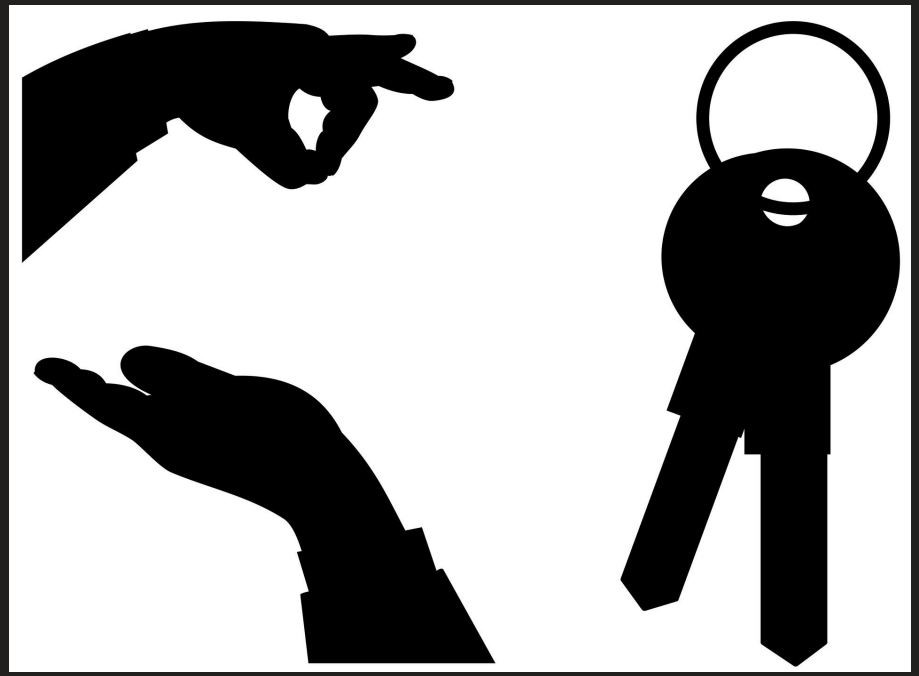
We don’t want…
permission granting to require this




“Gravity” pulls policies
larger over time.

In the physical world, permissions are taken away when no longer required.

My landlord takes my keys when I move out of an apartment.



If I stop working at Netflix, they'll revoke my badge.



My passport stops working if I don’t renew it.



In software, these status changes are typically silent and not actioned.



Abandoned projects, anybody?

If your environment is small, least privilege may be possible.



The rest of us need a different approach.



At Netflix we automatically remove unused permissions from our AWS policies.

This talk is about how.

AWS implements RBAC via the IAM service.

Applications assume roles.

Roles have attached policies.

Ex: Cloud Based Word Processor



+



Ex: Cloud Based Word Processor

```
{
  "Effect": "Allow",
  "Action": ["*:*"]
  "Resource": "*"
}
```





+



Ex: Cloud Based Word Processor

```
{
  "Effect": "Allow",
  "Action": ["s3:*"]
  "Resource": "*"
}
```





+



Ex: Cloud Based Word Processor

```
{
  "Effect": "Allow",
  "Action": ["s3:GetObject", "s3:PutObject"]
  "Resource": "*"
}
```



+



Ex: Cloud Based Word Processor

```
{
  "Effect": "Allow",
  "Action": ["s3:GetObject", "s3:PutObject"]
  "Resource": "arn:aws:s3:::wp_bucket"
}
```





+



AWS provides data about service, action, and resource usage.

Example:
My (super) cool new application



We give a base set of ***benign*** permissions to start with.

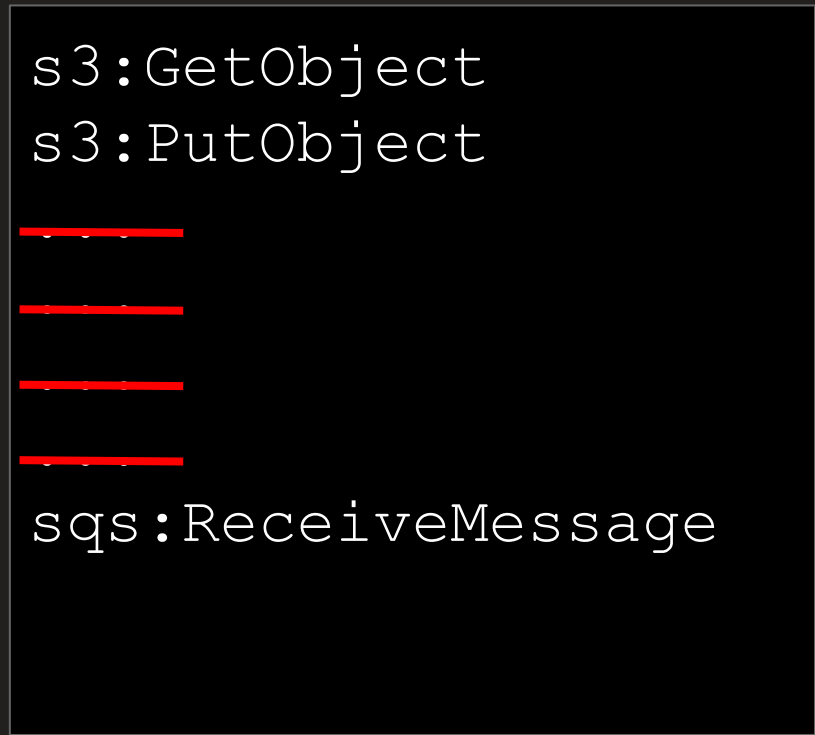
```
s3:GetObject
s3:PutObject
...
...
...
...
sqs:ReceiveMessage
```


We passively watch the application for 90 days to see which permissions are used.

After 90 days we start taking away unused permissions.

```
s3:GetObject
s3:PutObject
...
...
...
...
sqs:ReceiveMessage
```

After 90 days we start taking away unused permissions.



Developers make a ***simple*** request when their application needs more permissions.



Developers make a ***simple*** request when their application needs more permissions.



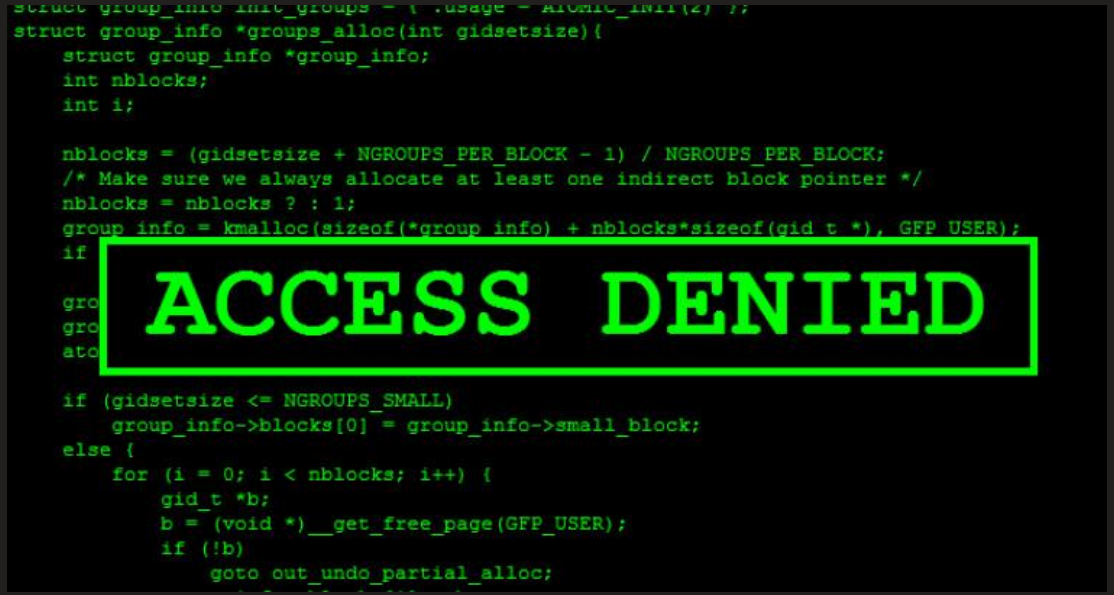

```
s3:GetObject
s3:PutObject
sqs:ReceiveMessage
```

Developers make a ***simple*** request when their application needs more permissions.




```
s3:GetObject
s3:PutObject
...
...
...
sqs:ReceiveMessage
```

When the application isn't using those permissions anymore they're removed.



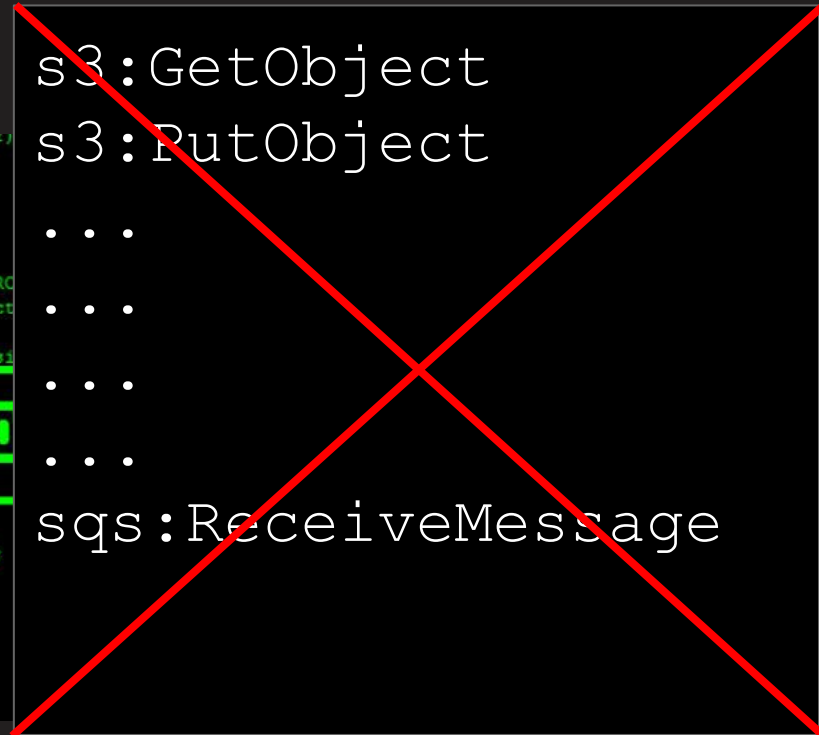
When the application isn't using those permissions anymore they're removed.




```
s3:GetObject
s3:PutObject
...
...
sqs:ReceiveMessage
```

When the application isn't using those permissions anymore they're removed.






Unused applications are
totally powerless
after 90 days!

At Netflix we removed 80% of permissions the first time we did this.

At Netflix we removed 80% of permissions the first time we did this.

Since then we've removed far fewer.

But what about...



… infrequently used permissions?



… infrequently used permissions?



Let’s leave those alone

… if we break applications?

… if we break applications?

We’ll detect and put them back - easily

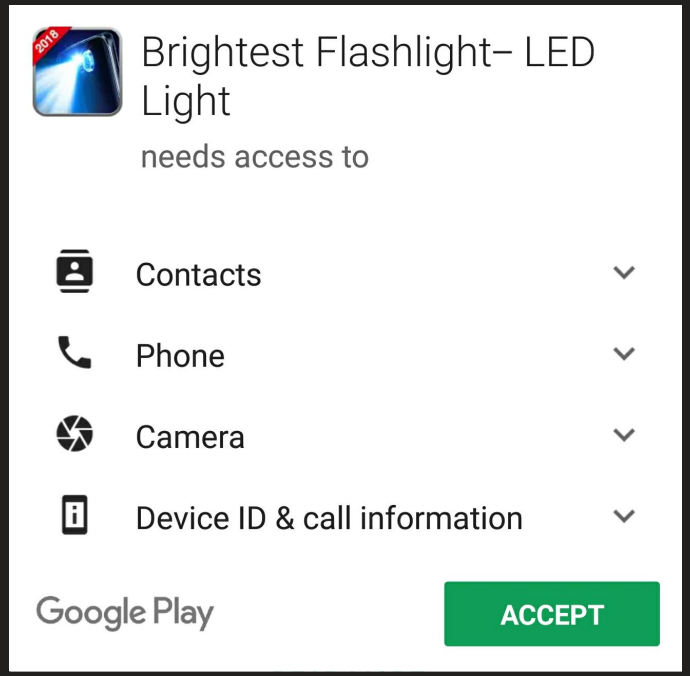
… if an attacker compromises your system and uses permissions the application needs?




... if an attacker
compromises your system and
uses permissions the application needs?
This is not a “silver bullet”!


What else!?

Can we extend this to mobile application privileges?





Can we extend this to container capabilities?

Closing thoughts

Closing thoughts



Closing thoughts





Closing thoughts







Closing thoughts









Repokid



https://github.com/Netflix/repokid

Thank you. Questions?

NETFLIX

@travismcpeak