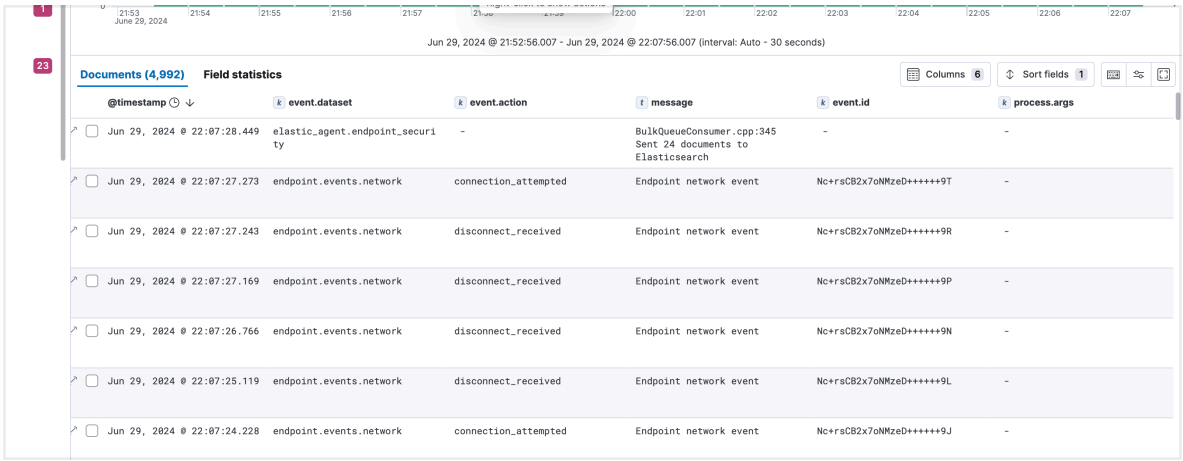
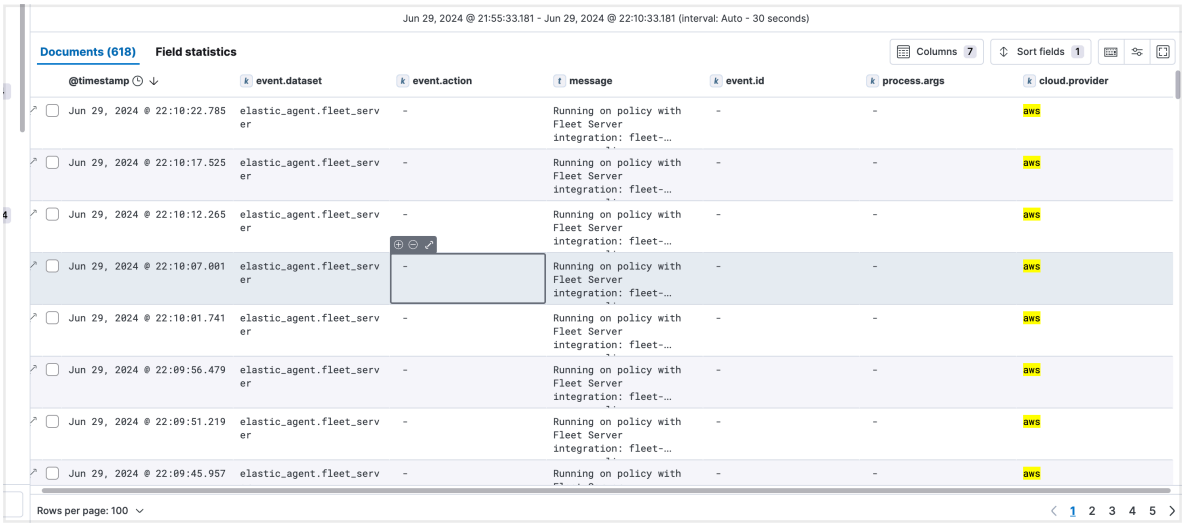
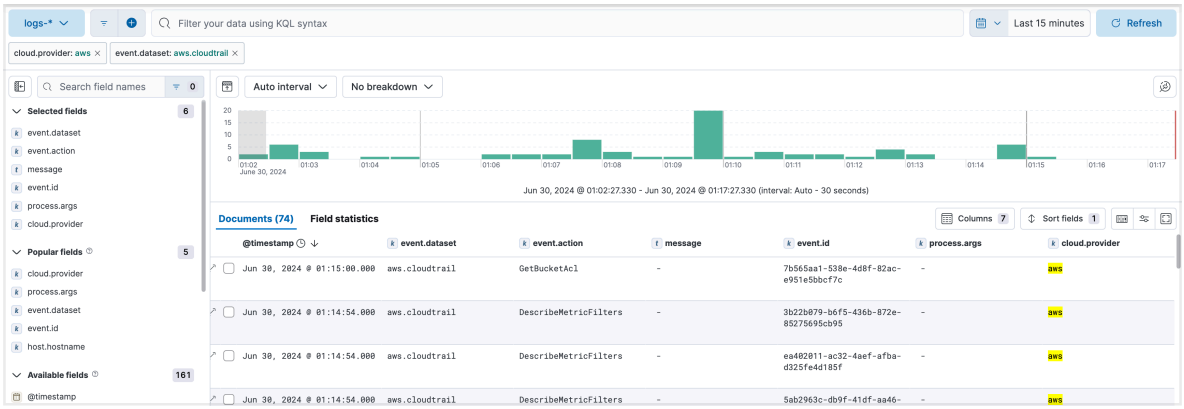
```
{
  "Version": "2012-10-17",
  "Id": "example-ID",
  "Statement": [
    {
      "Sid": "example-statement-ID",
      "Effect": "Allow",
      "Principal": {
        "Service": "s3.amazonaws.com"
      },
      "Action": "SQS:SendMessage",
      "Resource": "arn:aws:sqs:us-east-1:590184048130:myqueue",
      "Condition": {
        "StringEquals": {
          "aws:SourceAccount": "590184048130"
        },
        "ArnLike": {
          "aws:SourceArn": "arn:aws:s3:::aws-cloudtrail-
logs-590184048130-245a3a85"
        }
      }
    }
  ]
}
```





https://github.com/pro-prodipto/Netflix-Website-Project
https://github.com/mattweidner/bucket_finder



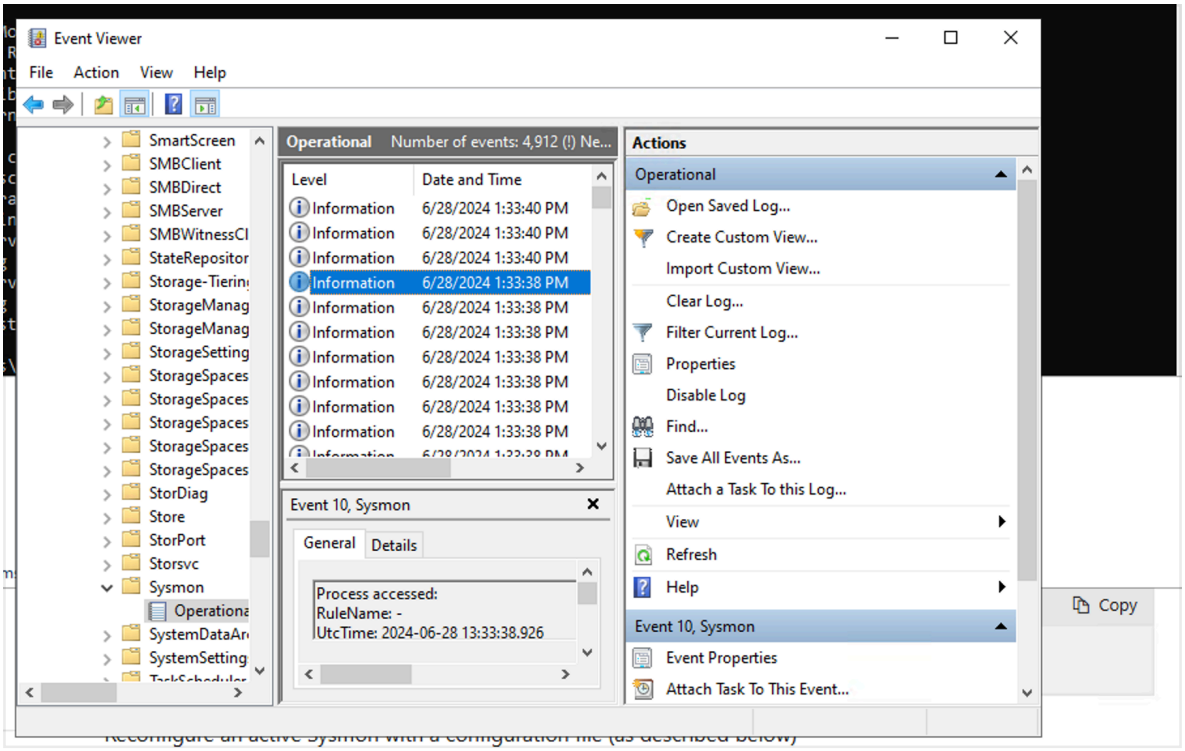
I don't think any of the current responses explain the concrete problem very well, so I'll give it a shot.

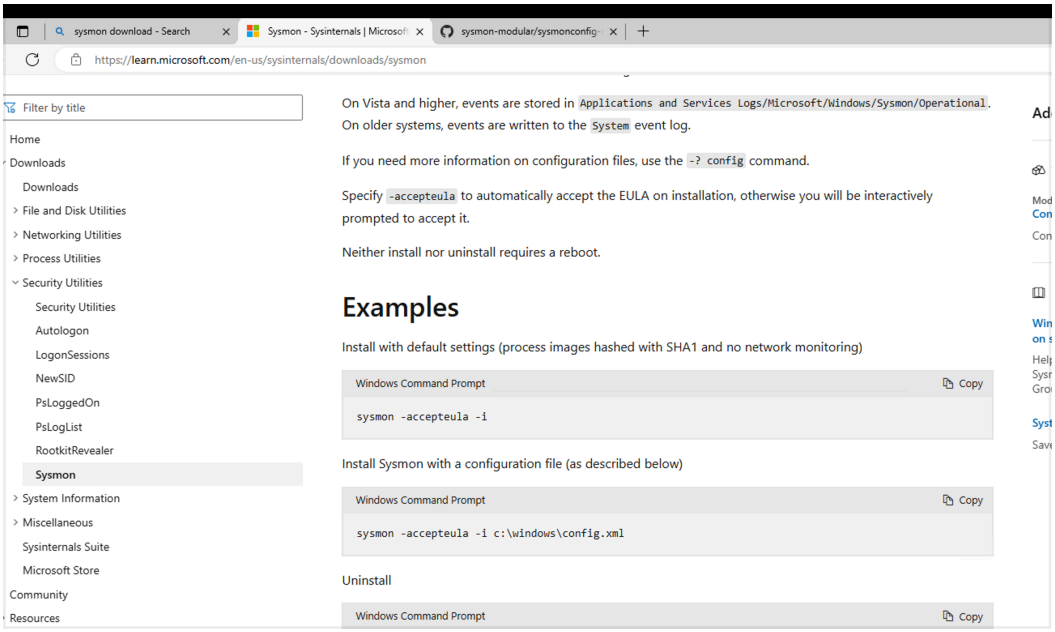
Let's imagine a 3rd party service that fetches data from your AWS account and then exposes it via a reporting interface. You sign up and create a cross-account role that trusts this 3p account. So now the 3P can assume your role, pull back data, and then show it to you through its own web interface.

Now imagine an attacker, Eve, signs up for the same service. She registers a totally legitimate account and has a valid username/password to the 3P system. But when she links her AWS account, instead of providing a role for an AWS account she owns, she actually manages to guess the ARN of *your* role and uses that. Because your role already trusts the 3P, it works totally fine. Now when Eve authenticates as herself to the 3P, the 3P fetches data from your account and then returns it to Eve. The 3P is now a confused deputy.

The mitigation here is to ensure Eve controls the role she provides and if she doesn't, prevent the 3P from assuming it on her behalf (but still allow it when it's operating on your behalf). So the 3P generates a unique External ID as part of the registration process and stores that with each account/role. So now, when the 3P assumes a role they pass in the External ID they generated for the

specific account in question. This means when you're logged in it's passing a different External ID than when Eve is logged in even though the role ARN is the same. The role assumption will only work if the external ID that the 3p passes in matches what's in the trust policy which means the person controlling the role/ trust policy is the same one making the request to the 3P.
The external ID doesn't need to be a secret because what matters is that your role trust policy only accepts the external ID associated with your 3P account. Even if Eve knows what that value is, it doesn't matter because the external ID for her 3P account is different. She could update every other role in the world to trust that external ID and it still wouldn't matter. The only thing that would help her is if she could convince you to update your trust policy to include her external ID, but that doesn't rely on the external ID being a secret.





Windows_Server-2022-English-Full-Base-2024.06.13
ami-04df9ee4d3dfde202