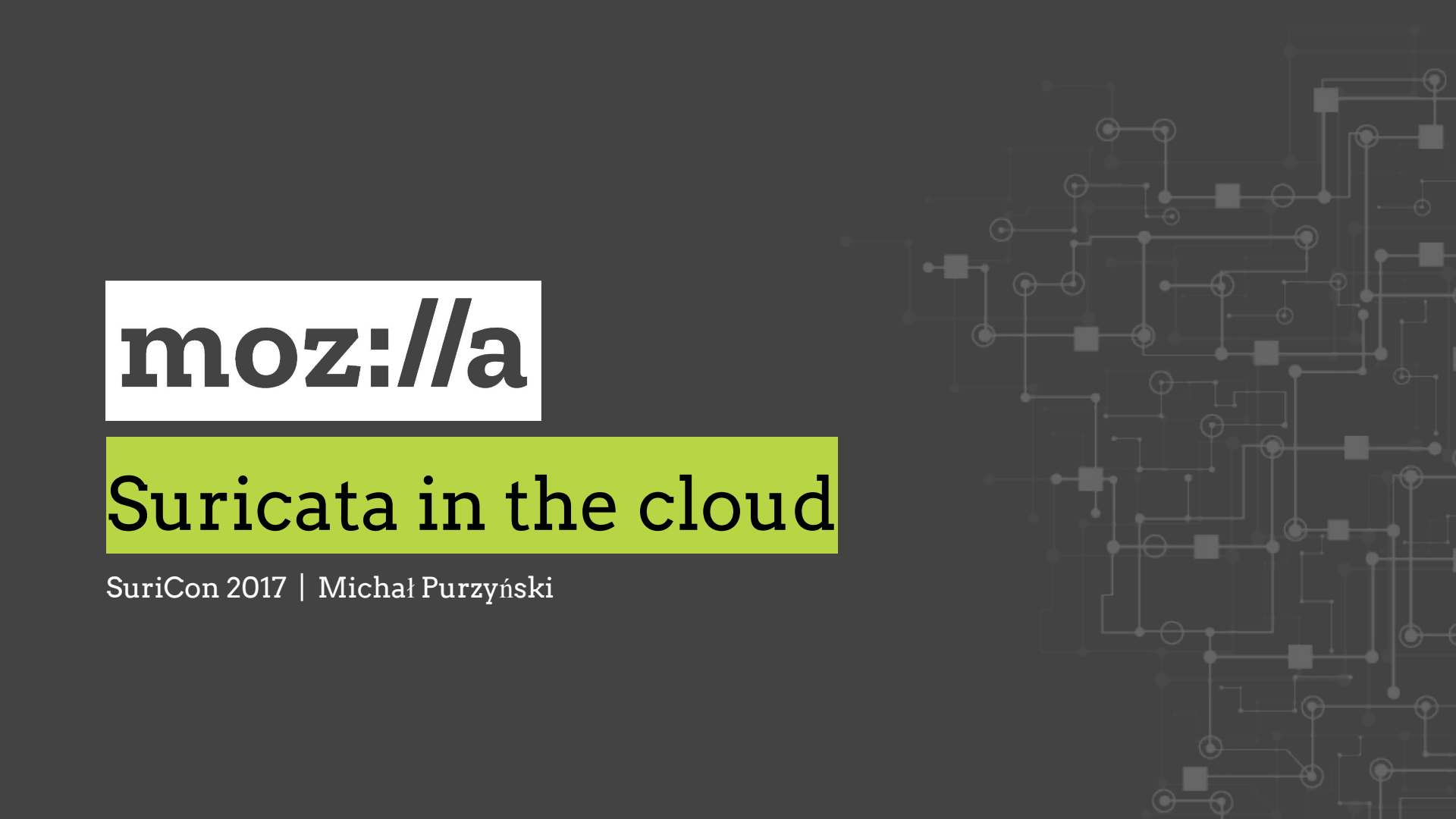


Suricata in the cloud

SuriCon 2017 | Michał Purzyński

You and your meerkats

You had your mob

You and your meerkats

Tuned

You and your meerkats

Monitored

You and your meerkats

Well taken care of

When it rains, it pours

C-something/VP-of-something

“Our 2012 strategy is to migrate to the cloud”

Lesson number one

Don’t even ask why

This happened to us

11.5 Suricata sensors

9.5 offices

1.75 datacenter

This happened to us

Bad news

We are migrating that datacenter to the cloud :(

This happened to us

Good news

It’s almost 2018 and we are still migrating ;)

A typical reaction

What does not help

Calling your “friendly” recruiter

Guess what

Everyone’s migrating to the cloud

What helps

Vacation :-)

Take your camera with you, leave Meerkats home

A good night sleep

Or 7

Drink water

Stop thinking about the cloud

And then

This happens

And then

What does your Suricata do, after all?

What does your Suricata do?

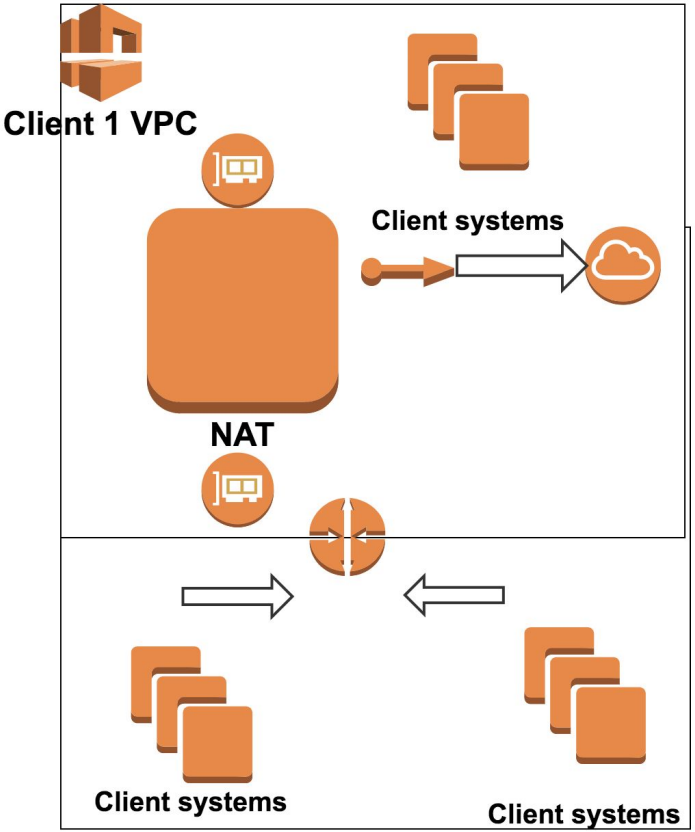
Performs pattern based detection

Produces high-quality connection logs

Saves the past DNS queries and responses

Extracts files

Your network in AWS



Pattern based detection

Must stay in Suricata

NAT instance?

High quality connection logs

Bad connection logs - AWS netflow parody

Run collector on clients? (In a docker? With latte?)

NAT instance can see it all so….

DNS queries and responses

Capture on clients?

Deploy your own DNS server?

NAT instance...

File extraction

See - pattern based detection

Before we go this path

Let's have some crazy ideas, shall we?

Grow The Mob

Run Suricata on each endpoint

Shrink the mob

Do not run Suricata

Intercept sys_connect(), listen(), socket()

Takes _easily_ more than 30% CPU

Feed your Meerkat through a pipe

Run a dedicated sensor away from NAT-i

Capture traffic on clients (netsniff-ng)

Send to Suricata over GRE

(this is less crazy than it sounds, vendors do it)

Feed your Meerkat through a pipe

(a puny photo would be SO cool here)

Suri riding the NAT

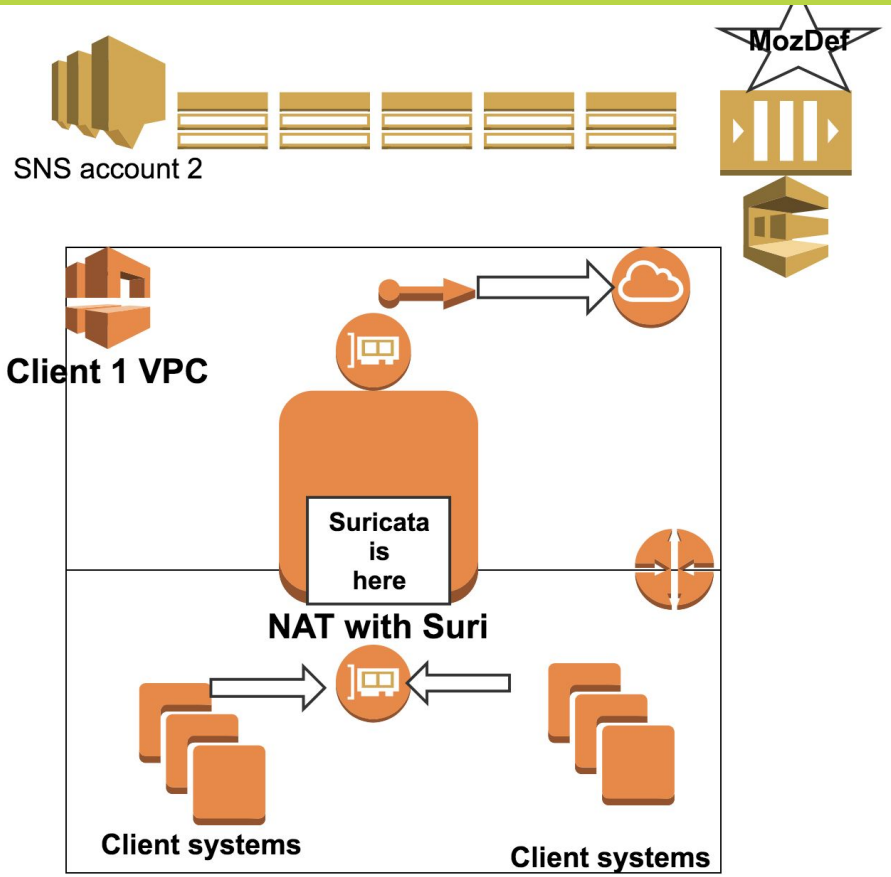
| public subnet + IGW | --- | int0 - NAT - int1 | --- | private subnet |

Run Suri on int1

Keep clients in private

Forget about ELB traffic

Suri riding the NAT



There is one more thing...

If you, like me, inspect your LB traffic with NSM

You have a problem

Where can we move that functionality?

To application logs.

Suri riding the NAT

Suricata 4.x

Ubuntu 16.04, Amazon Linux

Forget about CentOS, RHEL (or anything with broken kernel)

AF_Packet strikes back :-)

Suri riding the NAT

Wait, there's more!!

Suri riding the NAT

Packets -> Suri is like 15%

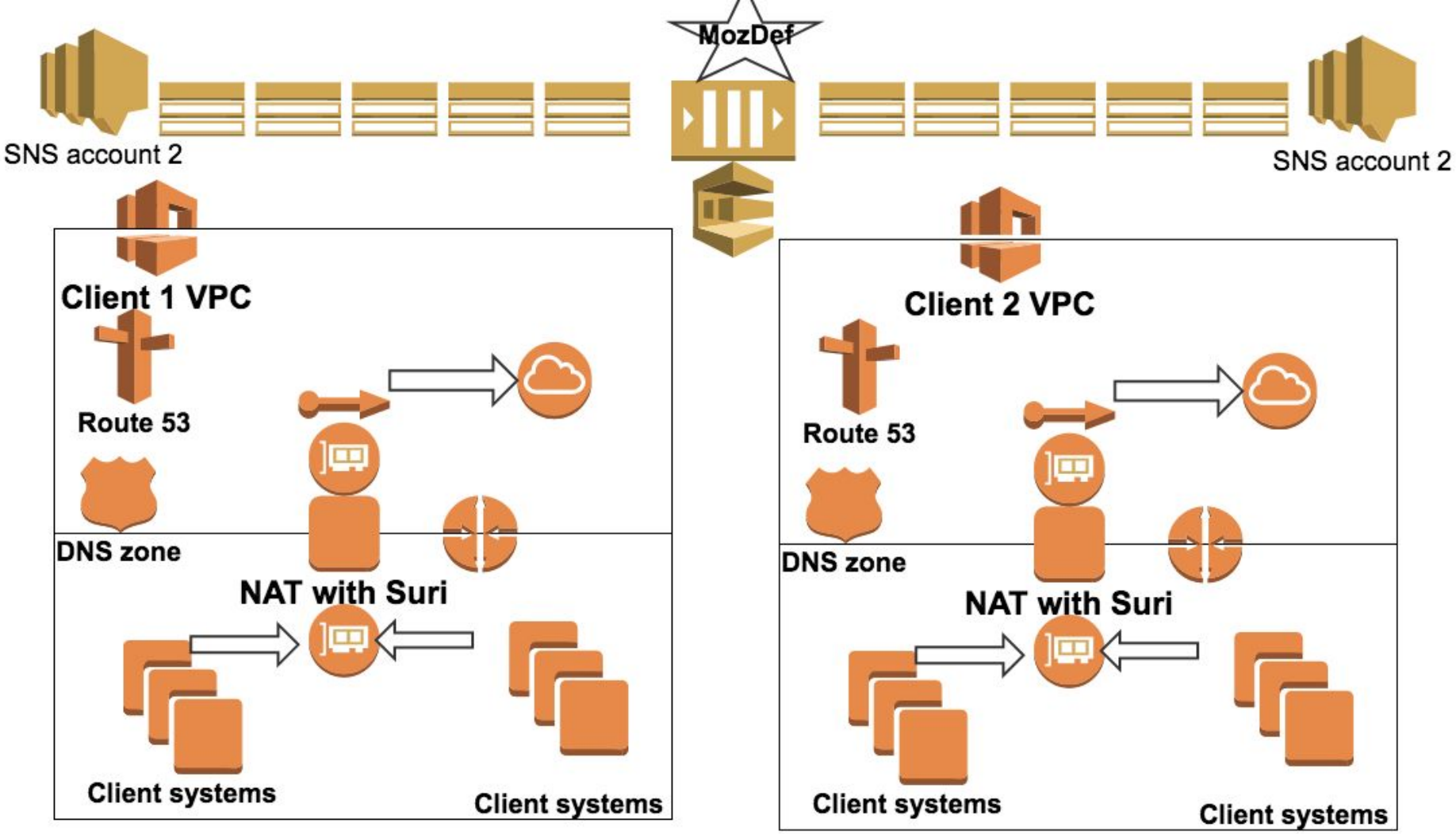
Some other things to worry about

Shipping logs back

Some other things to worry about

Credential management

Quick deployment and integration


MozDef
SNS account 2
SNS account 2
Client 1 VPC
Route 53
DNS zone
NAT with Suri
Client systems
Client systems
Client 2 VPC
Route 53
DNS zone
NAT with Suri
Client systems
Client systems

moz://a

You. A future cybercloud subject matter expert

You. A future cybercloud subject matter expert

Learn to love CloudFormation

Think about shipping logs

When life gives you lemons...

Ship all the logs all the time

That can easily be a few thousands eps / sec

From multiple accounts

From multiple VPC

Ship all the logs all the time

Suricata -> eve.json

Eve -> Syslog-ng/Fluent/whatever -> SNS

SNS -> SQS

SQS -> SIEM

You shall use MozDef

Your SIEM cannot read SQS and SNS

Mine can read both (and it's opensource :)

More cloud in the cloud

Why SNS?

No credentials to manage per Suricata/NAT-i

IAM does everything for you

Federated something something

SNS per account

NAT-Suri ships here

One SQS

(or more - does not scale and is slow)

SQS subscribed to SNS topic, receives messages

SQS. What a wonderful piece...

SQS. 500 eps only from a local instance

20ms latency kills it

Deleting is extremely slow. Do not delete. Expire.

Run into "too many messages in flight"

SQS. What a wonderful piece...

Writing 5000 eps to SQS from Fluentd

>30 processes

Reading 5000 eps from SQS

>50 processes

Haiku

NAT has a role

Role has permissions

To write to SNS

No credentials needed

Credentials management

What credentials?!

Your rules

Thresholds

Oinkcode :-)

Ansible, GPG, Credstash

It is almost 2018

No AWS, CF, Docker == no friends

One command to deploy it all

It is almost 2018

Ansible deploys multiple CF stacks

CF deploys VPC, EC2, IAM, Route53

CF passes “user-data” to your instance

It is almost 2018

Cloud-init runs user-data (base64...) with bash

Script installs ansible on EC2, fetches playbooks

Fetches secrets, decrypts, runs Ansible locally

playbooks

Cloudformation templates

1707

YAML lines

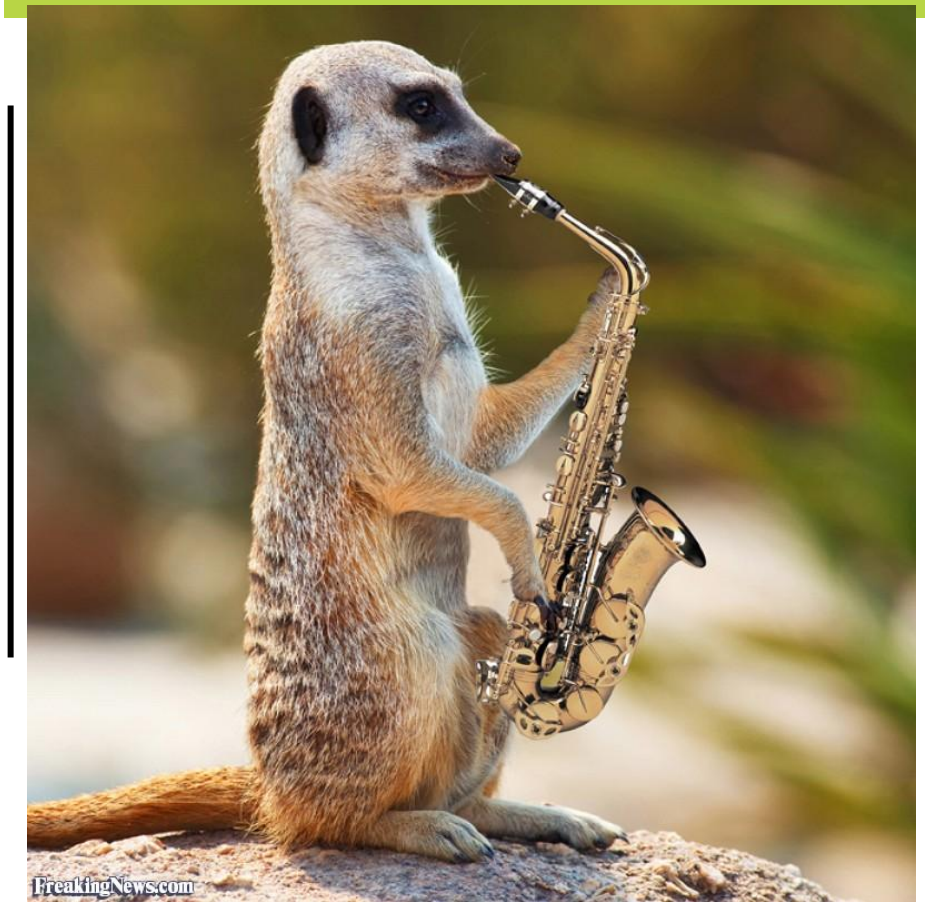
How to get friends and influence clouds

To deploy a small VM

With 2 pieces of software

Live demo 😱

VivaLaTrance



Some basic technologies used

Your future tools

1. Ansible
2. Cloudformation
3. VPC
4. EC2
5. Route53
6. IAM
7. SQS
8. SNS
9. Coffee
10. Non-alcoholic beverages
11. CHF. Lots.

Thank you

Gene Wood - **Insane patience**

My boss - **Open Mindness**

You know who you are - **free accommodation**

City of Zurich - **Guess where I made slides ;)**

Austrian OOB - for leaving us in the middle of nowhere, in snow

Linz

Thanks!

Michał Purzyński

Twitter @michalpurzynski

GitHub github.com/michalpurzynski

Mozilla mozilla.org