

Pentesting on steroids using performance monitoring

BSides Lisbon, 2022



DATADOG

Jean-Baptiste Aviat, Datadog

Staff Engineer for Security Products

 **sqreen** CTO & Co-founder

Former  (Red Team), pentester

Email: jb@datadoghq.com

Twitter: @jbaviat





What is ~~performance monitoring~~?



What is observability?

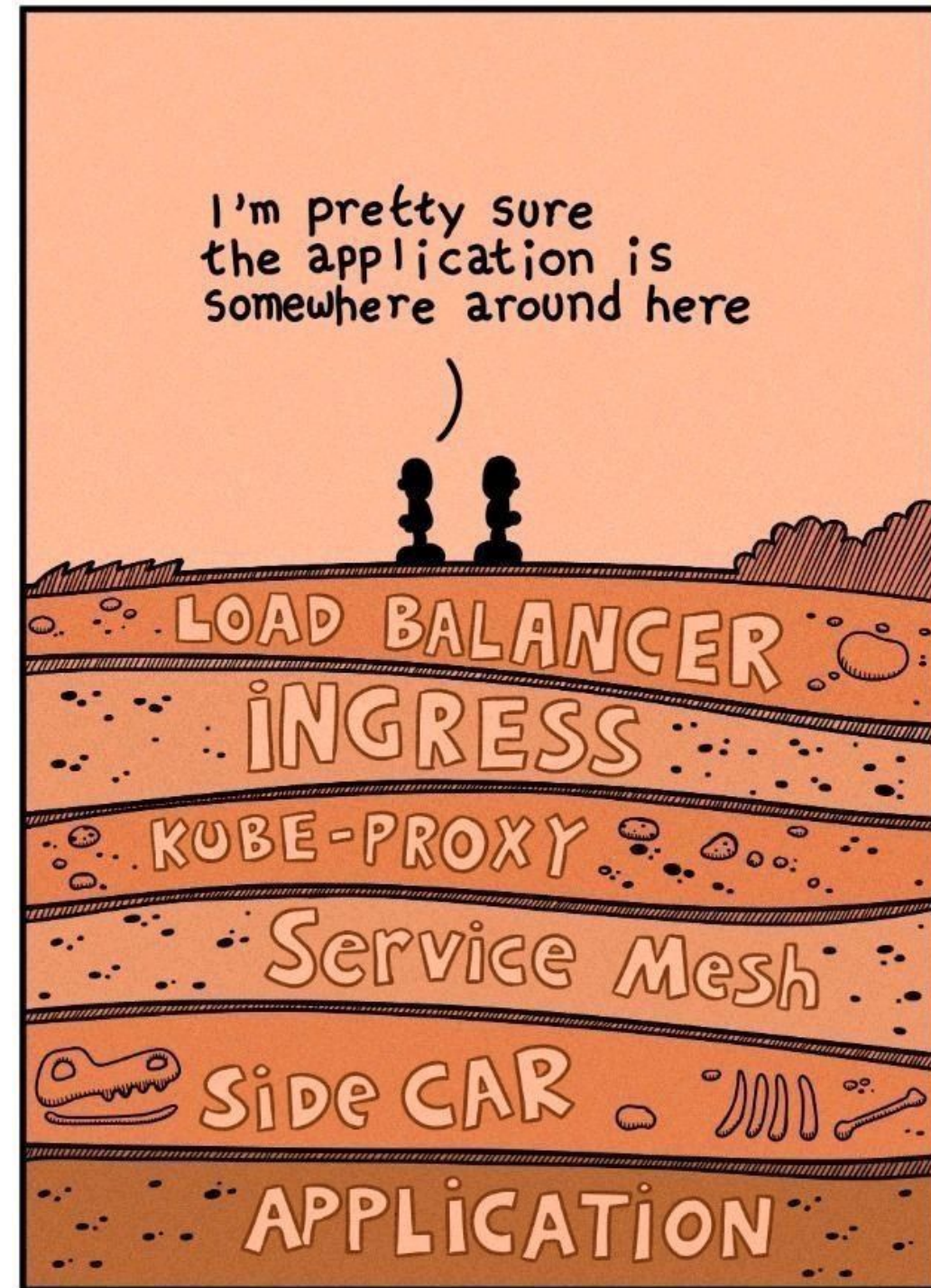
**And why do engineering
teams care about it?**

Building and maintaining complex systems

Systems engineers are dealing with grew in complexity

They need specific tools to help them...

- visualize
- understand the flows
- debug
- monitor deviations



Attacking complex systems

Lots the needs are similar with engineering needs:

- Accelerate recon
- Facts instead of hypotheses
- Observe systems internal response to stimuli
- Discover hidden components





How do we do this?

product engineers



How do we do this?

Modern observability products give *full stack* visibility

Data plane

Container orchestrator

Hosts & containers

Applications

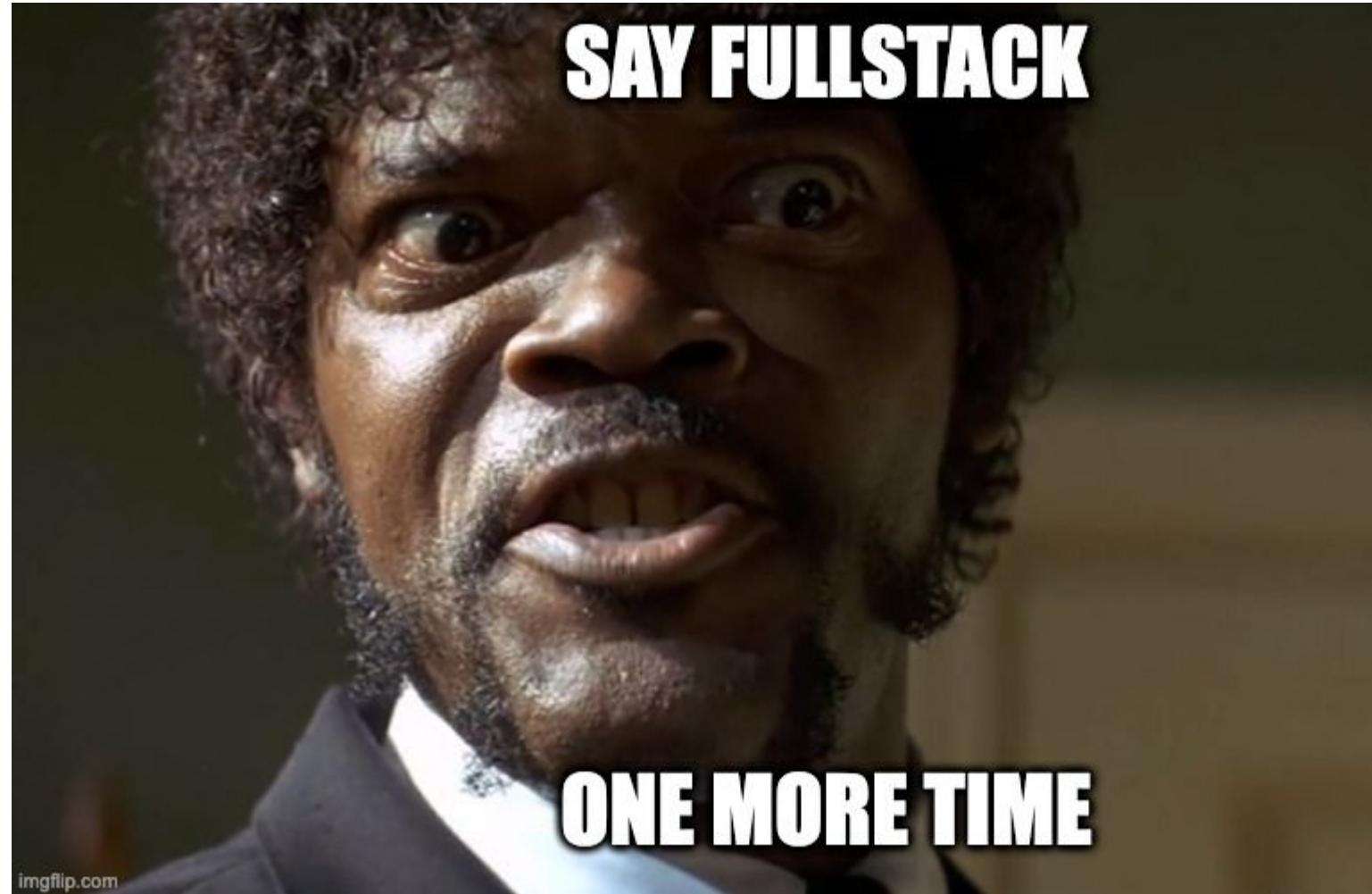
Datastores

Cloud control plane

IAM users & permissions

Security settings

Network configuration



Many paths to observability

Many products: open source or commercial

Disclaimer: I talk about Datadog - but assume similar capabilities from most other products

OPEN SOURCE

 Grafana

 Prometheus

 graphite

 ZABBIX


DATADOG

 APPDYNAMICS

 dynatrace

SignalFx

Nagios

 **BROADCOM**



Amazon Cloudwatch



New Relic

sumo logic



LIGHTSTEP

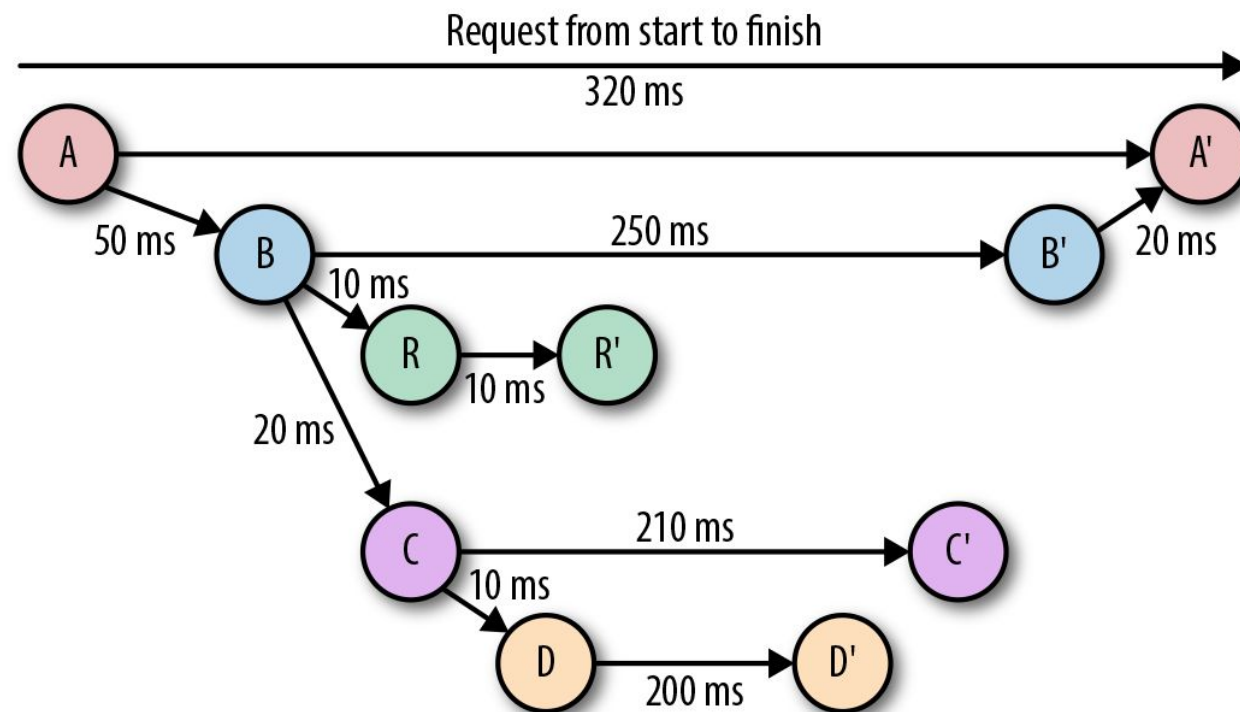
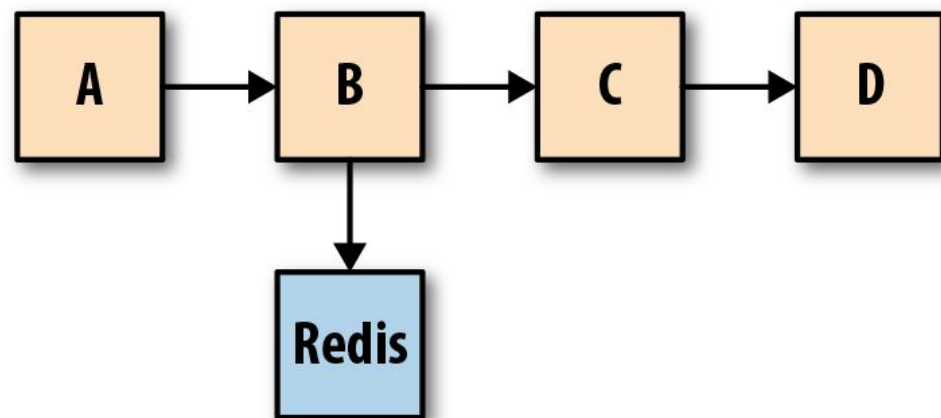
solarwinds 

INSTANA

+ others

Demo of an observability product

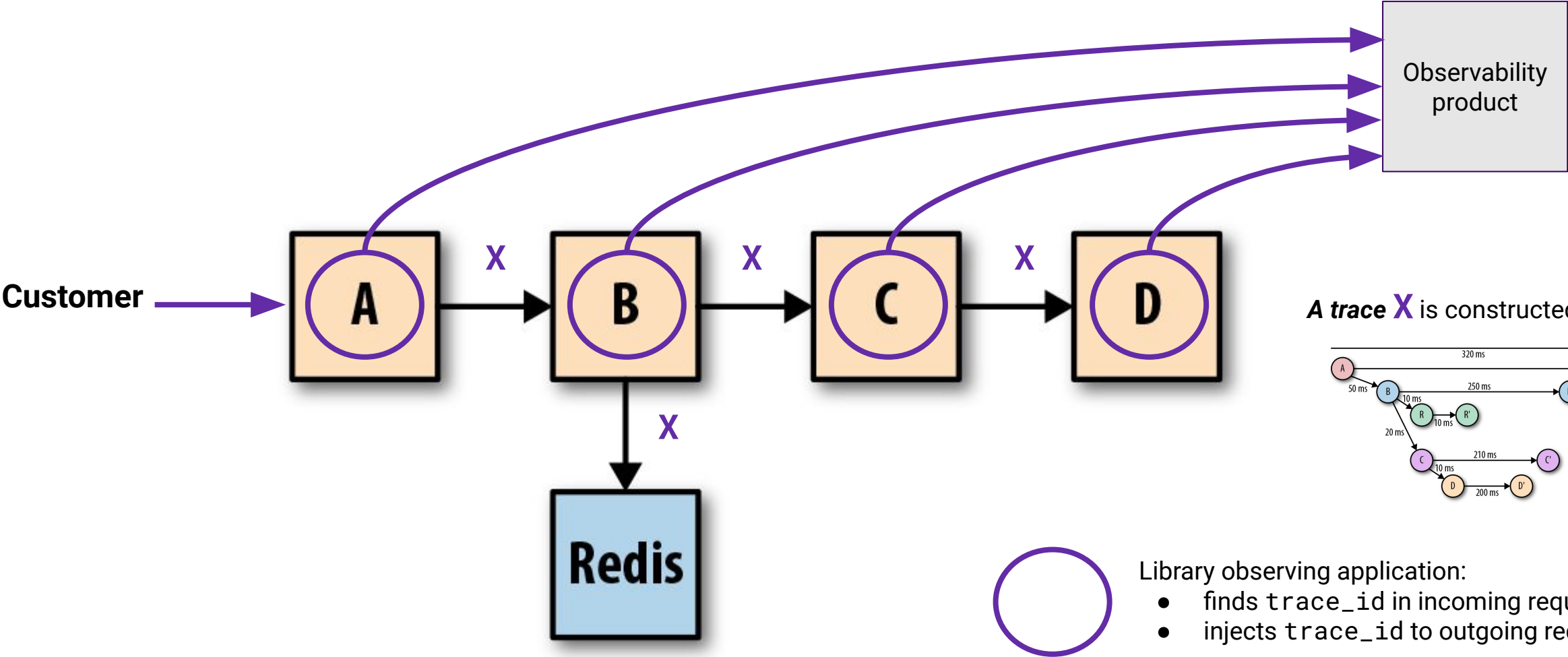
How does distributed tracing work?



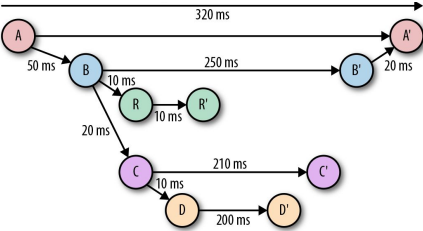
How do APM work?

In-app libraries injecting and propagating an ID

Data centralization



A trace X is constructed:



Library observing application:

- finds `trace_id` in incoming requests
- injects `trace_id` to outgoing requests

X

`trace_id`: random number identifying a **trace**

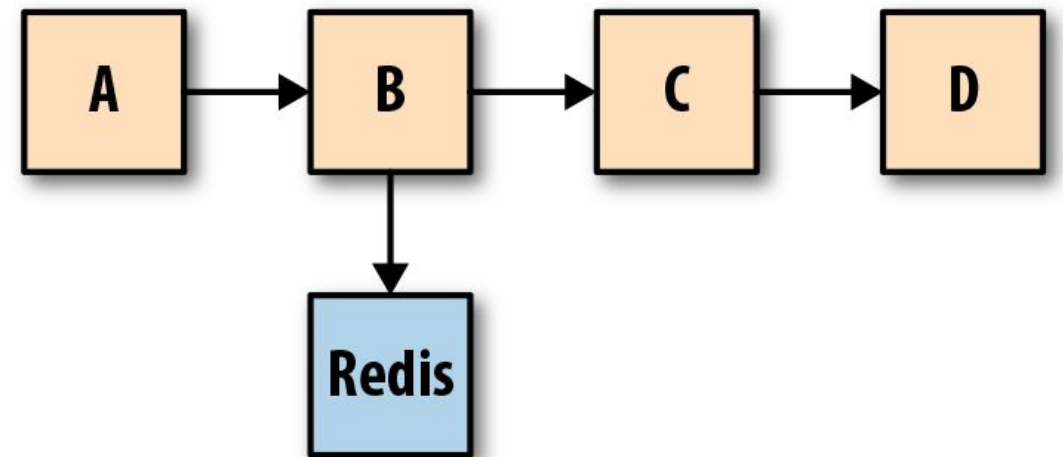
APM is magic!!



APM limitations

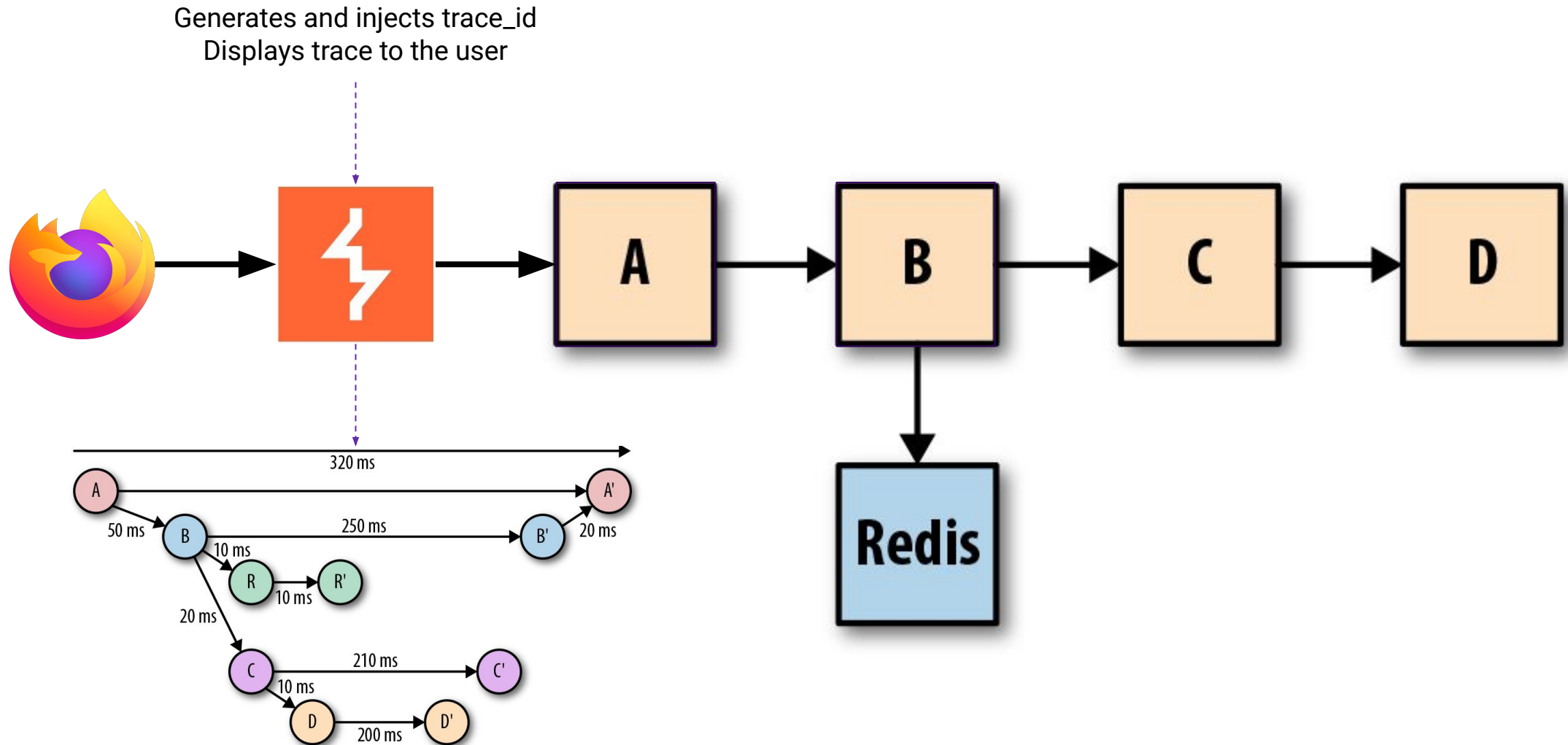
Need to support lots of libraries: if $B \rightarrow C$ is gRPC, gRPC support is mandatory not to lose C and D

Sampling: ingesting and indexing traces is costly, hence sampling is applied



Pentesting with observability

Turn Burp Suite into an APM library



What is Burp Suite?

Repeater Sequencer Decoder Comparer Extender Project options User options

Dashboard Target Proxy Intruder

Intercept HTTP history WebSockets history Options

Filter: Hiding CSS, image and general binary content

# ^	Host	Method	URL	Params	Edited	Status	Length	MIME type	Exten
7	https://update.googleapis.com	POST	/service/update2/json?cup2key=10:1...	✓		200	14648	JSON	
8	http://redirector.gvt1.com	GET	/edgedl/release2/chrome_component/...			302	1053	HTML	
12	https://portswigger-labs.net	GET	/index_files/jquery-2.js			200	85908	script	js
14	https://portswigger-labs.net	GET	/index_files/portswigger-logo.svg			200	8309	XML	svg
15	https://portswigger-labs.net	GET	/index_files/ps-mobile-logo.svg			200	963	XML	svg
17	https://portswigger-labs.net	GET	/Content/Fonts/DroidSans/s-BiyweUP...			200	21722		woff2
18	https://update.googleapis.com	POST	/service/update2/json	✓		200	1026	JSON	
20	http://redirector.gvt1.com	GET	/edgedl/release2/chrome_component/...			302	1023	HTML	
22	https://update.googleapis.com	POST	/service/update2/json	✓		200	1026	JSON	
23	http://redirector.gvt1.com	GET	/edgedl/release2/chrome_component/...			302	1067	HTML	
25	https://update.googleapis.com	POST	/service/update2/json	✓		200	1026	JSON	
26	http://redirector.gvt1.com	GET	/edgedl/release2/chrome_component/...			302	1027	HTML	
28	https://update.googleapis.com	POST	/service/update2/json	✓		200	1026	JSON	

Request

Pretty Raw \n Actions ▾

```
1 GET /index_files/ps-mobile-logo.svg HTTP/1.1
2 Host: portswigger-labs.net
3 Connection: close
4 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64)
  AppleWebKit/537.36 (KHTML, like Gecko)
  Chrome/87.0.4280.88 Safari/537.36
5 Accept:
  image/avif,image/webp,image/apng,image/*,*/*;q=0.8
6 Sec-Fetch-Site: same-origin
7 Sec-Fetch-Mode: no-cors
```

Response

Pretty Raw Render \n Actions ▾



```
1 HTTP/1.1 200 OK
2 Date: Wed, 03 Feb 2021 09:55:06 GMT
3 Server: Apache/2.4.41 (Ubuntu)
4 Upgrade: h2
5 Connection: Upgrade, close
6 Last-Modified: Fri, 29 May 2020 10:53:20 GMT
7 ETag: "2b1-5a6c740e3a67e"
8 Accept-Ranges: bytes
9 Content-Length: 689
10 Content-Type: image/svg+xml
```

INSPECTOR


Pentesting x Observability: SQL injection

Full visibility into SQL and triggered errors


Accelerate SQL injection exploitation by accessing the full query and generated exceptions.

 hsqldb hsqldb.query >  select userid from sql_challenge_users where userid = ?

Span: Info **Errors 1** Infrastructure Metrics Logs 0 Code Hotspots

 hsqldb | select userid from sql_challenge_users where...

Parsed Raw

 **java.sql.SQLException: malformed string: 'tom'**

[View similar errors](#) | This issue is 2 minutes old - Last seen 2 minutes ago

java.sql.SQLException: malformed string: 'tom'

at org.hsqldb.jdbc.JDBCUtil.sqlException(Unknown Source)

at org.hsqldb.jdbc.JDBCUtil.sqlException(Unknown Source)


at org.hsqldb.jdbc.JDBCStatement.fetchResult(Unknown Source)


Pentesting x Observability: SSRF


Pentesting x Observability: SSRF


Understand the actions performed by a endpoint.
Here, no network access limit the exploitability of this SSRF.


Trace: Flame Graph **Span List 9** Map

 Web


 DB


 Cache


 *fx* Function


 Custom


RESOURCE


>  hsqldb

▼  webgoat

>  UserTrackerRepository.findByUser

>  JpaRepository.saveAndFlush

>  SSRFTask1.completed

>  POST /SSRF/task1

Pentesting x Observability: code insights

Pentesting x Observability: Code Insights

See what libraries are actually running, and in which paths.

Filter flame graph

Options

CPU Time, per minute: 2.3s (Click to reset)



CPU Time by Library

Standard Library	1.42s
Datadog	411ms
Runtime	128ms
Spring Framework	91ms
<unknown>	73ms
OkHttp	61ms
Undertow	20ms
Hibernate	20ms
Spring Request Handling	19ms
Jackson	16ms
Okio	12ms
Reflection	11ms
Moshi JSON	8ms
LZ4 Compression	5ms
Jboss	4ms

Who can use this?

Red teams

- leverage existing internal observability products
- discover blind spots - what's not monitored is the perfect place to build persistence

Pentesters: open box testing

- ask access to customers
- great support to better understand an architecture

Pentesters: closed box testing

- stealing an access to an observability tool is **very** interesting :)

And what is coming/could come next?

Link profiles to traces and then to URLs

Get information about vulnerable libraries

Vulnerability detection

Source code integration

Open source alternatives

Open Telemetry

Prometheus

Elastic



Resources and links

Questions?



<https://aviat.github.io/2022-bsideslisbon.html>

Datadog is hiring! In remote, Lisbon, Paris, NYC, San Francisco...
<https://securitylabs.datadoghq.com/#work-with-us>