
AWS
re:Invent
NOV. 28 – DEC. 2, 2022 | LAS VEGAS, NV

BOA204

When security, safety, and urgency all matter: Handling Log4Shell

Abby Fuller

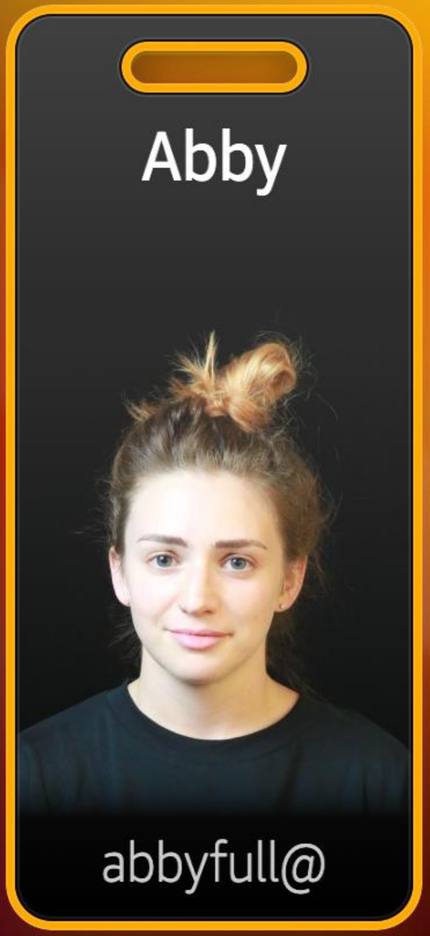
Security @ EC2
AWS

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

I’m Abby, I’m a Senior Principal Security Engineer at AWS.

I mostly work on security for Amazon EC2.

Along with a handful of other security-minded Principal Engineers, I also work on a cross-AWS team we’ve called “Ghostbusters.”



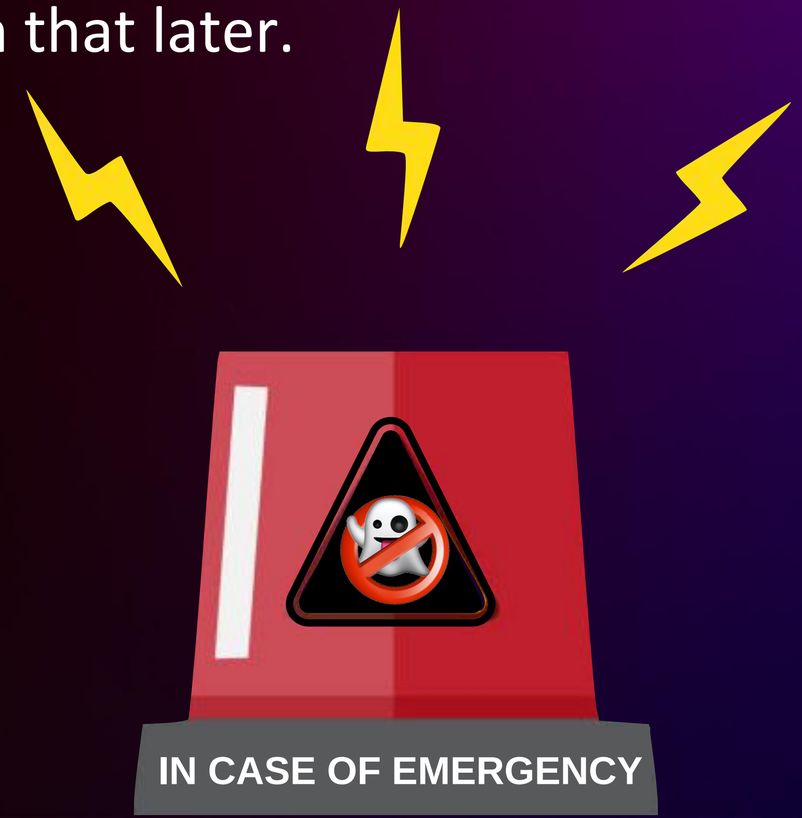
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

We use Ghostbusters as a security big red button.

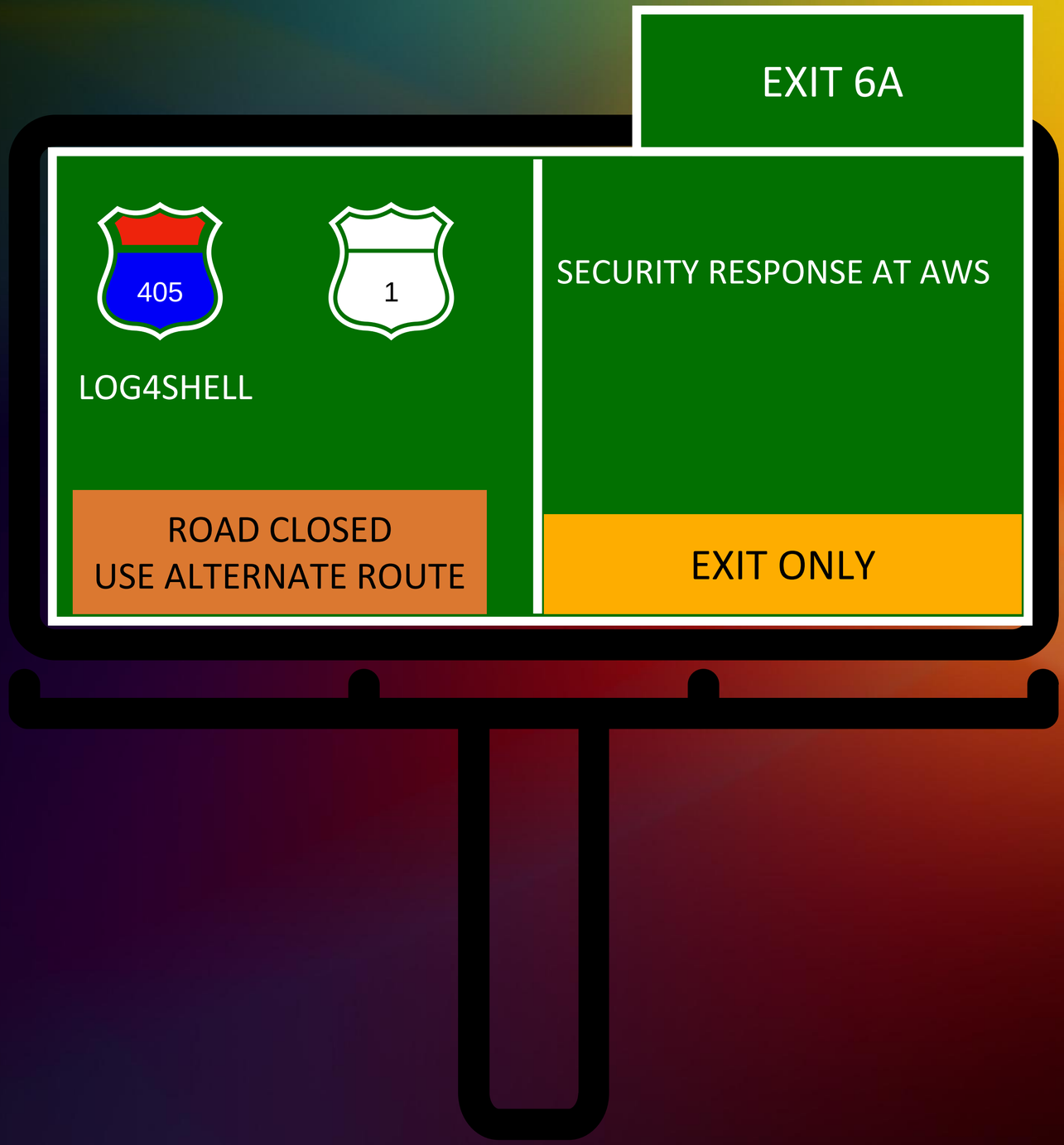
Title

We handle escalations, we help assess impact and blast radius, and work with service teams and AWS Security.

In events like Log4Shell, we help lead the response. More on that later.



© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.



© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

At AWS, most event response takes place on conference calls

That process looks something like this:

TOS CALL
"Tech Operations Call"



Focused on mitigation and recovery

DASHBOARD CALL
"The everything else call"



Focused on service health and customers

aws
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

AWS Security Operations

Under ordinary circumstances, security response at AWS is handled by Security Operations (SecOps).

SecOps is global, and is staffed 24/7. Shift rotations follow the sun, so you always get someone who is awake, during their normal daytime hours.

SecOps is engaged via ticket on a potential security issue. This can be by a service team, customer, reporter tooling, etc.

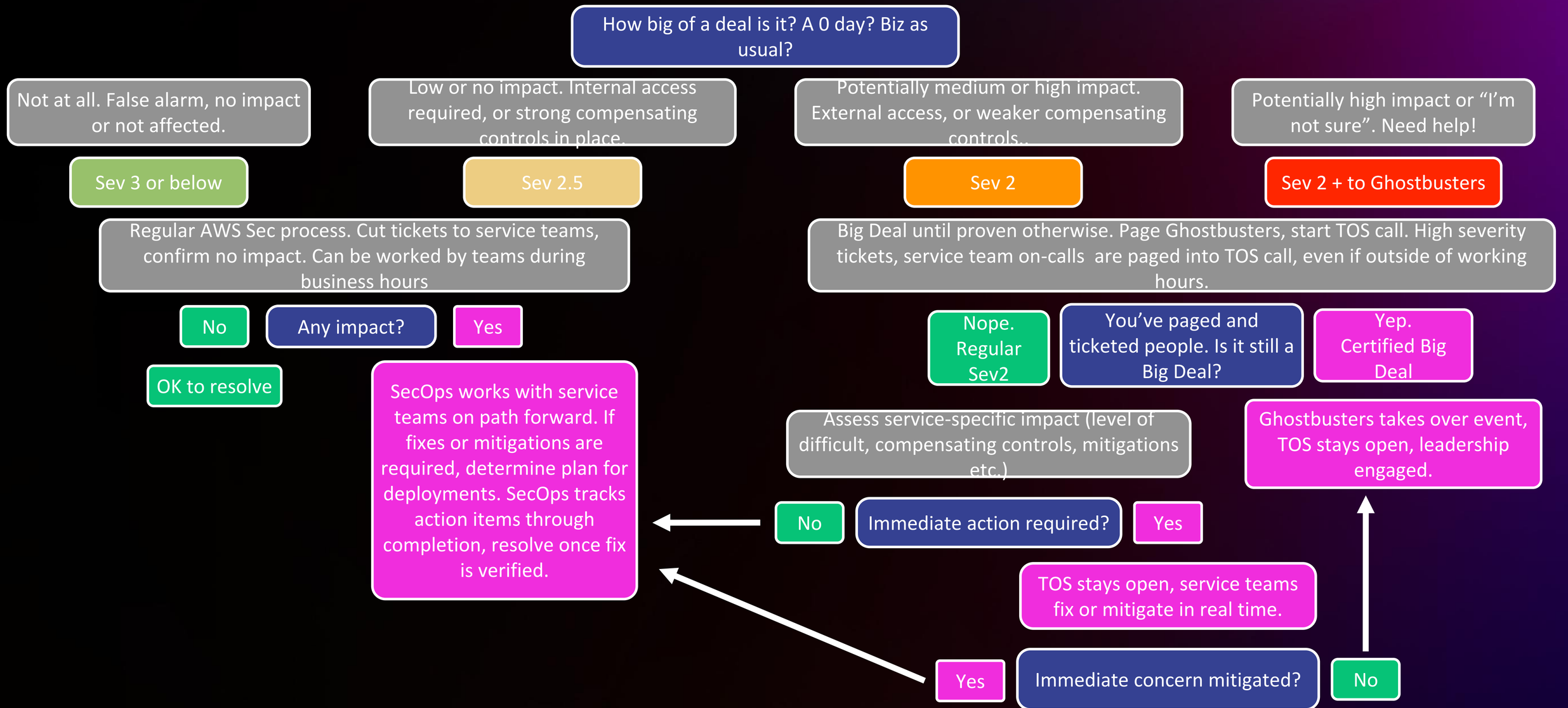
Ticket is triaged by SecOps, SecOps engages escalation (like Ghostbusters) if needed.

SecOps engages service team, SecOps and service team align on risk, urgency, proposed mitigations or fixes, and any communication.

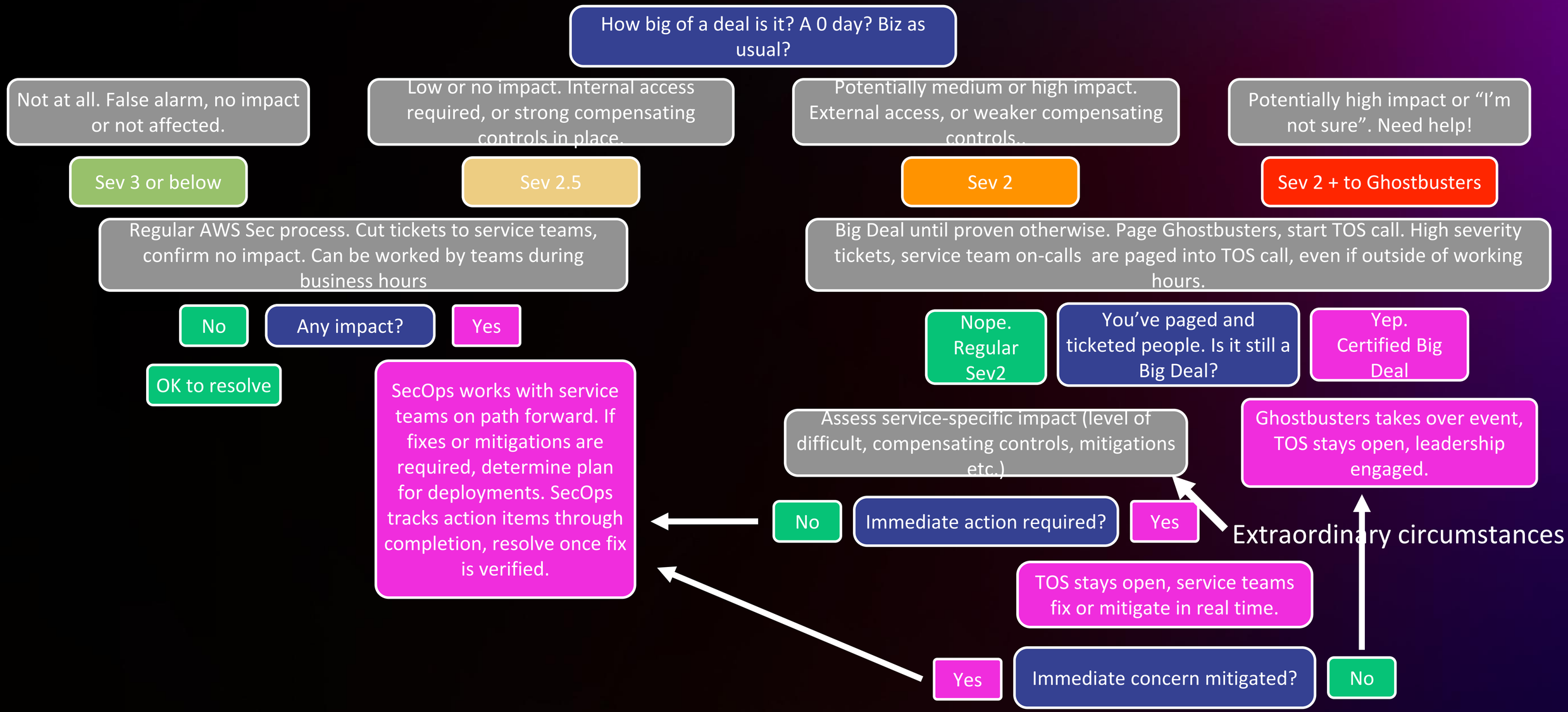
Service team completes any mitigations, leadership engagements, and messaging. SecOps tracks all action items through to completion.

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Potential events are treated as serious until proven otherwise



© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.



© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Ghostbusters takes over event, TOS stays open, leadership engaged.

MITIGATION/RECOVERY

Main TOS call

Is a fix available?

Yes

No

Engage SMEs to develop a fix.

Is a fix available?

Yes

No

Test proposed fix.

Fix ready?

Yes

No

Engage Builder Tools to start builds, engage large fleets to start preparing for deployments.

Builds ready? Fleets ready?

COORDINATION

Main TOS call + Slack

Engage teams to assess accurate blast radius. Determine how to track and report on turndown. Start regular leadership syncs.

Coordinate communication with broader service team leadership. Set priority and expectations: Should teams expedite deployments? Should they pause other work?

Prep for large campaign; finalize blast radius, ticketing plan, deployment plan, order of engagement for teams, clear and tested direction for on-calls, support and staffing plans.

Campaigns and communications prepped? Everything ready?

COMMUNICATION

Dashboard call

Post initial security bulletin and/or internal notices. Prep for field and customer communications.

GO TIME! Start deployment waves, kick off campaign and communications.

aws

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

This isn’t the first time we’ve had to respond quickly.

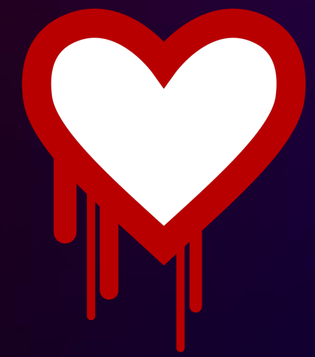
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

April 7, 2014: Heartbleed



Heartbleed was a memory disclosure issue with the OpenSSL implementation of Heartbeats. Heartbeats were designed for things like keep_alive, and allow a sender to request a specific response (a string, plus the strings length as an integer) from a receiver. The OpenSSL issue returned the payload, plus the length, plus the memory buffer associated with the length. There was no bounds verification around the payload length – you could get up to 64 kb.

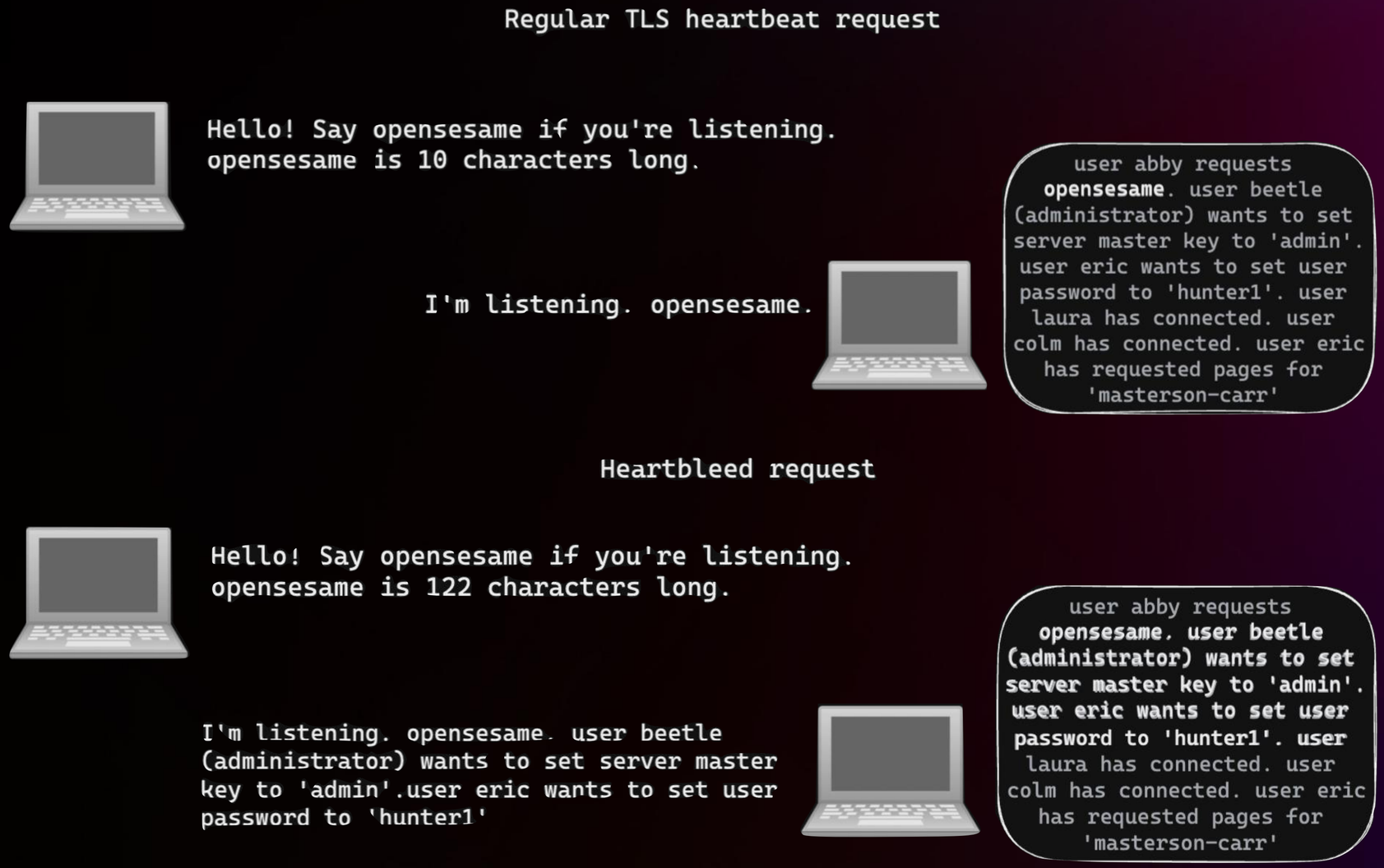
This let you request a payload with a very large length, and receive the contents of the corresponding memory buffer. Since this was likely memory that OpenSSL had used previously, you could get private keys, usernames and passwords, certificates, etc., and continue to eavesdrop on your target until they patched the bug and rotated all the materials.



aws

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Simplified, Heartbleed looks something like this:



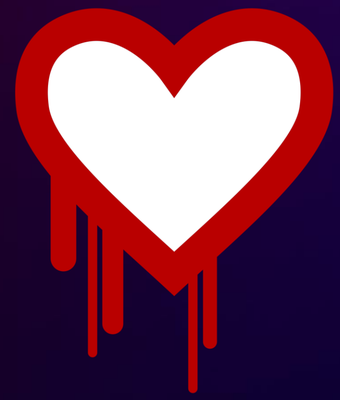
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Patching Heartbleed was a mad dash for the internet

For Heartbleed, we hotpatched. Two line fix, millions of deployments, patched in hours. The night of Heartbleed, we scanned the *entire* Amazon EC2 IP space so we could send emails to every potentially impacted customer. Benevolent chaos.



colmmacc@ 2019 thread on Heartbleed



© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Over the last decade, the industry has had an assortment of security events of various sizes. Most, however, have required specialized knowledge or configurations to exploit. Heartbleed (and log4j) were special in that they were easy to take advantage of. No micro architectural knowledge required.



aws

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

So far, we've only covered the reactive part of security: an issue comes in, and we respond.

But there is a proactive element, too. Both in predicting what issues might be coming, and in identifying when issues are being exploited.



aws

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

How can we predict when (or if) something will happen?

More often than not, we can comb through different sources and signals to find potential predictors of what might be coming:



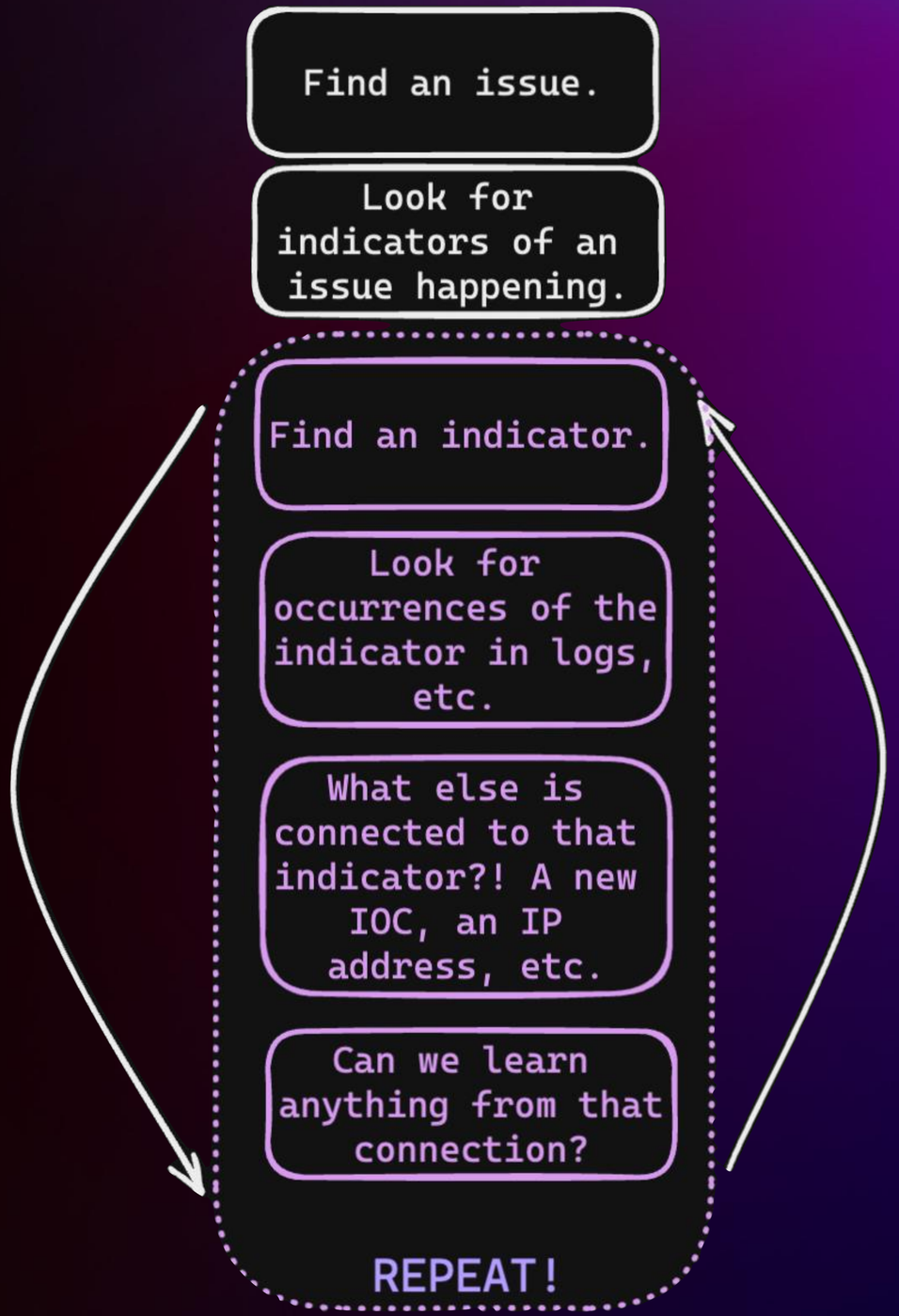
aws
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

From there, it’s basically detective work with a constant feedback loop

Look through sources, find something interesting, pivot, repeat. This is largely how things work in both ordinary, every day circumstances where we are trying to predict what might happen, and in events like Heartbleed and Log4Shell, where we might be trying to predict if an issue has been exploited.

The different data sources is where things get interesting.





aws
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Log4Shell day 0: what IS log4j anyway?



© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Our cast of characters

Apache Log4j: a logging implementation for Java

Developers can log different levels of data from inside their applications. This includes things like user input, or HTTP requests.

Log4j v1: original implementation, EOL in 2015, but still widely used in Linux distributions like RHEL, Amazon Linux, etc.

Log4 v2: re-write, released in 2014. Not API-compatible with v1, which contributed to wider-than-preferred used of v1 post-EOL.



JNDI: Java Naming and Directory Interface

API that lets Java applications lookup Java objects and names at runtime.

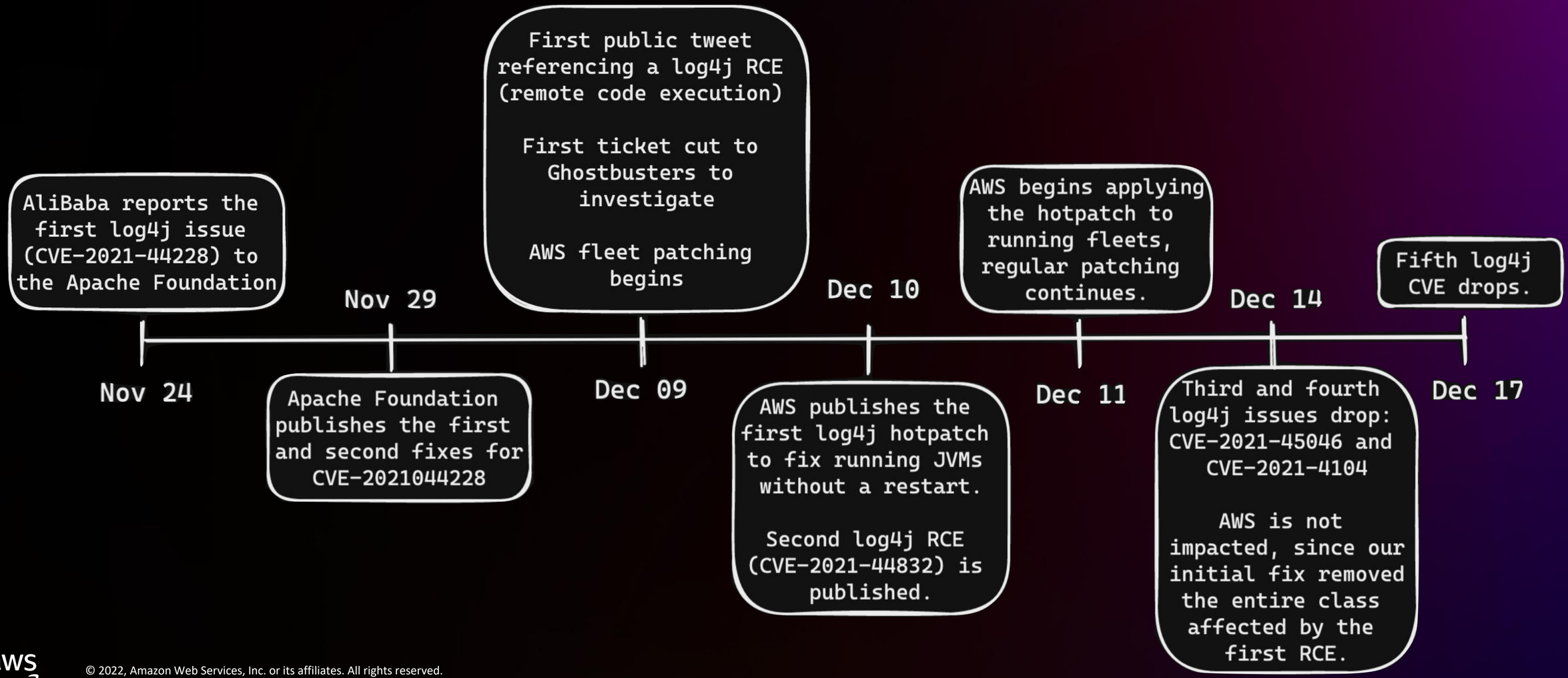
The JDK provides a few directory types out of the box, like DNS and LDAP.



aws

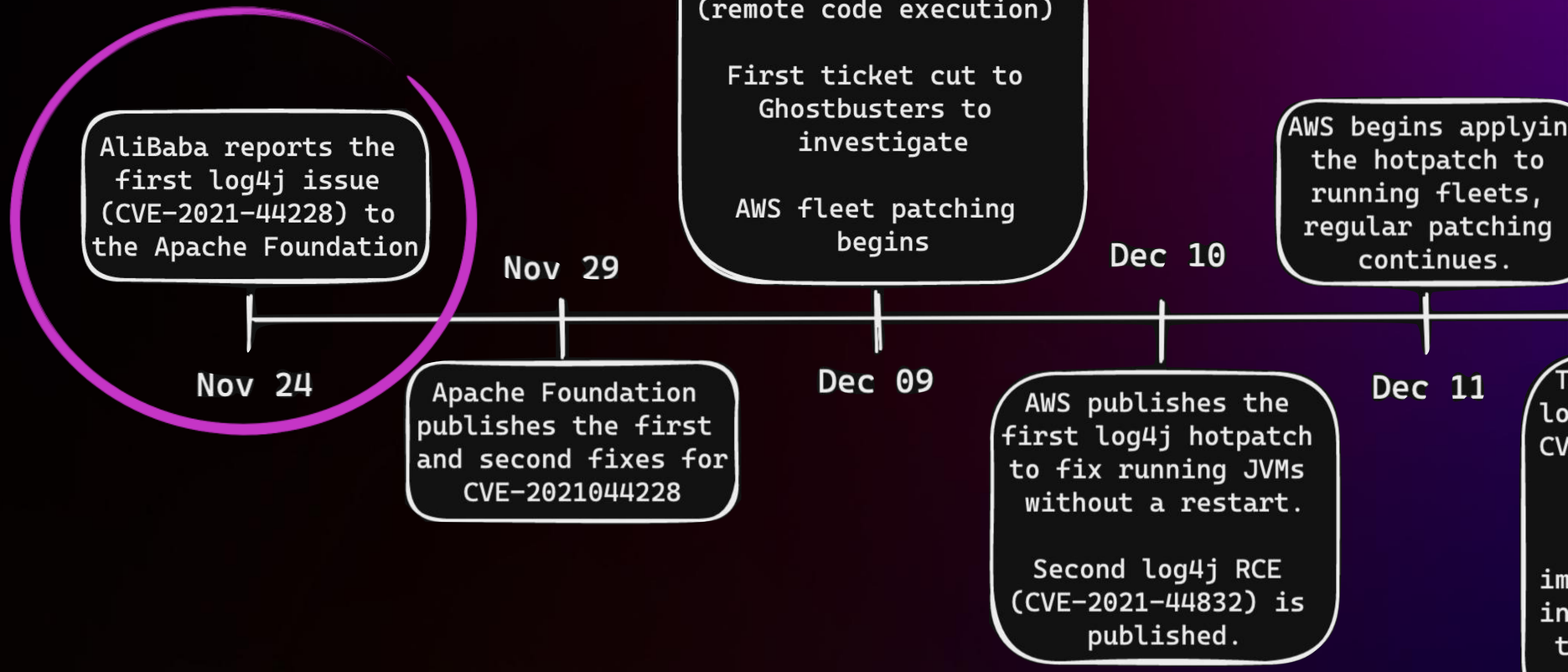
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Log4Shell™



© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Log4Shell™



© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

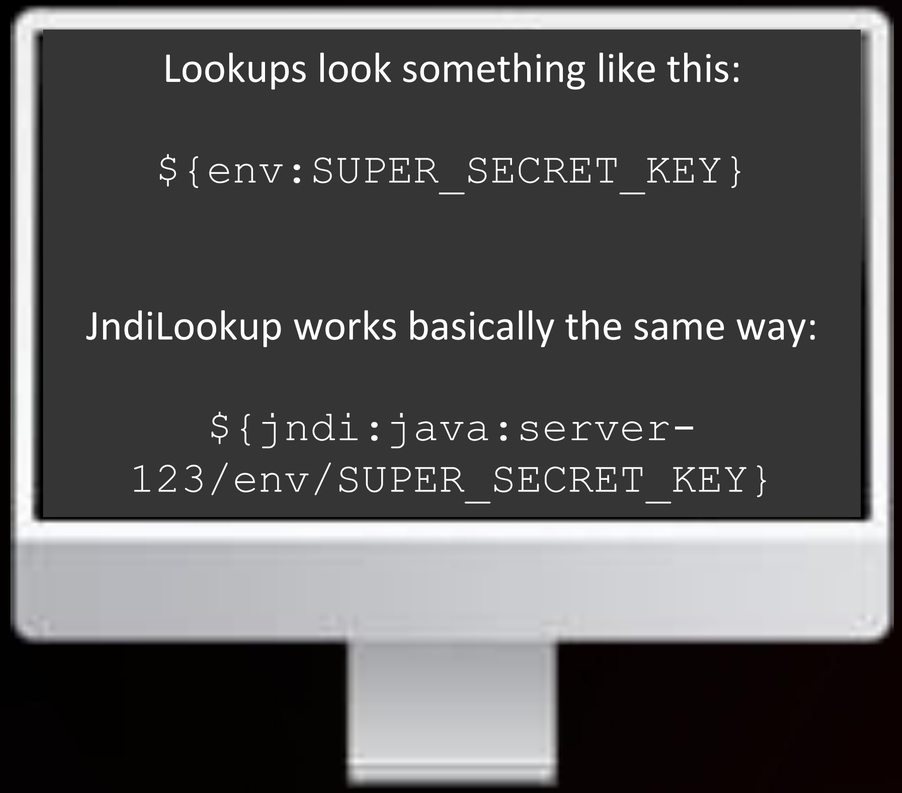
“JNDI features used in configuration, log messages, and parameters do not protect against attacker controlled LDAP and other JNDI related endpoints. An attacker who can control log messages or log message parameters can execute arbitrary code loaded from LDAP servers when message lookup substitution is enabled.”

CVE-2021-44228 (also known as “Log4Shell”)

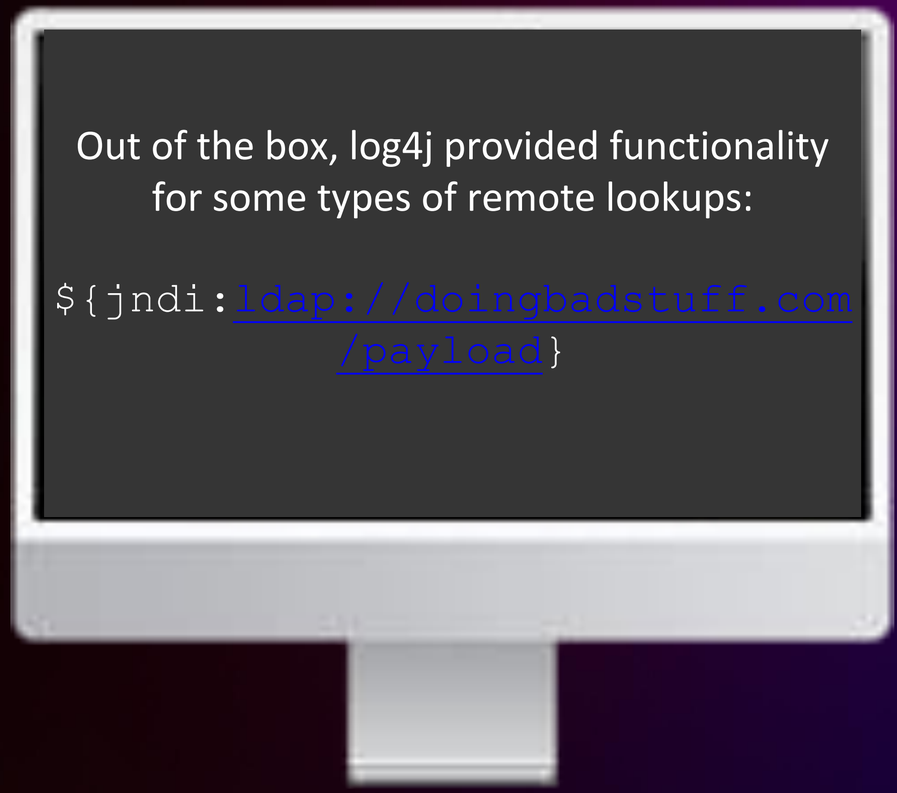
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Tl;dr: remote code execution via lookups. Log4j allows for users to pass input (like http requests) that will get evaluated (“looked up”) and logged at runtime.

With Log4Shell, those inputs were not properly validated.

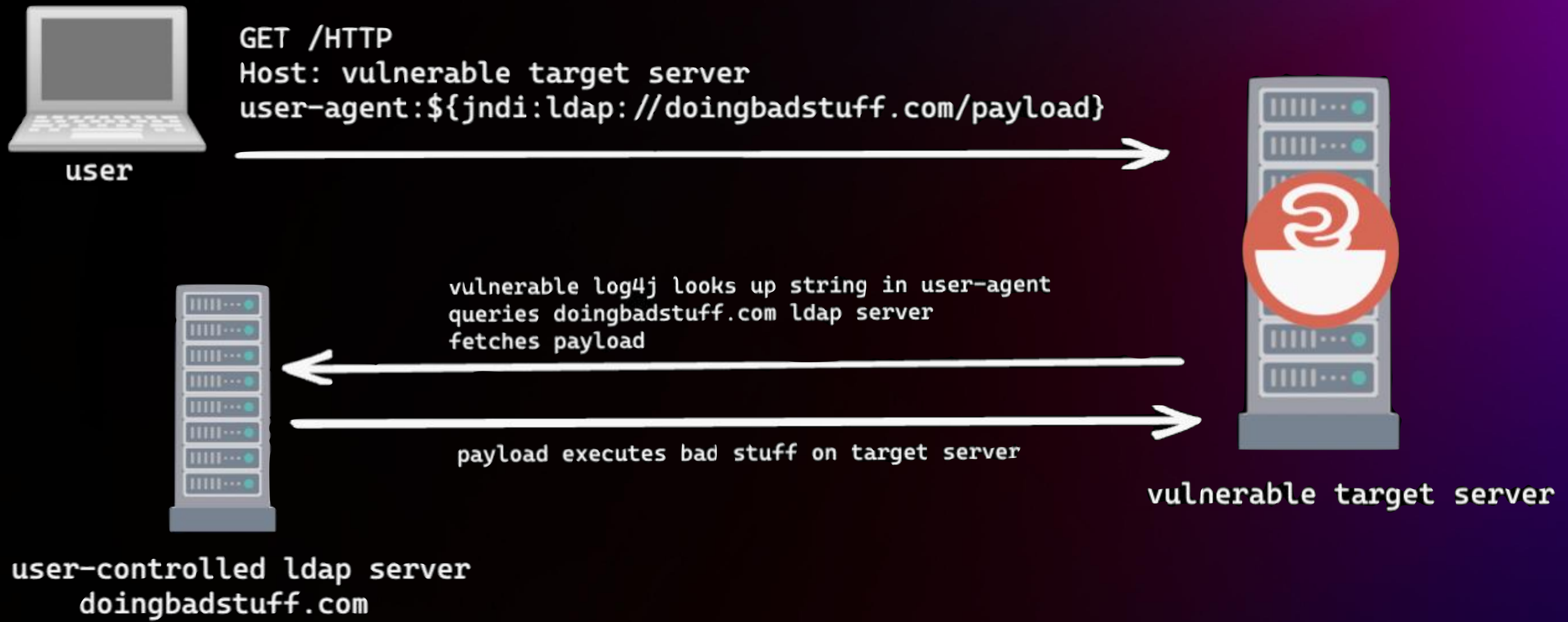


Regular lookups



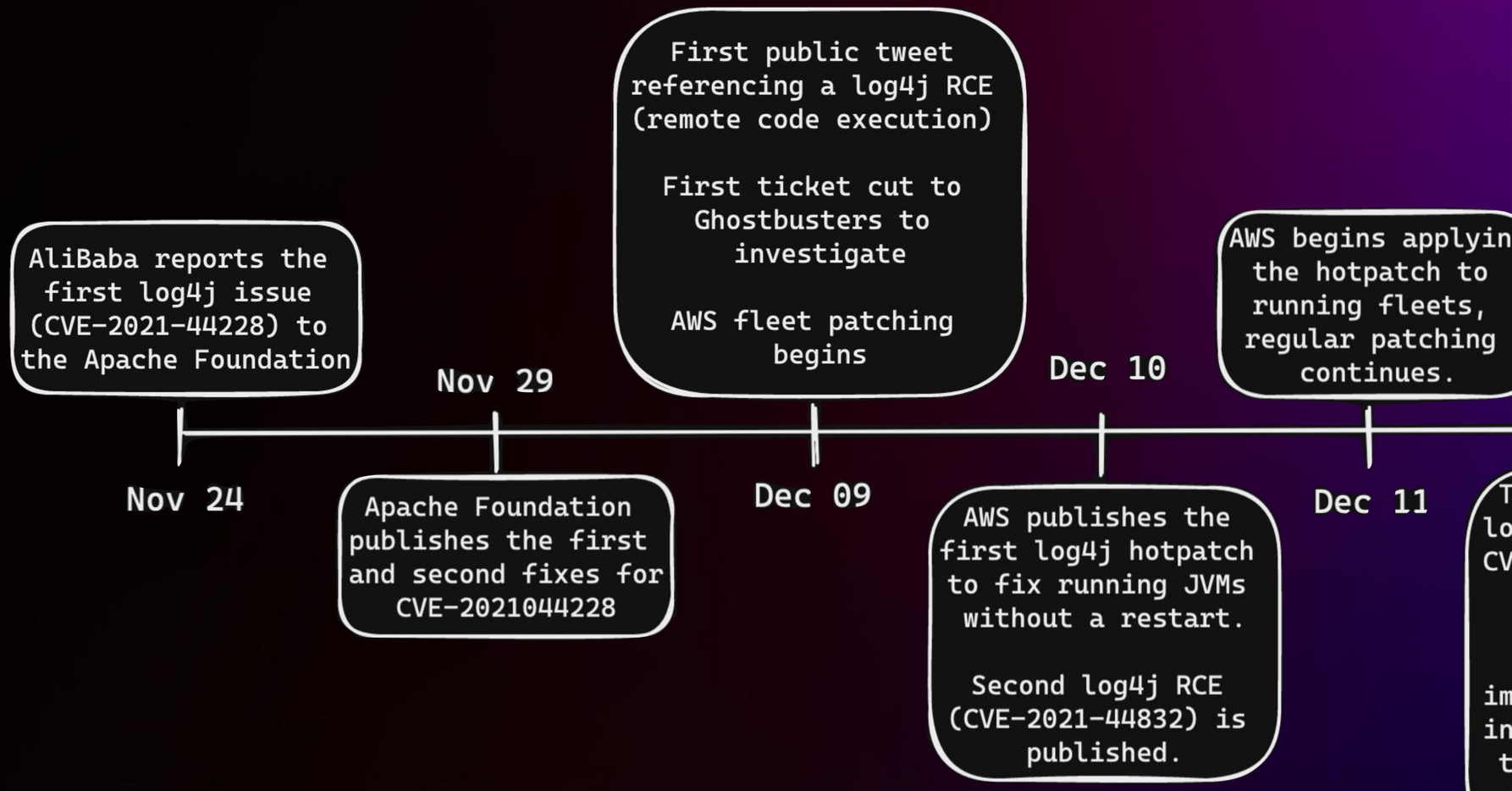
JndiLookup

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.



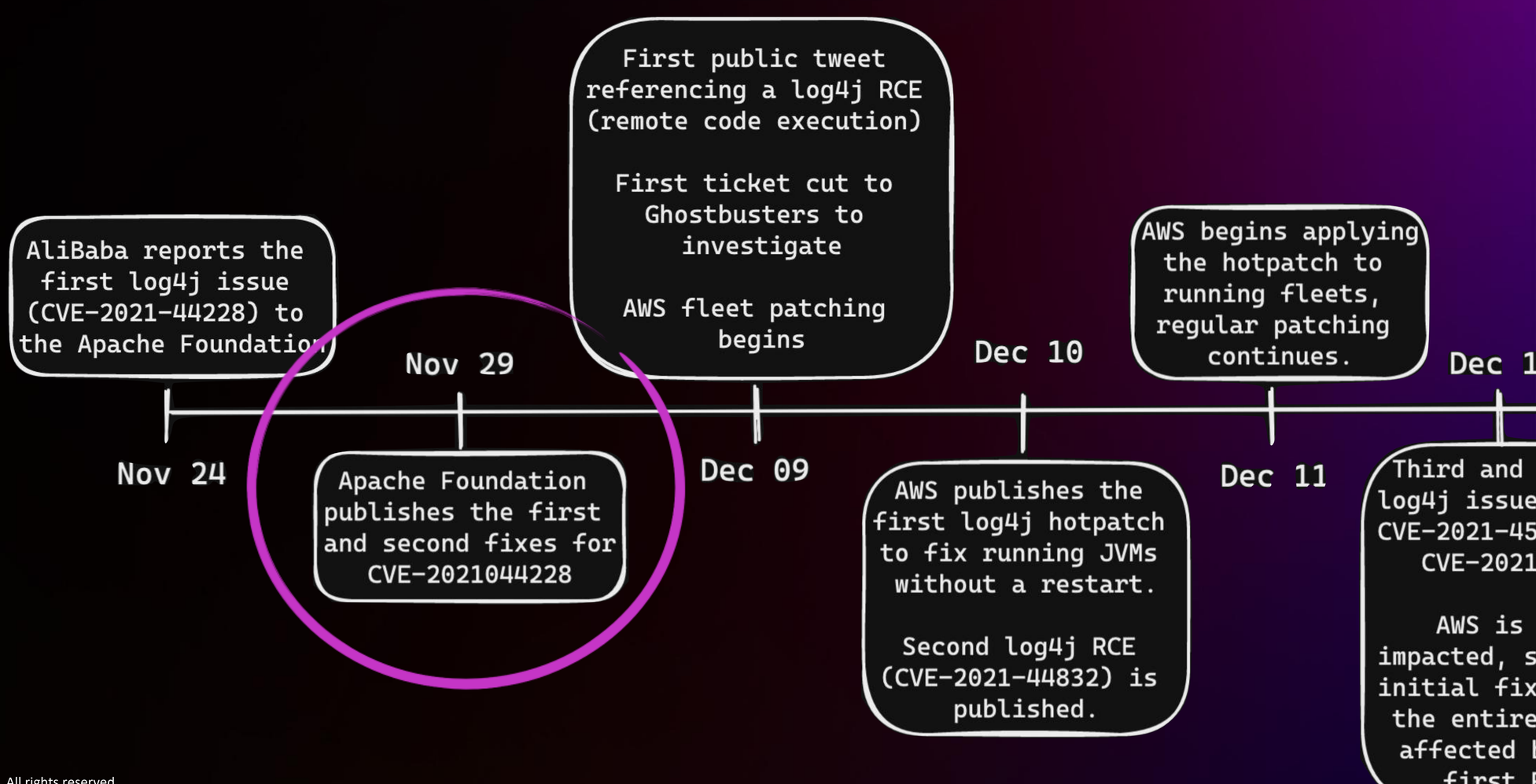
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Log4Shell™



© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Log4Shell™

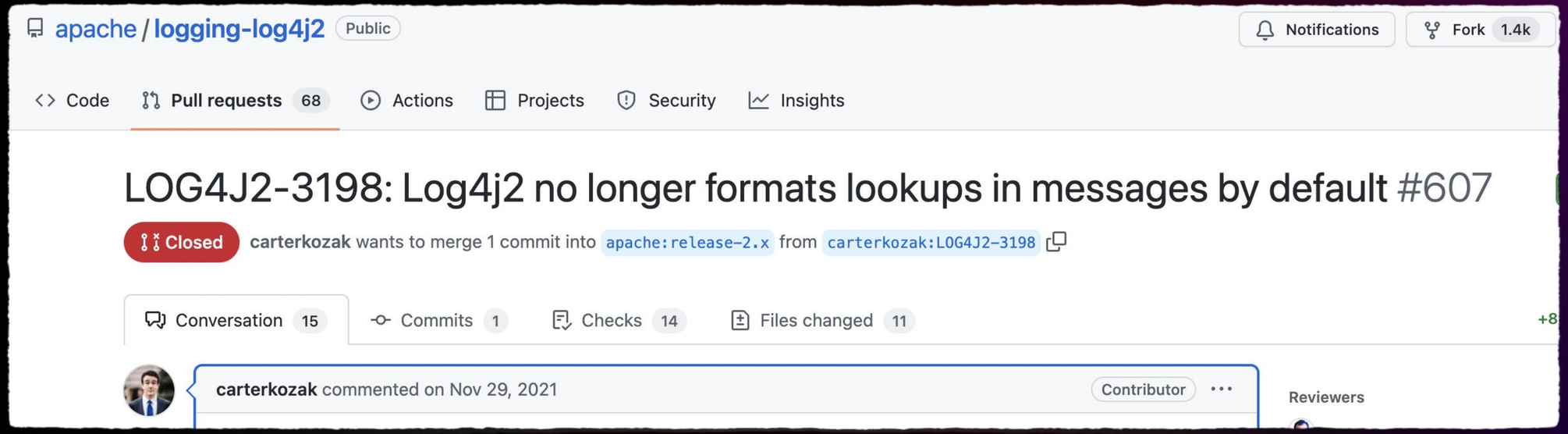


© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

The Apache Foundation publishes the first (and second) fixes to the log4j repos.

NOV 29

In the best tradition of patches for security issues, the first commit message is innocuous:



The original fix PR

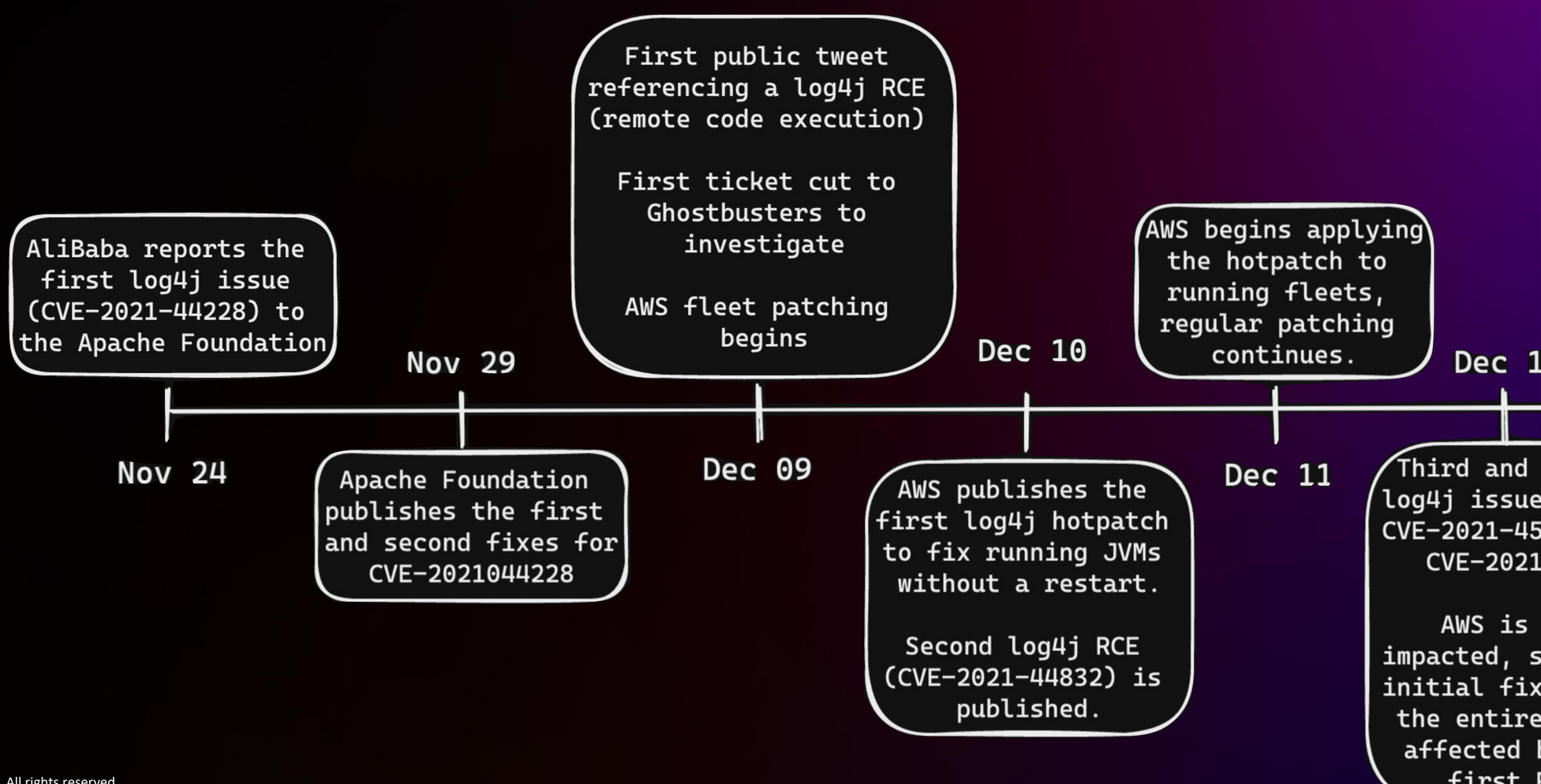
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

From November 29th until the first tweets start appearing on December 9th, the log4j issues fly mostly under the radar.

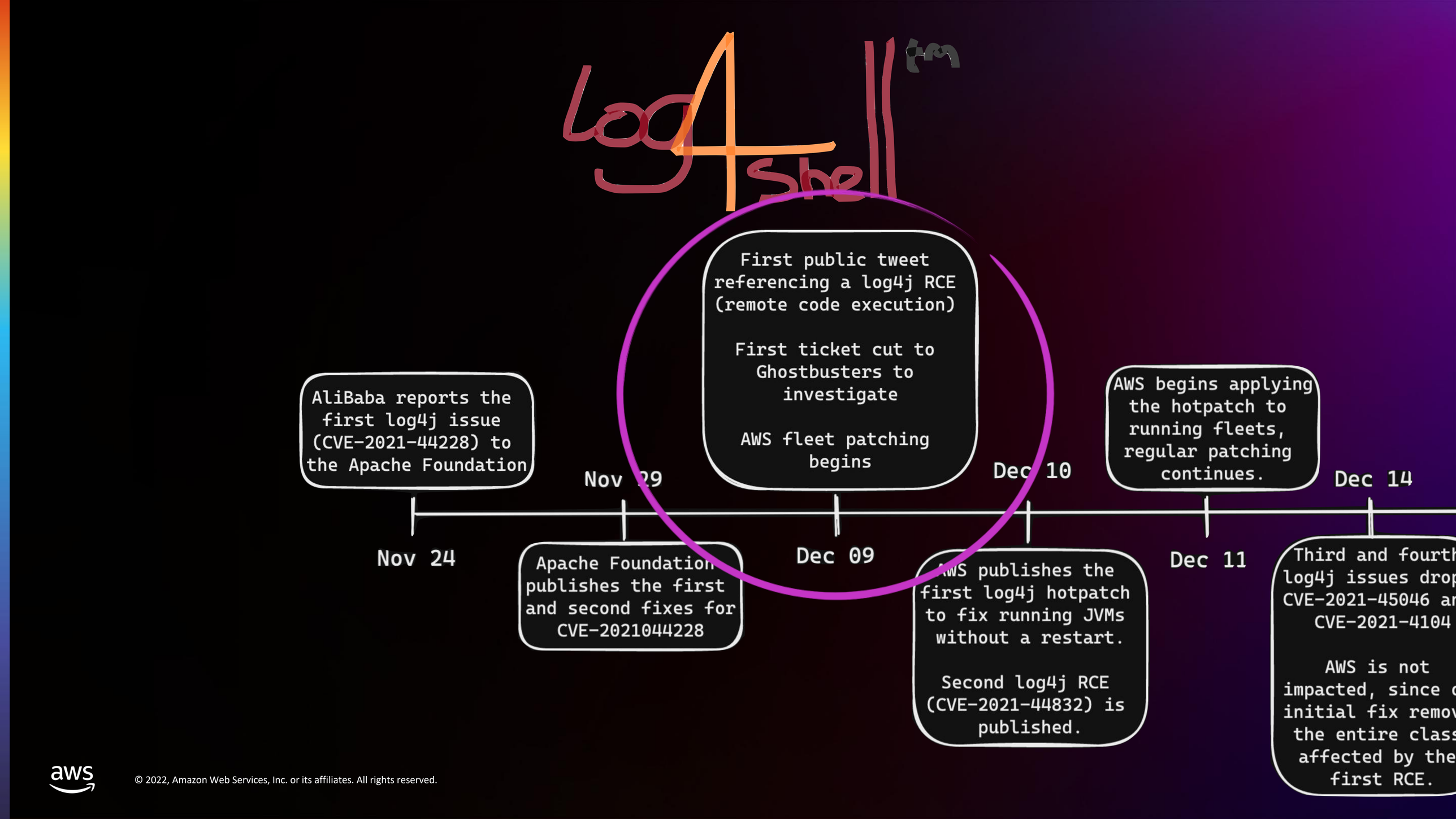


© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Log4Shell™



© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.



© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.



Log4Shell day 1: here we go!

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.





The first Twitter mention of a potential log4j issue.

This catches people's attention. A few things make this one stand out: it references an RCE (“remote code execution”), and it doesn’t specify an individual version of log4j.

Bonus points: log4j is really, really widely used.


heige
@80vul

Please set log4j2.formatMsgNoLookups to true, or you should upgrade to log4j-2.15.0-rc1 github.com/apache/logging... to defend the latest apache log4j2 RCE vulnerability 🤣

apache/**logging-log4j2**

Apache Log4j 2 is an upgrade to Log4j that provides significant improvements over its predecessor, Log4j 1.x, and provides many...

142 Contributors · 0 Issues · 3k Stars · 1k Forks

github.com
Release log4j-2.15.0-rc1 · apache/logging-log4j2
[maven-release-plugin] copy for tag log4j-2.15.0-rc1

7:40 AM · Dec 9, 2021 · Twitter Web App

The first tweet (that I can find)

aws
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.





Marc cuts the first AWS Ghostbusters ticket.

Tickets > Log4j 2 RCE?

2 Log4j 2 RCE?

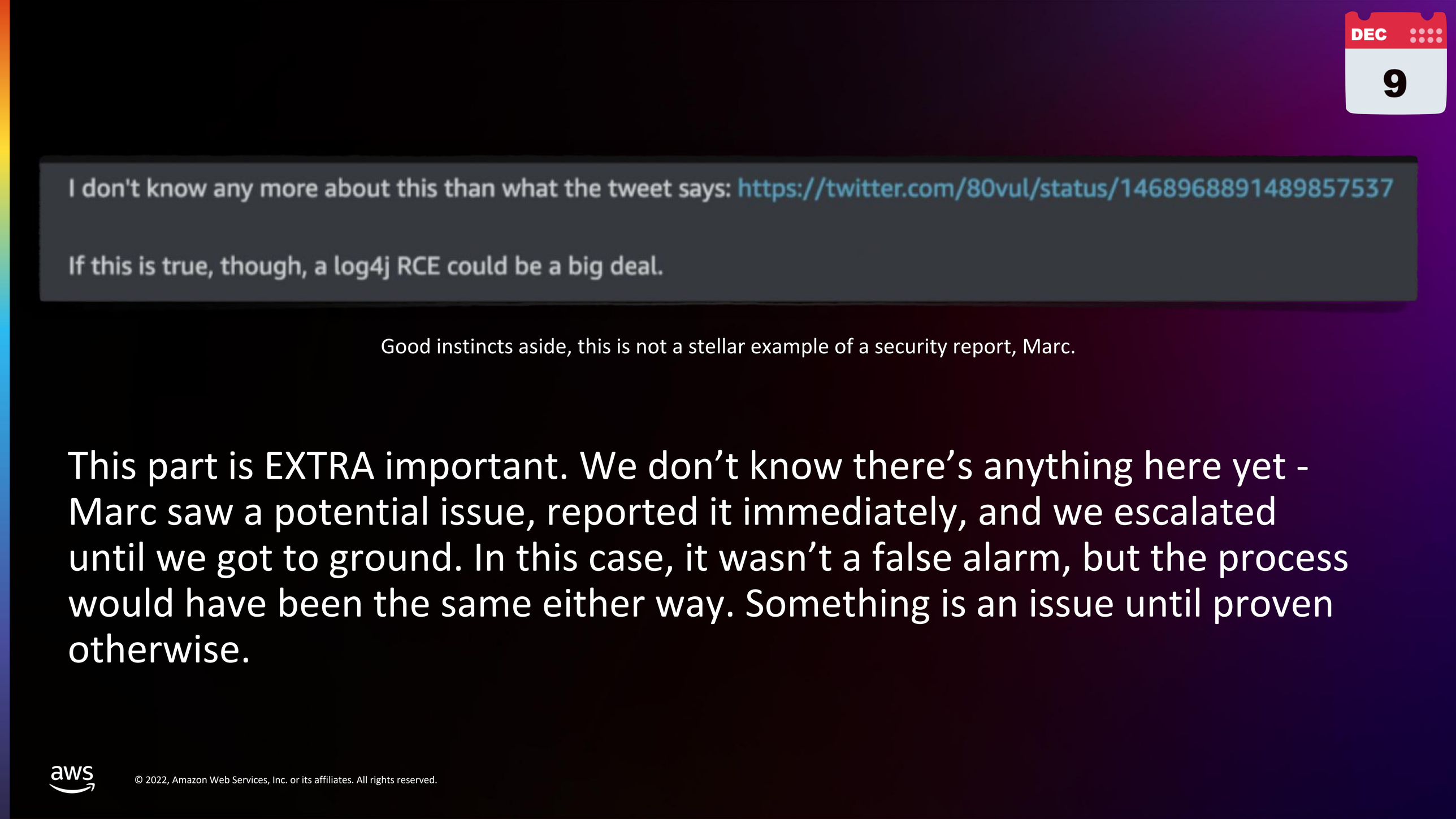
The original Ghostbusters ticket. There are older tickets from tooling and scanners, but this is the first 5-alarm fire.

I don't know any more about this than what the tweet says: https://twitter.com/80vul/status/1469688891489857537

If this is true, though, a log4j RCE could be a big deal.

This is foreshadowing

aws
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.



Good instincts aside, this is not a stellar example of a security report, Marc.

This part is EXTRA important. We don’t know there’s anything here yet - Marc saw a potential issue, reported it immediately, and we escalated until we got to ground. In this case, it wasn’t a false alarm, but the process would have been the same either way. Something is an issue until proven otherwise.

aws
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.



Colm notifies Steve



First briefing for the rest of AWS senior leadership

Briefings continue twice per day through the first week of Log4Shell.





© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.



The first mitigation rolls out

Even before we understand the full picture, pro-active mitigations went out. The first pass was a set of traffic-filtering rules.

```
jndi:
%6andi:
j%6edi:
%6a%6edi:
jn%64i:
%6an%64i:
j%6e%64i:
%6a%6e%64i:
jnd%69:
%6and%69:
j%6ed%69:
%6a%6ed%69:
jn%64%69:
%6an%64%69:
j%6e%64%69:
%6a%6e%64%69:
jndi%3a
%6andi%3a
j%6edi%3a
%6a%6edi%3a
jn%64i%3a
%6an%64i%3a
j%6e%64i%3a
%6a%6e%64i%3a
jnd%69%3a
%6and%69%3a
j%6ed%69%3a
%6a%6ed%69%3a
jn%64%69%3a
%6an%64%69%3a
j%6e%64%69%3a
%6a%6e%64%69%3a
```

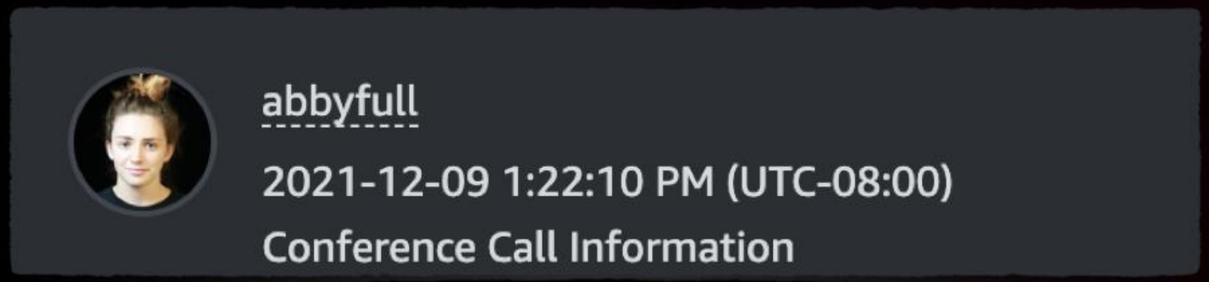


aws

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.



Ghostbusters start the first TOS call



That’s me.



In the end, the first call runs continuously for nearly 3 DAYS.

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.



Ryan proposes the first internal patch.

```
[0a903ec by                ] Remove JndiLookup
```

Thanks Ryan!

This initial patch disables the ENTIRE JndiLookup class.

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.





TOS call decides to patch both v1 and v2

The initial upstream CVE-44228 was only issued for log4j v2. However, it didn’t take us long on the TOS call to decide that we needed to patch both v1 AND v2.

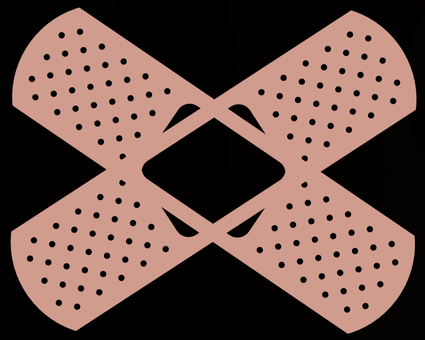
aws

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Event response typically follows a rough pattern at AWS:



MITIGATION



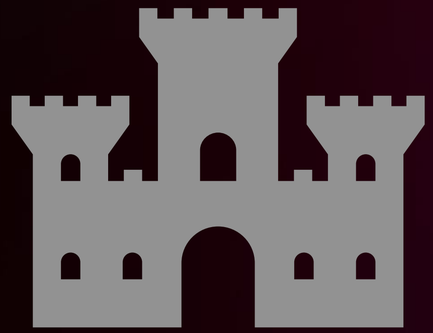
Stop the bleeding. Prevent new occurrences of an issue, or prevent an issue from getting worse.

DETECTION



Can we detect if this issue happens again? Are there warning signs?

PREVENTION



Do we know root cause? Close whatever gap led to the issue happening in the first place.

REPAIR

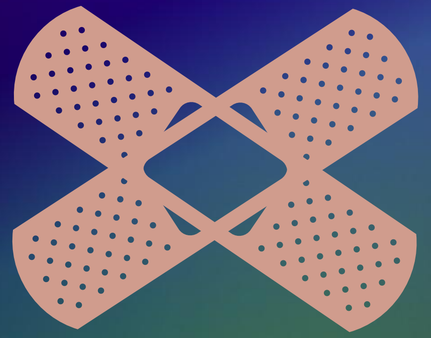


Identify and fix any existing impact.

aws

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

At this point, we have a short-term mitigation in place: the WAF filters



© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

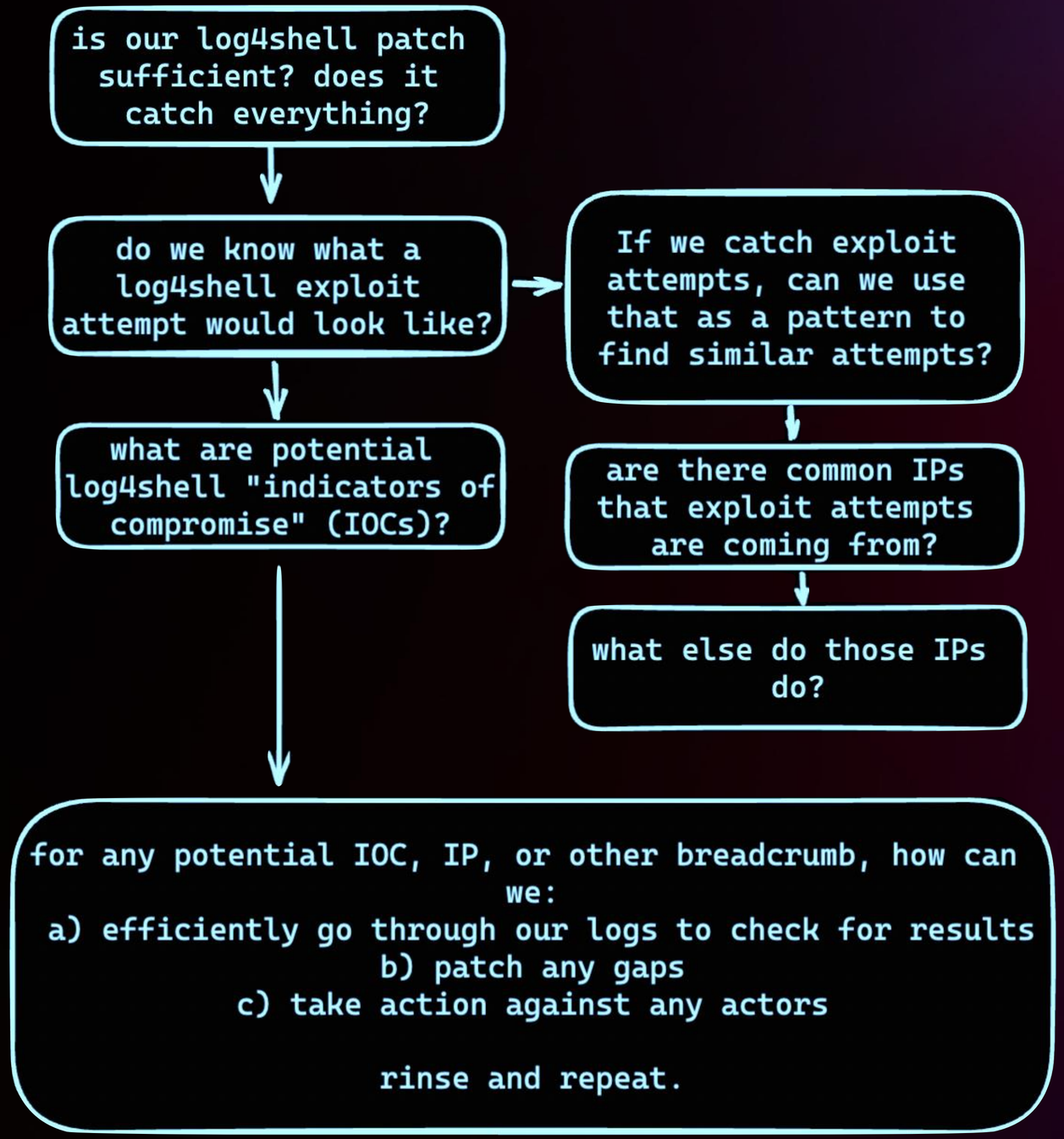


If there are attempts to leverage Log4Shell, can we catch them?

We think yes! A handful of AWS Security teams spin off to look at this.



© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

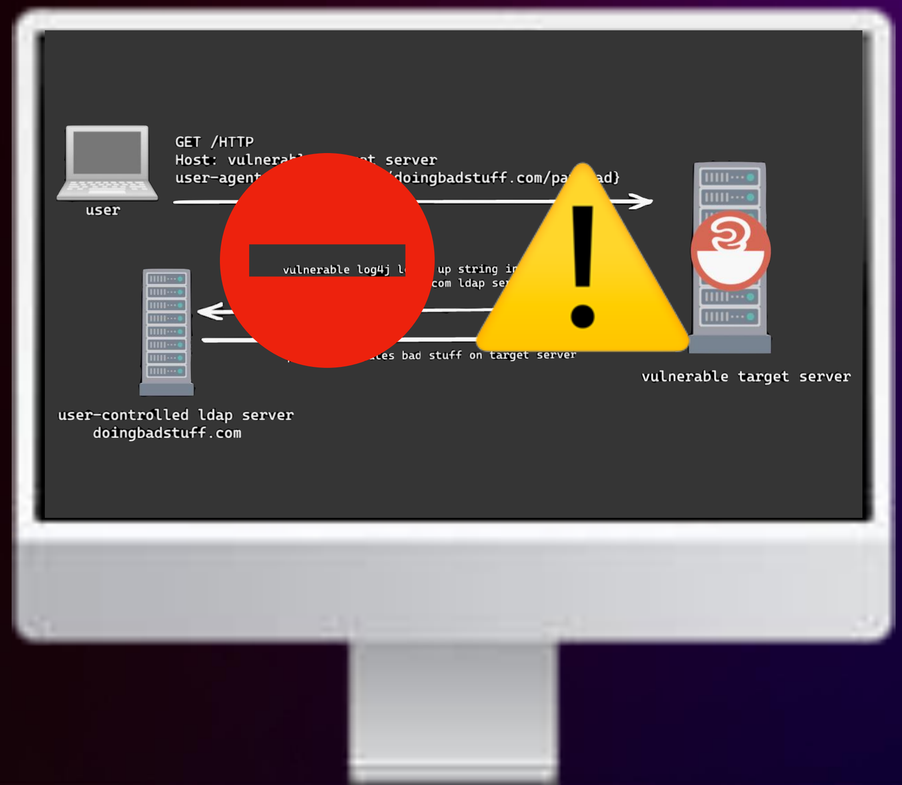


aws
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

The investigation and analysis work is one of the first branches off our main event call.

The work that comes out of this ends up being one of the coolest parts to come out of this.

By the end of the initial event response, this tooling had identified and blocked over 2 billion attempts to leverage Log4Shell.



© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

We have mitigation and detection. With the patch rolling out, we move on to prevention.



aws

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.



Mass deployment prep begins for AWS teams



[CRITICAL] Log4j / JNDI Security Issue Upgrade Campaign

First of many.

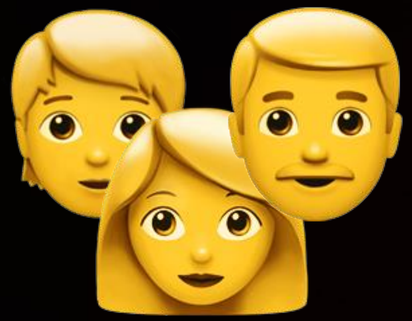
aws
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.



What happens when you have to prioritize work across so many services?

From the first day of Log4Shell, it was clear that we needed to be extra thorough. Ultimately, everything needed to be running a properly patched version of log4j, even if they weren't directly impacted.

Everything is hard, though, and there are a few constraints:



People



Availability

aws

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

What happens when you have to prioritize work across so many services?



WAVE ONE

Network-connected services that directly take customer input.
Large, low-level fleets.
Stateful services with strict durability requirements.
Common dependencies for other services.

WAVE TWO

Network-connected services that don't directly take customer input.
Network-connected internal-only services
Container images
Lambda functions.

WAVE THREE

Non-network-connected services
Developer desktops and test environments
Less common dependencies for other services.

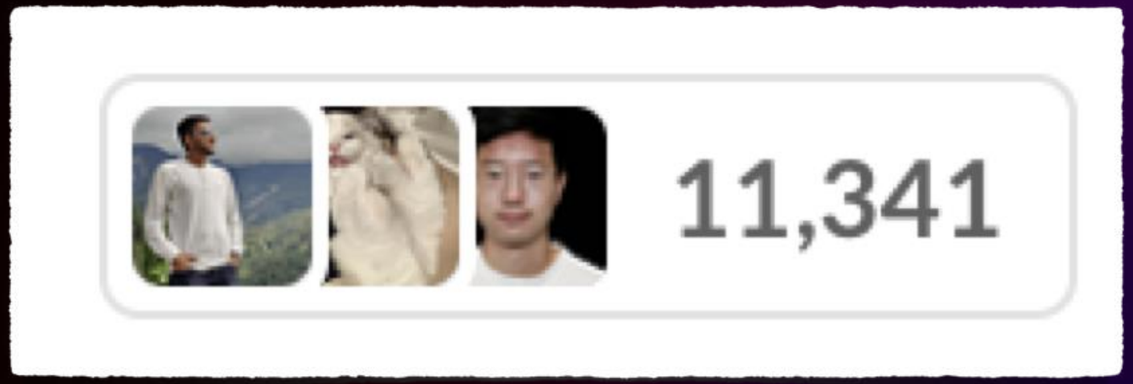
aws
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Coordinating across this many teams is a massive effort



Starting a campaign involving this many resources means tens of thousands of tickets. That means thousands of on-calls (and thousands of questions).

Good campaigns have good documentation. That means clear, easy-to-follow directions for resolving an issue, answers to common questions, and somewhere to send people to get help.



I wasn’t kidding, these were BUSY. This isn’t the peak, just the peak of “Abby remembered to document”.

aws
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.



AWS change control takes effect



To handle the volume of builds and upcoming service deployments, “change control” takes effect to prioritize any log4j response. This means that other builds and deployments are paused, and leadership approval is required to override the setting.



aws

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.



The response team starts to contact large teams directly.



One of the benefits of having (very) distributed teams is that we're able to work around the clock in emergencies. At around 20:00 in Seattle, we start contacting large fleets to work through deployment plans.

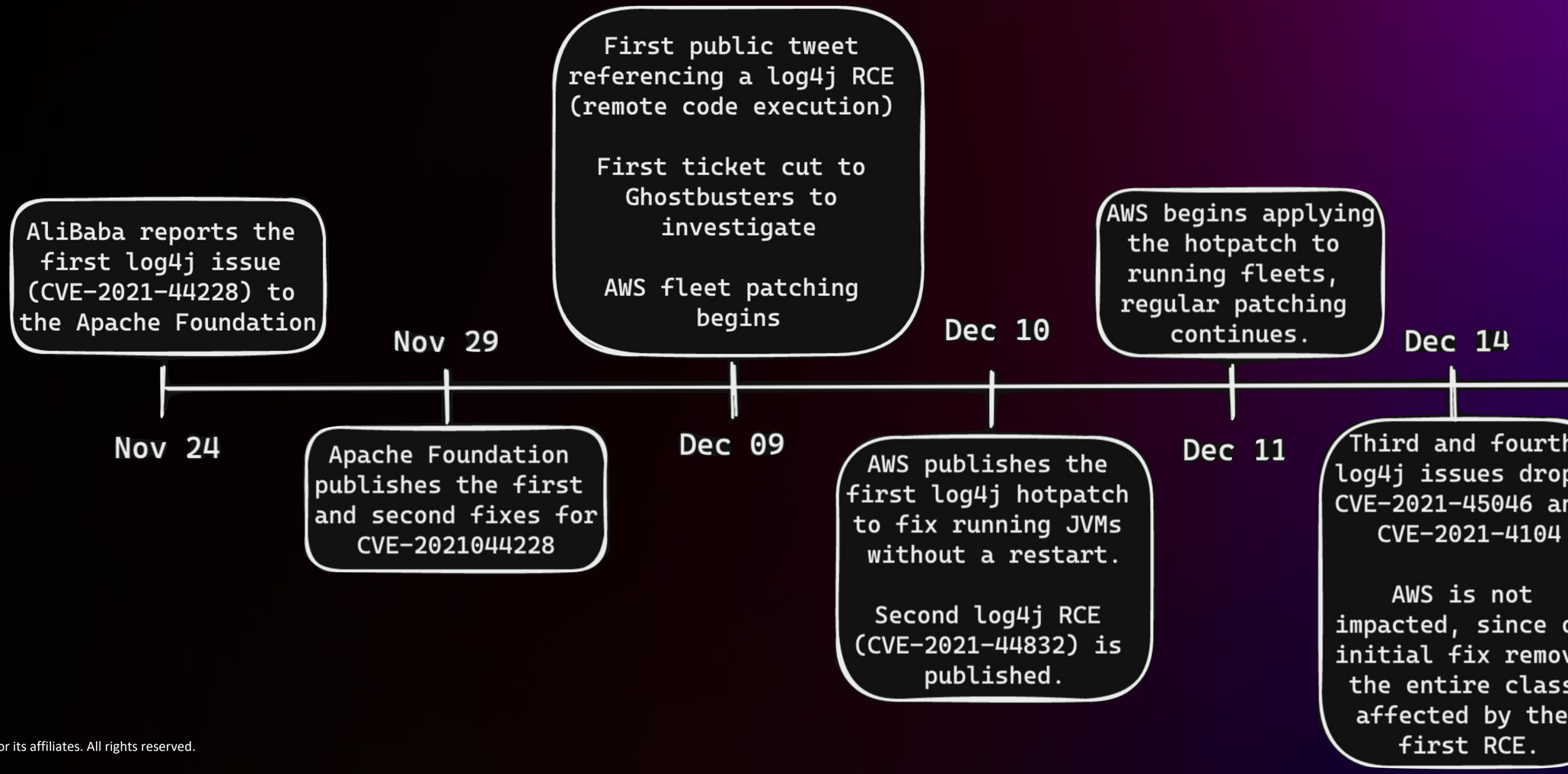
Amazon EC2 is split between Seattle and Cape Town. During Log4Shell, we take advantage of this and patch during the SAST daytime (+9 from SEA). Ultimately, the vast majority of Amazon EC2 patches that first night.



aws

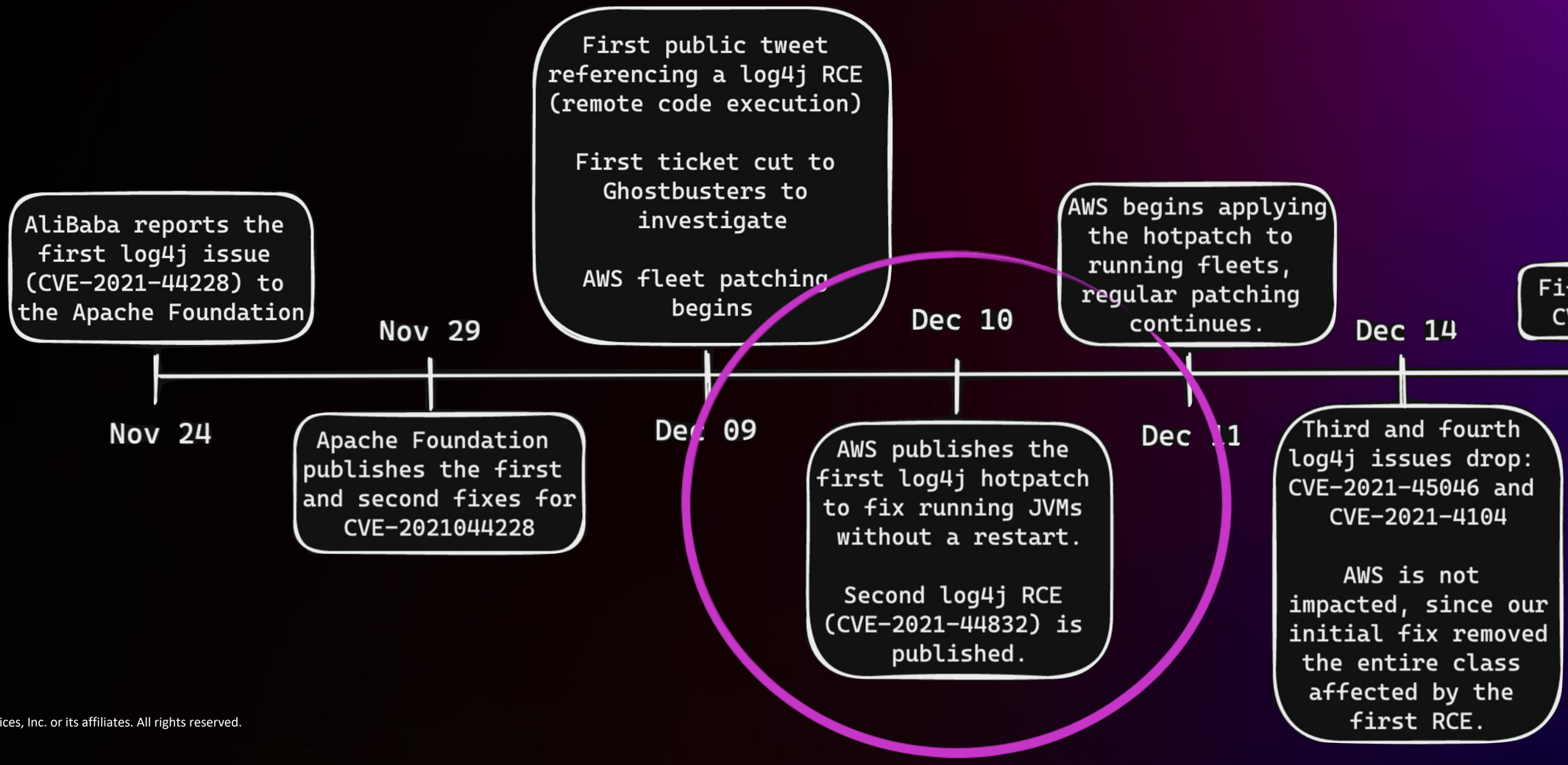
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Log4Shell™



© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Log4Shell™



© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.



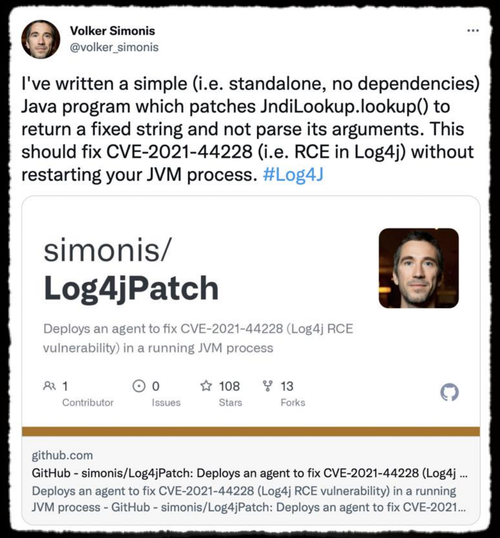
Log4Shell day 2: “just a quick hack”

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.





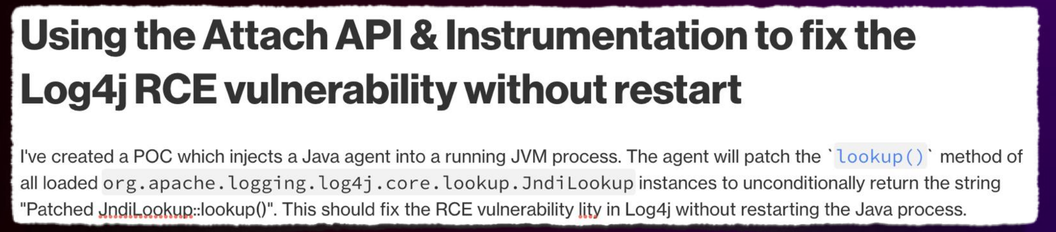
Volker tweets his log4j hotpatch.
This lets you patch a running JVM without a restart (patch in place)



The original tweet



The hotpatch at its new, permanent home:
github.com/corretto/hotpatch-for-apache-log4j2

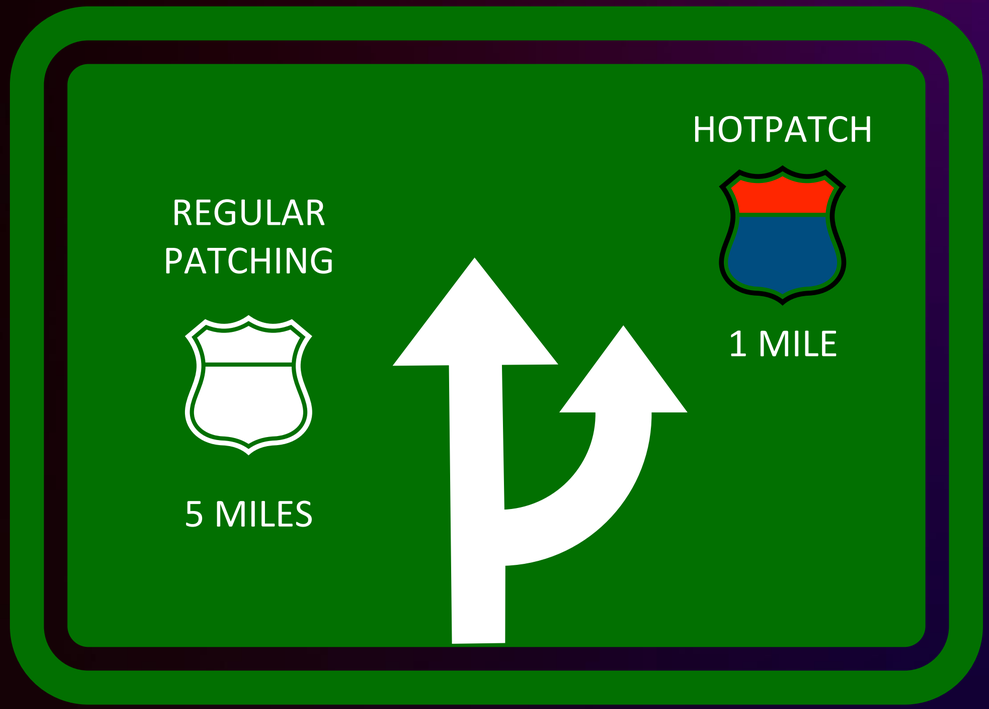


The first internal proposal

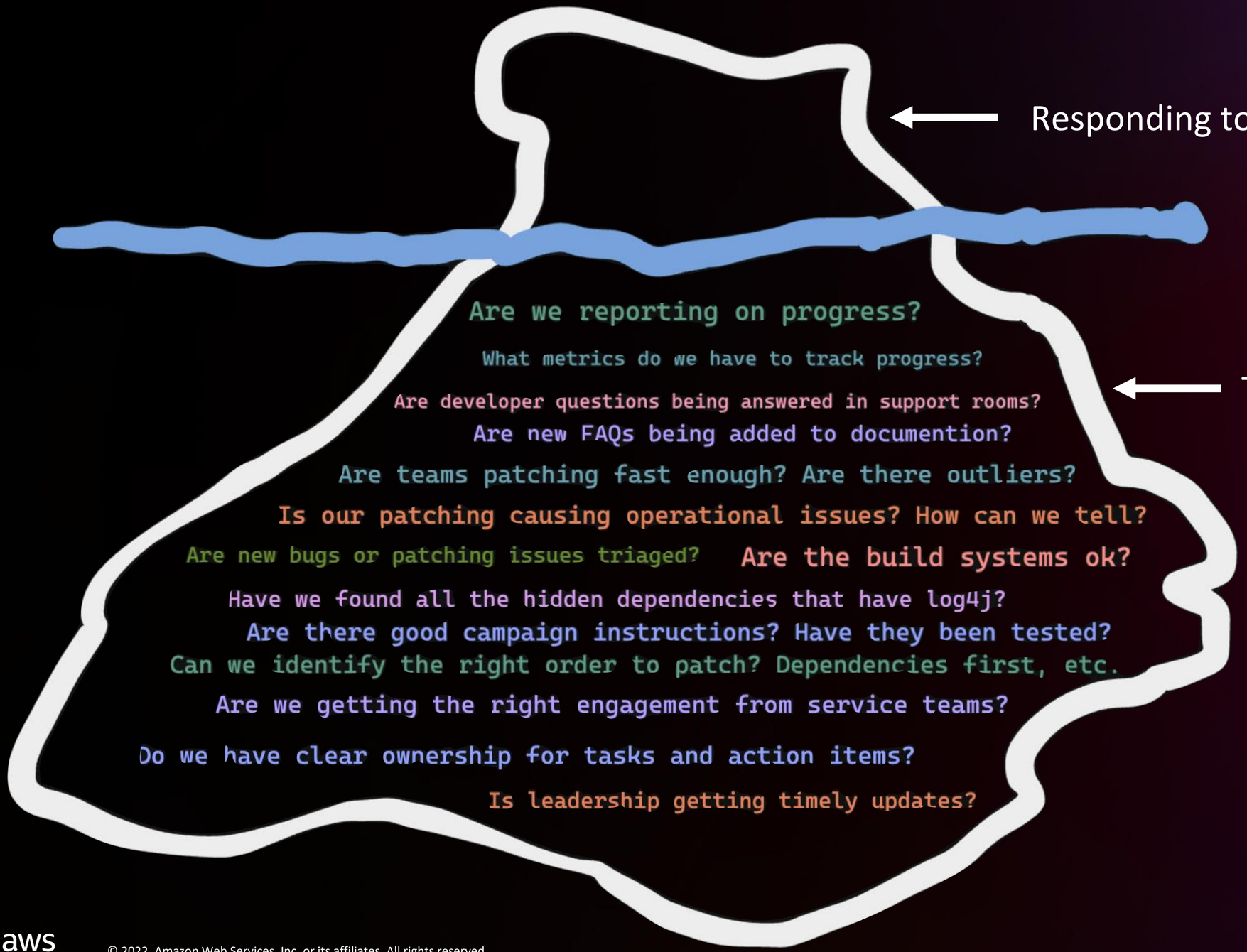
aws
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.



The Seattle crew is back on-shift. We wake up to both Volker tweeting his patch and to the news that S3 is rolling out the live patch alongside the main one.



© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.



© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.





Builder Tools starts second of deployments



All non-Log4Shell builds stopped



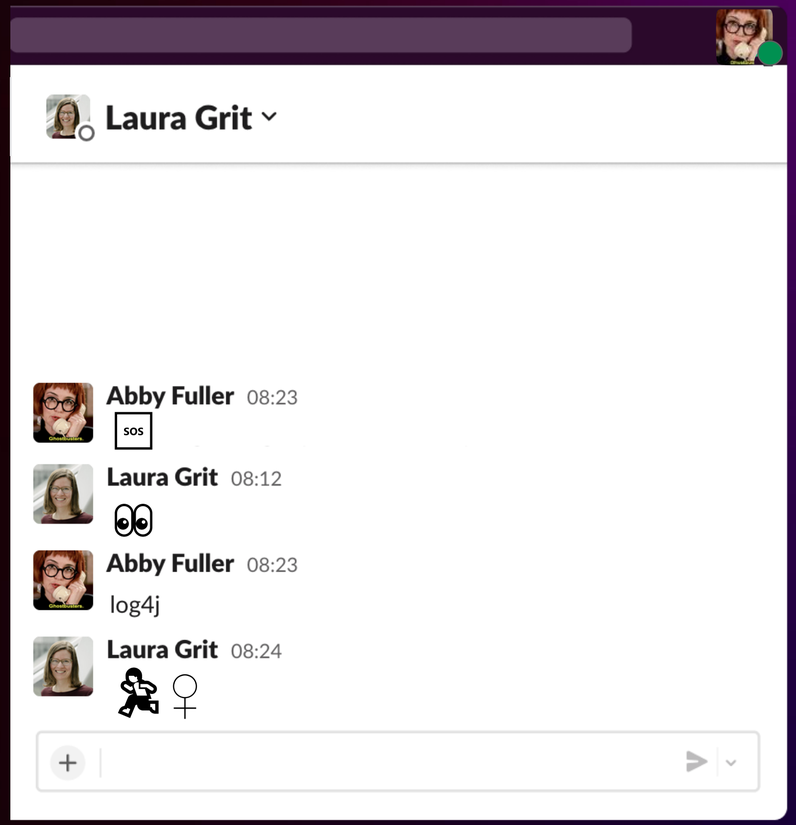
aws
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.





“We have a patch, now what?”

AWS is a big place, and making things happen requires special skills! Enter Laura.



A less-dramatic reproduction of how I actually asked for help during log4jshell

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.





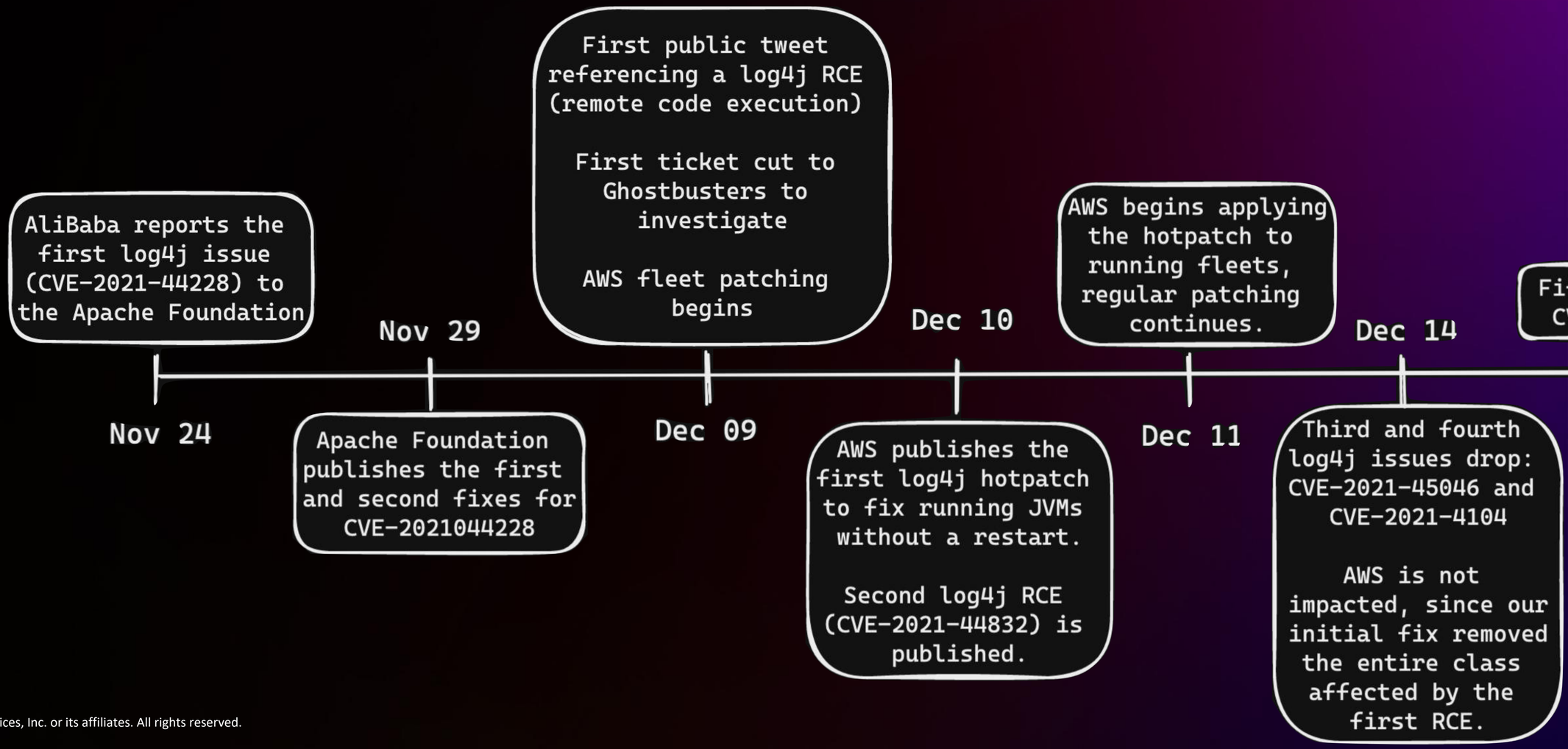
At 19:00 PST or so, the second log4j CVE drops.

CVE-2021-44832: *Some non-standard configurations of log4j are vulnerable to RCE if they use JDBC Appender with a JNDI LDAP source.*

AWS removed the JndiLookup class entirely, so this was covered in our existing patch.

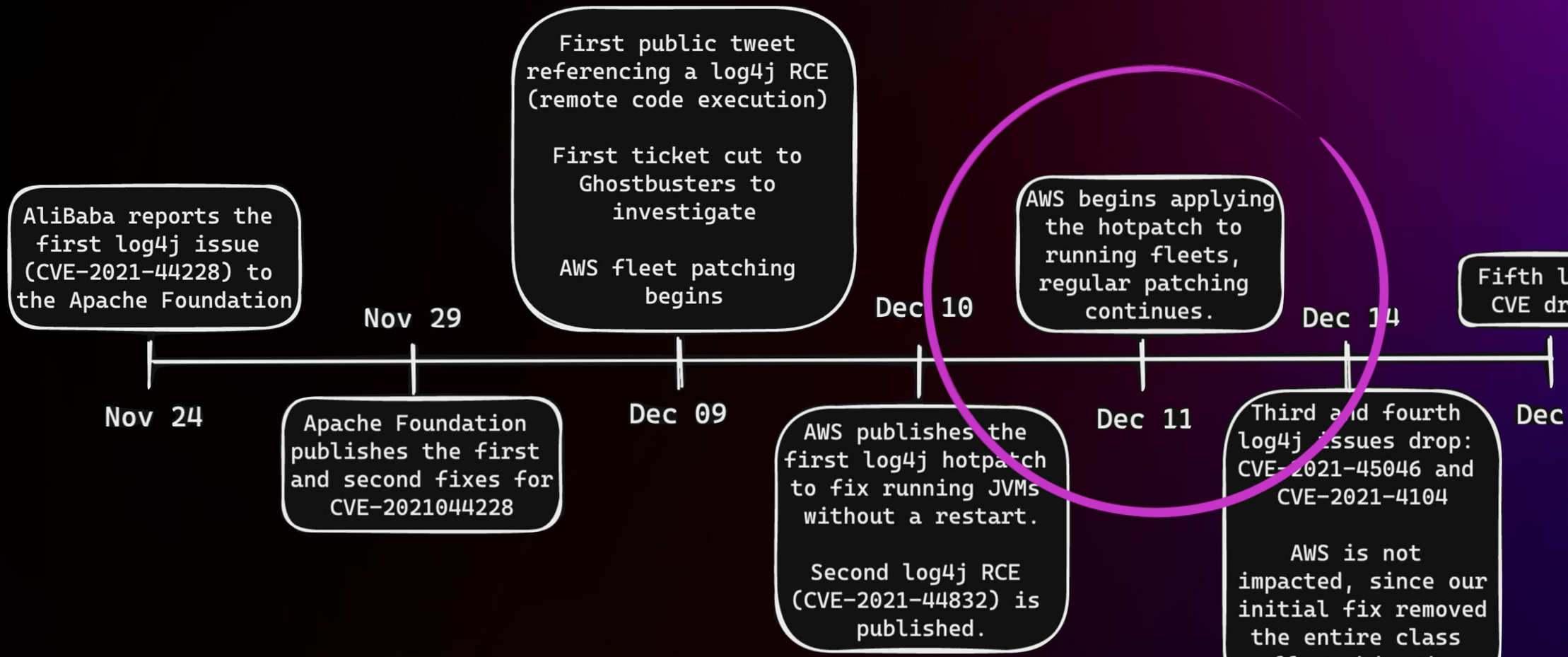
aws
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Log4Shell™



© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Log4Shell™



© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.



Log4Shell day 3: are we still doing this?

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

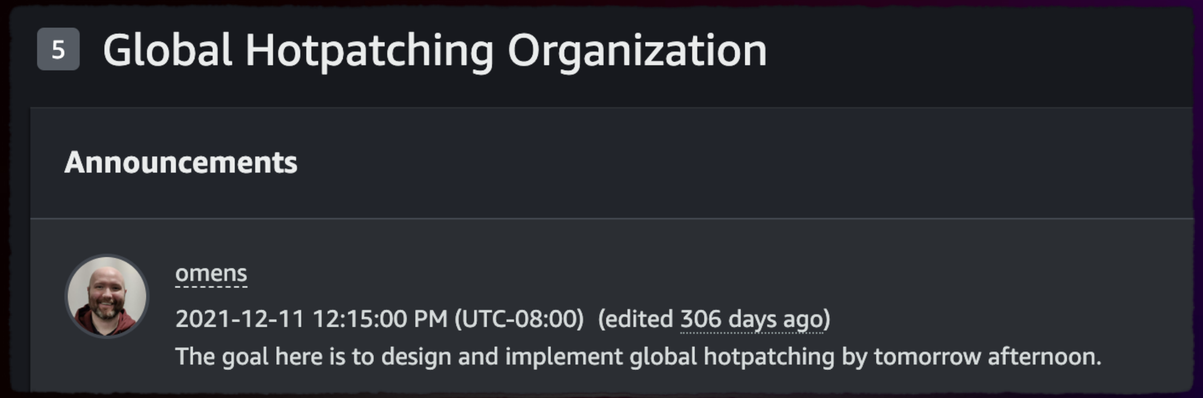




Global hotpatching is a go. After testing, working on packaging, and figuring out a safe strategy for this many parallel deployments, we're off!



Hotpatching starts a few hours earlier, but it starts picking up speed around 15:00, after we've safely made it through multiple Regions.


5 Global Hotpatching Organization

Announcements

omens
2021-12-11 12:15:00 PM (UTC-08:00) (edited 306 days ago)
The goal here is to design and implement global hotpatching by tomorrow afternoon.

The main hotpatching coordination ticket.

aws
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.





With Laura on the job, we turn the attention of the non-hotpatching thread to tracking down stray dependencies, Amazon Elastic Container Service (Amazon ECS) tasks, and developer desktops.



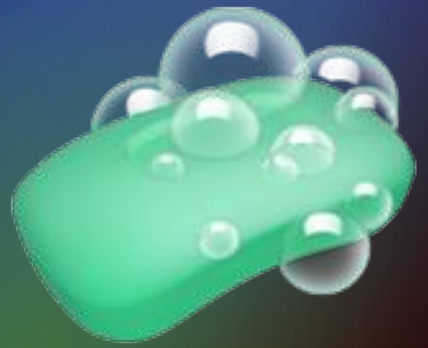
We cut a set of tickets to a third wave of teams, this time for ECS tasks.



aws
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.



From December 11th to December 14th, patching clean-up continues. We track down stray tasks and functions and go through developer desktops.



© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

“I’ve patched, now what?”



© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.



Follow up investigations for service teams start.

A separate thread splits off the main group to help service teams with any follow up investigation.

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.



Log4Shell day 4: Lambda gets creative

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Lambda doesn’t include log4j in its managed runtimes or base container images



However, Lambda customers do. On December 12th, we make a change to the Lambda managed runtimes and images that would mitigate the Log4Shell issue for vulnerable customers. For customers on managed runtimes, this change applied automatically.

AWS Lambda

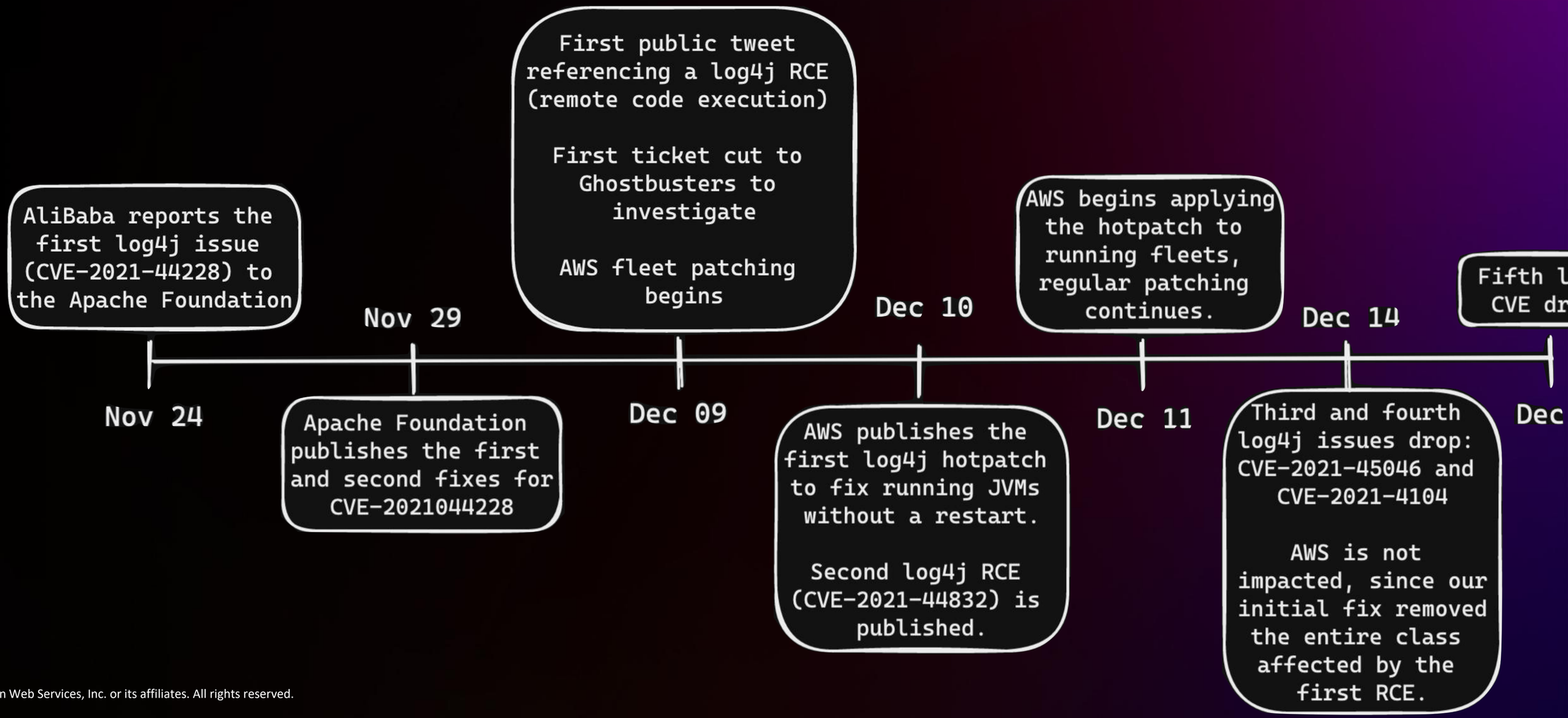
AWS Lambda does not include Log4j2 in its managed runtimes or base container images. These are therefore not affected by the issue described in CVE-2021-44228 and CVE-2021-45046.

For cases where a customer function includes an impacted Log4j2 version, we have applied a change to the Lambda Java managed runtimes and base container images (Java 8, Java 8 on AL2, and Java 11) that helps to mitigate the issues in CVE-2021-44228 and CVE-2021-45046. Customers using managed runtimes will have the change applied automatically. Customers using container images will need to rebuild from the latest base container image, and redeploy.

The December 12th security bulletin for Lambda

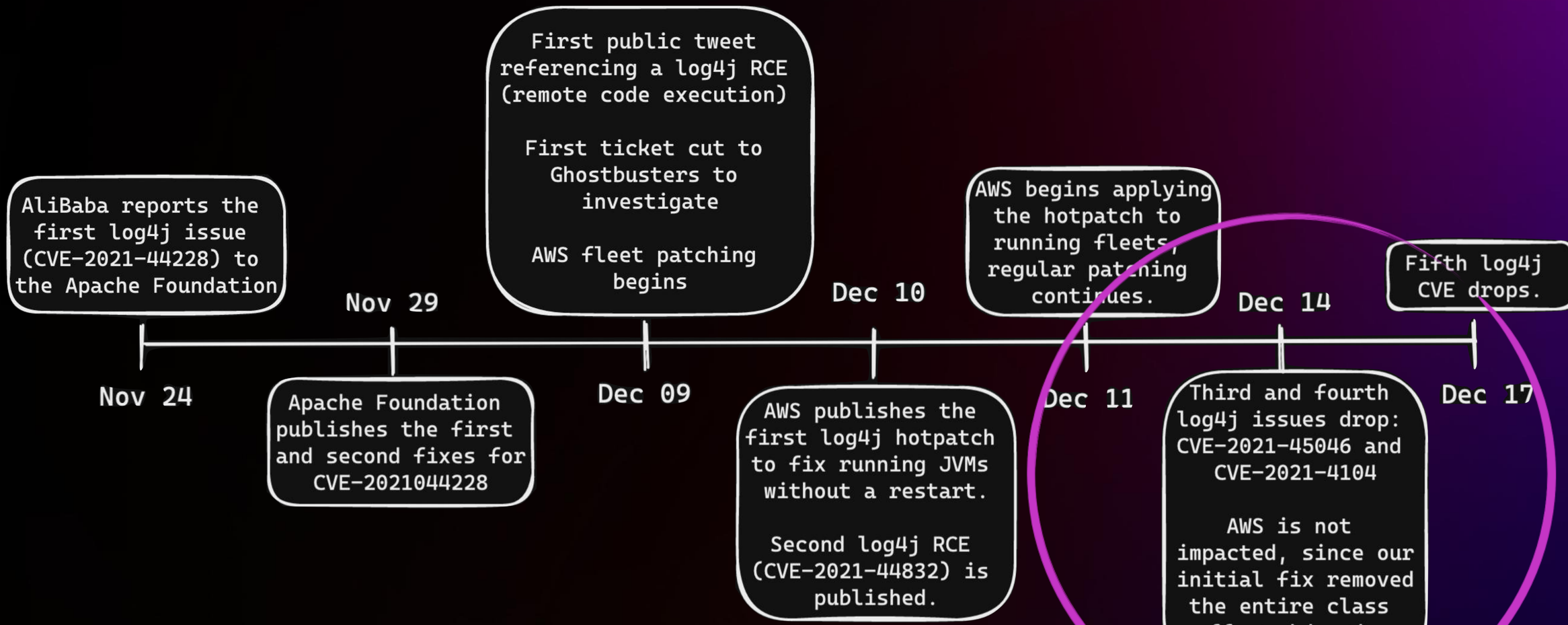
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Log4Shell™



© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Log4Shell™



© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.



Log4Shell day 6: the light at the end of the tunnel is a train.



aws
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

On December 14th, two additional log4j CVEs drop



CVE-2021-45046: *the original patch for CVE-44228 was incomplete. RCE was still possible In some non-standard configurations where an attacker controls a lookup context or thread context.*

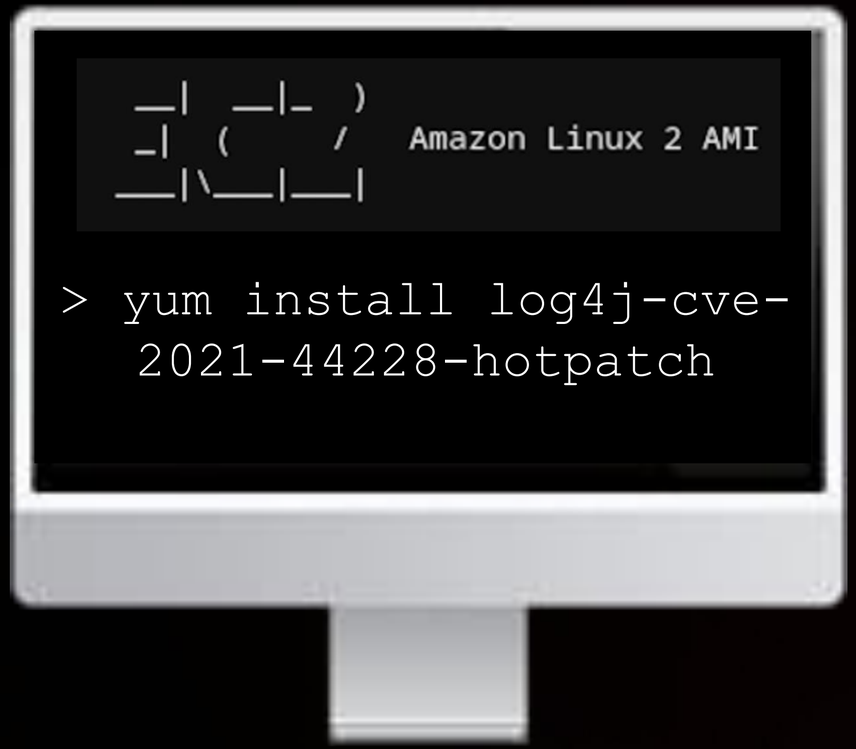
CVE-2021-4104: in some non-default configurations in log4j v1, where JMSAppender is enabled, and an attacker has write access to the log4j configuration file, Topic settings can be leveraged for JNDI lookups, and RCE is possible.

AWS is NOT impacted by either of these, since the entire JndiLookup class had been removed by our first patch, and because we included v1 from the beginning.

aws

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Hotpatch rpm published for customer use



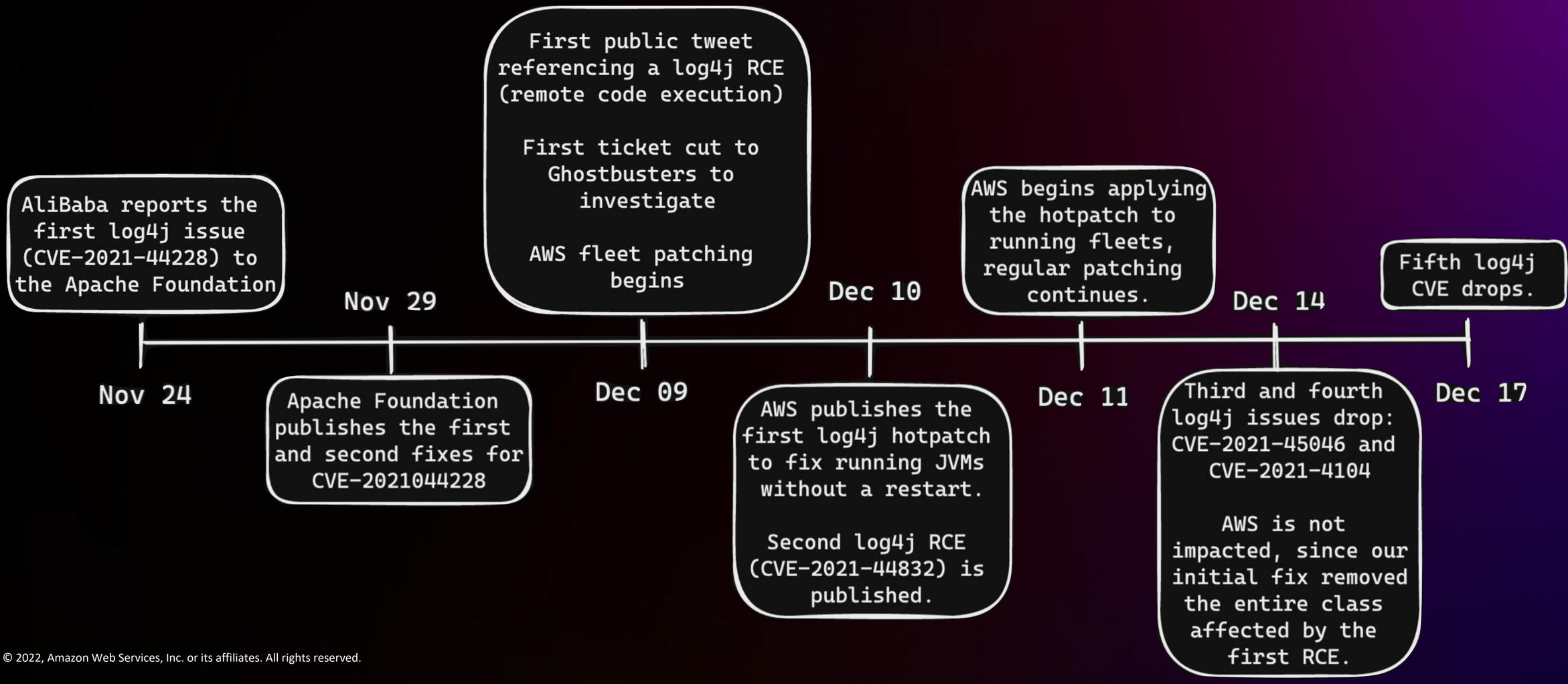
rpm hotpatch

Amazon Linux 1 and Amazon Linux 2 are not affected by Log4Shell by default, but some customers use their own versions.

On December 14th, we publish the rpm version of the hotpatch for easy consumption by Amazon Linux customers.

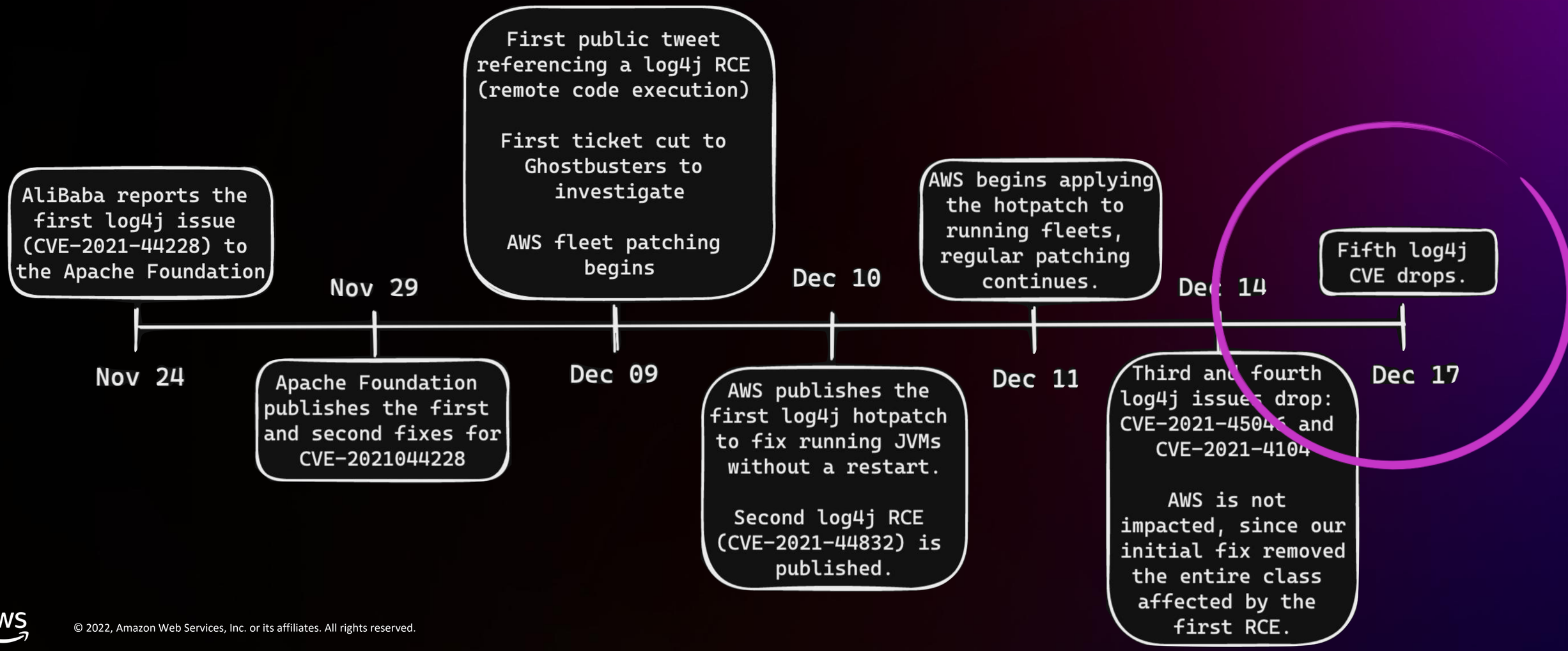
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Log4Shell™



© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Log4Shell™



© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.



Log4Shell day 9: the final encore.

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.



On December 17th, an additional log4j CVE goes public:

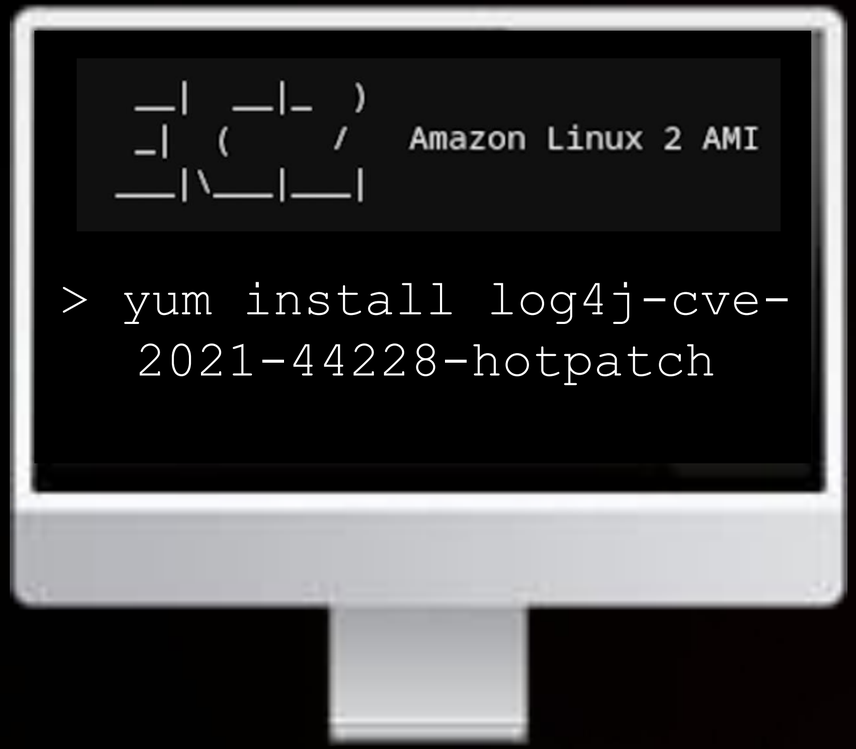
CVE-2021-45105: *some configurations of log4j are vulnerable to a denial-of-service (DoS) issue through uncontrolled recursion, if an attacker passes a specially-crafted string. This requires control over the thread context.*

AWS is NOT impacted.

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

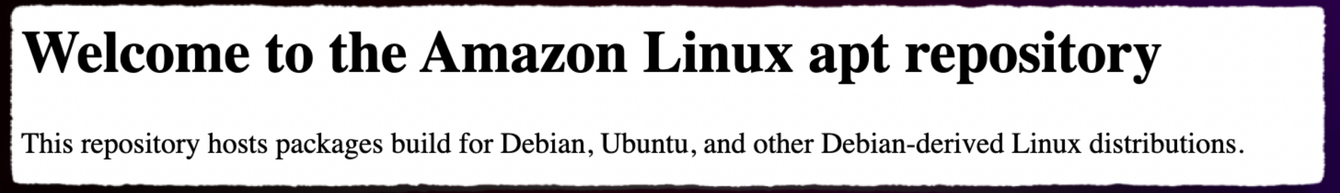
Amazon Linux 1 and Amazon Linux 2 start applying the hotpatch by default to installed JDKs

DEC 17



The rpm hotpatch

We also publish apt.repo.amazonlinux.com to make the hotpatch easily accessible for customers running Debian and Ubuntu.



The Amazon Linux apt repo

aws
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Phew. What went well? What didn’t?

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

There were less successful elements to our Log4Shell response

1. We had rocky starts for teams in terms of deployment instructions, especially as additional CVEs rolled out.

2. We were inconsistent with providing customer messaging updates, particularly during the first half of the event.

3. We had some hotpatch rollout snags, like the latency blips for Lambda.

4. Deploying things on behalf of teams en masse is unusual, and we caused some confusion with how we communicated what was happening with the hotpatch.

5. Initial tickets often went to service on-calls who weren’t familiar with Log4Shell. In a lot of cases, we should have gone straight to senior engineers and managers.



aws

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.



The good news: there were some really, really cool wins here.

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

A few highlights

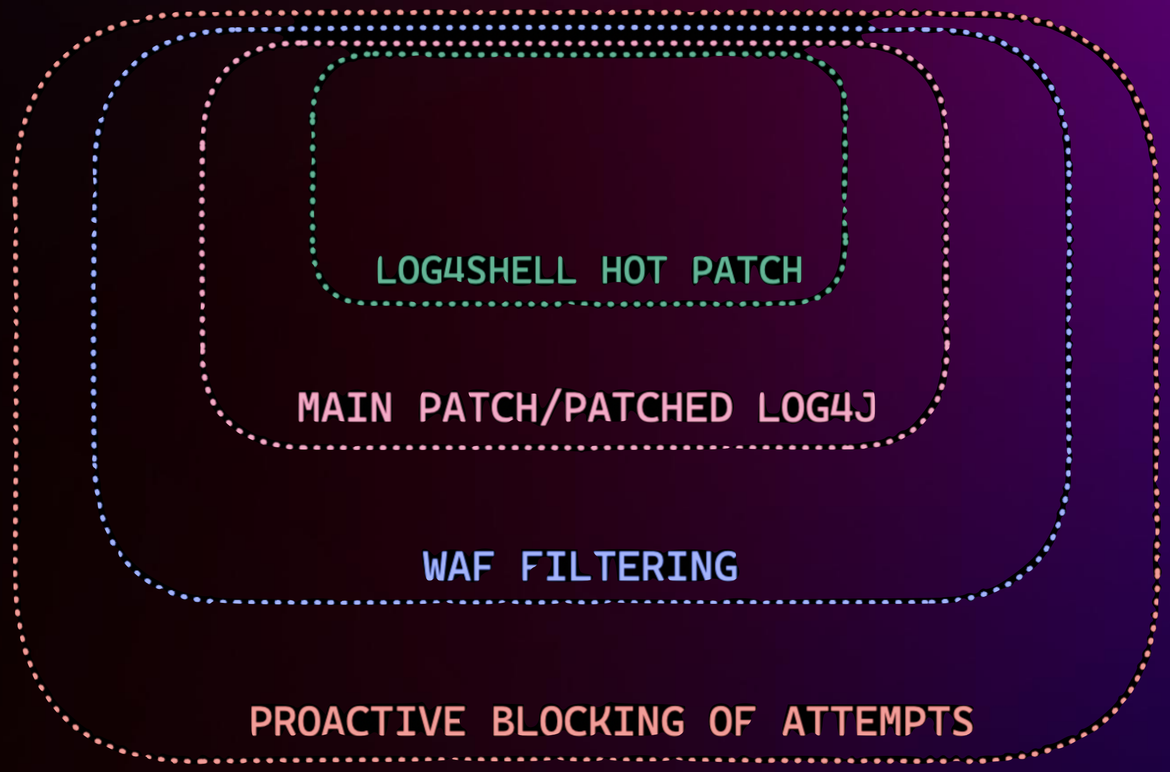
1. Ryan proposing removing the entire JndiLookup class, and assuming that log4j v1 was affected from the beginning.

2. Defense in depth: we continued to add layers throughout the response, and iterated on filters and blocks as we learned new info. The analysis from the AWS Security teams was key here.

3. The hotpatch from Volker

4. This was an utterly massive load on the build tooling. Proactive thinking from Builder Tools limited the headache here.

5. Hard work from thousands of teams during a particularly tough time of year. Nothing gets you in the holiday spirit like a 0-day!



A not-to-scale illustration of defense in depth.

aws
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

More generally, what did we learn about what works (and what doesn’t)?



© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

1. Engage leaders early.

2. (Over)communicate often.

3. Work backwards from what you can do FOR teams.

4. Have a plan for the long haul

5. You need who you need. It’s ok to go straight to senior engineers or leadership.

6. Iterate quickly.



© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

All told, we wrote around six separate retrospectives for Log4Shell, covering everything from the incident response itself to the build tooling performance.

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Lastly, this wouldn’t be a Log4Shell talk without acknowledging the incredible dedication and hard work from thousands of AWS teams.

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Thank you!

Abby Fuller
abbyfull@amazon.com
@abbyfuller



Please complete the session survey in the **mobile app**

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.