A guide to responding to the Log4j vulnerability

LACEWORK®

On Thursday, December 9, 2021,

the team of volunteers behind Log4j announced a serious vulnerability in the library. This whitepaper will walk through the background of Log4j and the discovered vulnerability, and four key steps you can take to better protect your organization.

What is Log4j?

Apache Log4j is an open-source Java library that is used to write data from an application to a log. It has been around for years and is the most popular Java logging library in use today.

As a part of modern software development, the use of open source software is standard practice. The benefit to using open source libraries like Log4j is that it saves time and money to use software and libraries that are written and easily used.

The downside is that when a vulnerability is discovered in open source, the effects can be pervasive across multiple companies and cloud providers and determining all of the applications and services that use the open source library is difficult and time consuming.

What is the Log4j vulnerability?

On December 9, 2021, a critical vulnerability affecting Log4j was disclosed. In fact, the National Institute of Standards and Technology (NIST) rated the severity of this issue as 10/10 in the National Vulnerability Database (NVD). At the time this whitepaper was written, there were already four common vulnerabilities and exposures (CVE) identifiers assigned to this issue: CVE-2021-44228, CVE-2021-45046, CVE-2021-45105, CVE-2021-4104. This vulnerability is not only a major concern because of how it can be exploited by attackers, but also because of how pervasive the use of Log4j is across all industries, verticals, and company sizes.

Logging is one of the fundamental tenets of good programming practice and a fundamental of proper security. Log4j has many features that can be used to ensure debugging and audit information is logged appropriately. One of these features takes advantage of a Java library called Java Naming and Directory Interface (JNDI). This API allows developers to interact with directory services, like Active Directory, to query user or directory information. Log4j uses JNDI to query directory information that can be included as a part of the logs (for example, the username of a person performing an action). The first Log4j CVE allows a manipulation of the JNDI string that would redirect the request to a rogue directory server. That malicious directory server would return a malicious payload, like a library embedded with malware, which would then be executed. The results of the exploit could allow attackers to take control of your system or implant malware on your server.

As stated above, more CVEs have been assigned as the impact and scope of the vulnerability becomes better understood.

When you dig into each of the new vulnerabilities, they take advantage of the same concepts used in previous CVEs. They exploit the JDNI and manipulate the payload using techniques like string manipulation to perform malicious actions like remote code execution or denial of service. Over time, it is likely that additional attack patterns will be discovered using similar methods of exploitation. It's important to continuously monitor your environment, as described below, to protect your workloads against new methods of attack.

It's important to continuously monitor your environment to protect your workloads against new methods of attack.

Steps to protect your organization

There are four key steps you can take to better protect your organization against the vulnerability and potential exploits. We'll walk through each of these in more detail:

1. Monitor for successful exploit attempts
2. Look for all installations of Log4j in your environment
3. Prioritize and remediate the affected systems
4. Continuously monitor your environment for exploit attempts from this event — and others that will follow

STEP ONE

Monitor your environment for successful exploit attempts

Any time a new critical vulnerability is disclosed, security teams immediately start trying to find vulnerable systems and remediate. Think about it like closing and locking all of the windows and doors to secure the office building. But what if an intruder has already gotten inside? How can you find out as quickly as possible before they cause damage or escape with something important?

As security teams start remediating the vulnerability, it's also critical to monitor your cloud environment for signs of compromise. The most effective way to do that is to look for anomalies in your runtime activity data. Attackers will use multiple methods to attempt to exploit the vulnerability and gain a foothold in your environment. We've already seen many examples of attackers taking advantage of this vulnerability. Many attackers are still using this opportunity to map out environments to see what infrastructure is susceptible to this vulnerability by entering JNDI attack strings across all available input variables.

Lacework® Labs has seen a lot of reconnaissance and gathering of vulnerable hosts through tracking callbacks. We have observed this data in our honeypots and continue to track IPs associated with this activity and report back to the InfoSec community to share data. In addition, Lacework Labs observed several botnets and cryptominers take advantage of this opportunity and install malware on vulnerable infrastructure. To date, from what Lacework Labs sees from our honeypots, these botnets are not detected well through heuristics due to changing data in the payload affecting signature checks. For more detailed information, see this Lacework blog post.

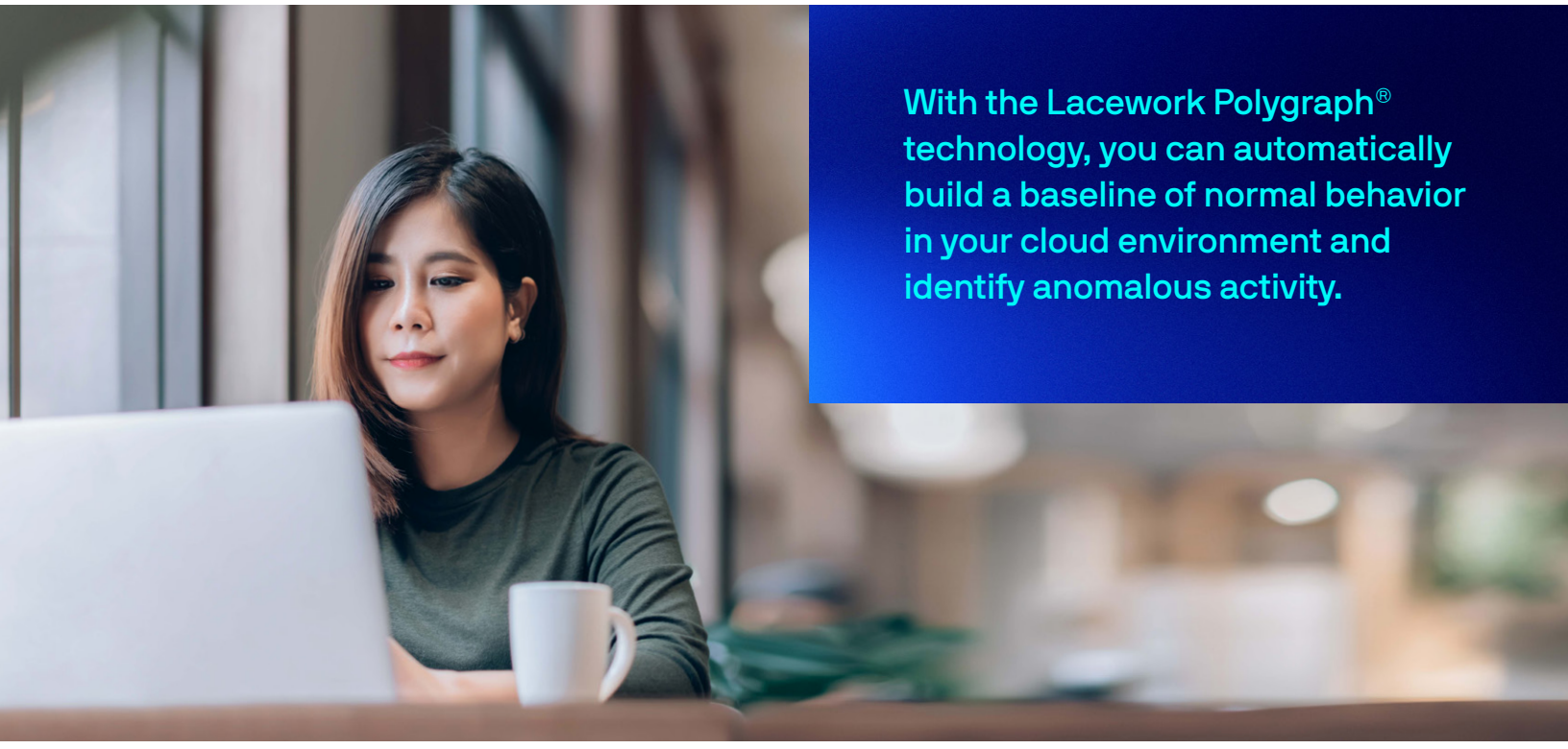
In addition, Lacework Labs noticed attackers trying different techniques to test the vulnerability and infrastructure affected. The pervasiveness of Log4j means that sometimes the payload may pass through a layer or two before it's eventually executed. This has led attackers to try to place the payload in a variety of places like the user-agent and refer header. Attackers also are trying a variety of techniques to exploit the payload like string manipulation and DNS lookups. More information can be found in this Lacework Labs post.

How to monitor for successful exploit attempts

Due to the changing nature of this exploit and the many ways it can be exploited, it is important to monitor not only as you're in the process of patching and identifying the vulnerabilities, but also as an ongoing best practice for your cloud environment. By looking for anomalies (a proactive approach) rather than trying to write rules (a reactive approach) to find what's bad, you'll be able to spot new techniques as they're deployed by cyber criminals. To do this, you need to understand the baseline of how your environment works and how it typically operates. If you have an ongoing understanding of what normal activity looks like, you can more easily spot suspicious activity that could be indicative of an intrusion or active attack. For example, you may want to look for new external requests from Java applications — especially if the applications have never made external requests before or if the IP addresses they're connecting to are new. By doing this, you can actually uncover attacker activity that's taking advantage of a zero day vulnerability — before it's been found or disclosed.

How Lacework helps

With the Lacework Polygraph® technology, you can automatically build a baseline of normal behavior in your cloud environment and identify anomalous activity. Instead of relying only on rules, Polygraph combines a patented form of autonomous machine learning, behavioral analytics, and anomaly detection and learns how your unique environment operates. We use machine learning to detect new things that have never happened before in your unique environment. Polygraph understands what your infrastructure looks like, and retains that understanding even when changes are made. With the cloud, new things happen all the time — new processes launch, patches are applied, new code deploys, etc. — but our machine learning automatically learns those patterns, recognizes when it's performing functions that are expected, and uncovers meaningful new behavior that you should pay attention to most. Lacework alerts based on the criticality of those changes and shows visual representations of all of the connections to make investigations easier for security and DevOps teams.



Let's walk through an example using the Lacework platform

One way a Log4j exploit can be identified is by looking for the combination of two things: 1) the presence of the Log4j CVE and 2) the vulnerable host also connecting to a new external host. Lacework detects this combination of events using the Polygraph® technology.

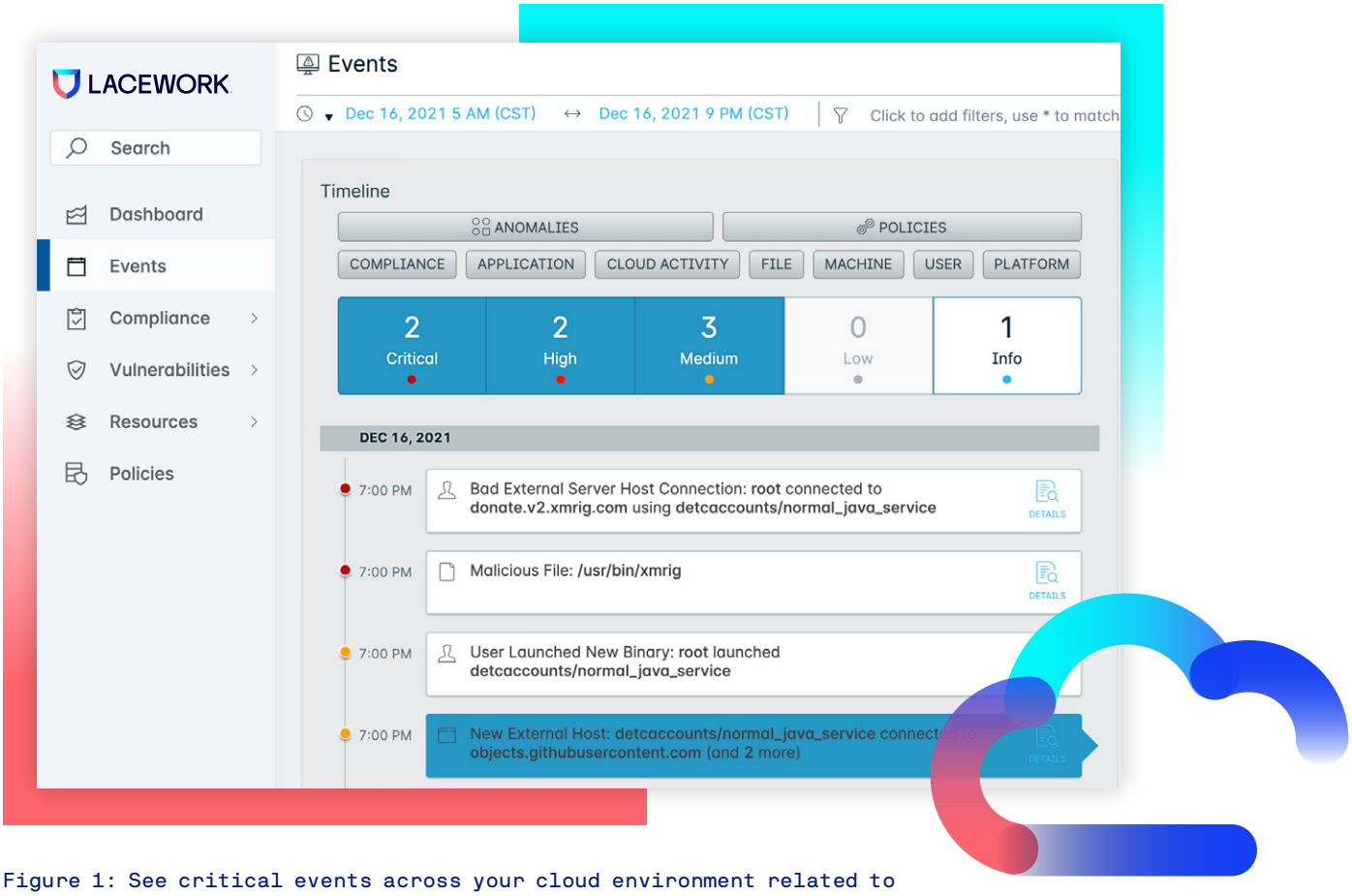


Figure 1: See critical events across your cloud environment related to exploit attempts of the Log4j vulnerability

By drilling into the last event, we see an internal Java service connecting to a new external host. The resulting event detail gives us rich information on the who, what, where, when, and why behind the event, including a Polygraph visualization showing the relationship and connections between the involved hosts and applications, and the anomalies Lacework identified. This will allow us to see exactly which workload is affected, which processes on that workload are vulnerable, and any network communications on that workload (either to other internal workloads or to external endpoints).

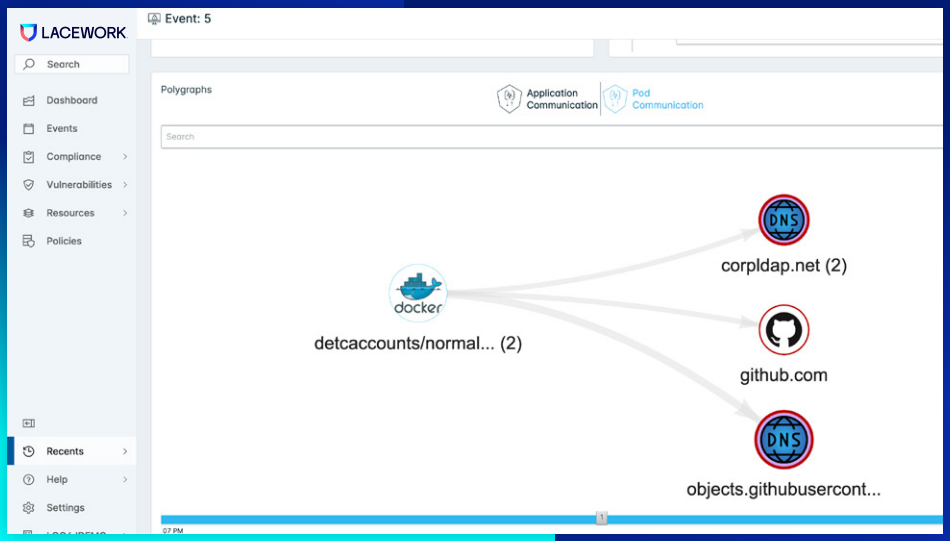


Figure 2: The Polygraph visualization shows external DNS requests made by a vulnerable Docker container.

The event detail also includes detail on the container image. In this example the image shows 3 critical vulnerabilities, as shown to the right.

Given this information, we can drill deeper into this workload and container. For the specific container, we can see the Polygraph visualization showing application communications and see that it is connecting to an external domain, v2xmrig.com, which is associated with cyptojacking. This is the result of the Log4j exploit.

We can then take appropriate steps to address the threat which likely would include destroying and rebuilding the container with the Log4j patch, and also blocking internal connections to these anomalous external hosts [below].

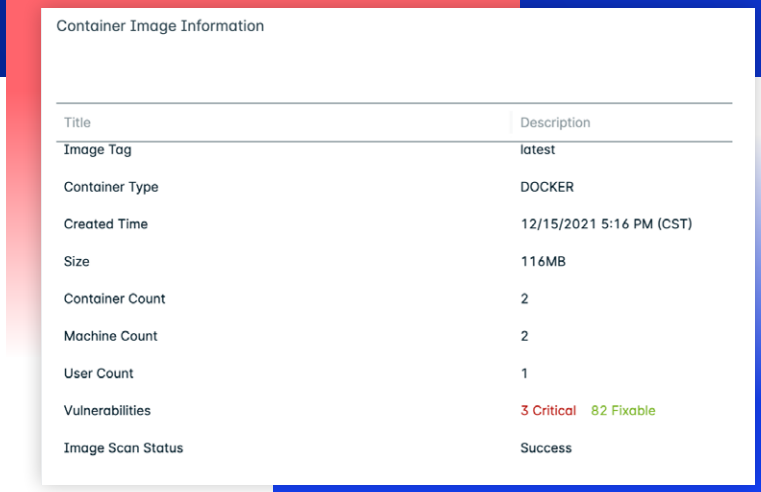


Figure 3: See the vulnerability details for the Docker container image.

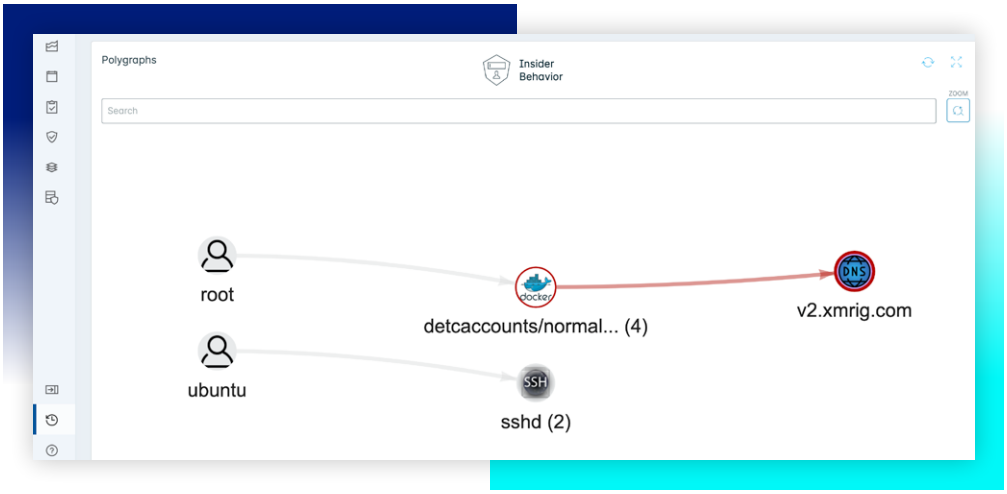


Figure 4: This shows the container connecting to a cryptomining site.

STEP TWO

Look for all installations of Log4j in your environment

Identifying where Log4j resides in your environment or applications can be difficult. There are several different ways the library can be installed, so there isn't one place to look. Sometimes it's installed as an application dependency, but other times, it may be installed by a 3rd party application without the end user's knowledge.

How to do it

One open source tool that can be used is the log4shell by LucaSec. This tool will scan your project and look for log4j libraries that are vulnerable to this attack. Additionally, Lacework can scan your workloads and containers for vulnerable versions of the Log4j library. The Lacework agent scans the OS and application libraries and flags systems where it sees Log4j versions that are susceptible to the CVE.

It's also important to look for 3rd party software in your environment. Many cloud services and packaged software are vulnerable to this vulnerability as well. Doing a thorough review of what is installed in your environment is important to help mitigate the risks.

How Lacework helps

With Lacework you can run vulnerability assessments both at build and runtime to uncover vulnerable containers and hosts. By continuously monitoring, you can find newly reported vulnerabilities and prevent the re-introduction of vulnerabilities. You can also search the Lacework platform for specific vulnerabilities that are present in your containers and hosts.

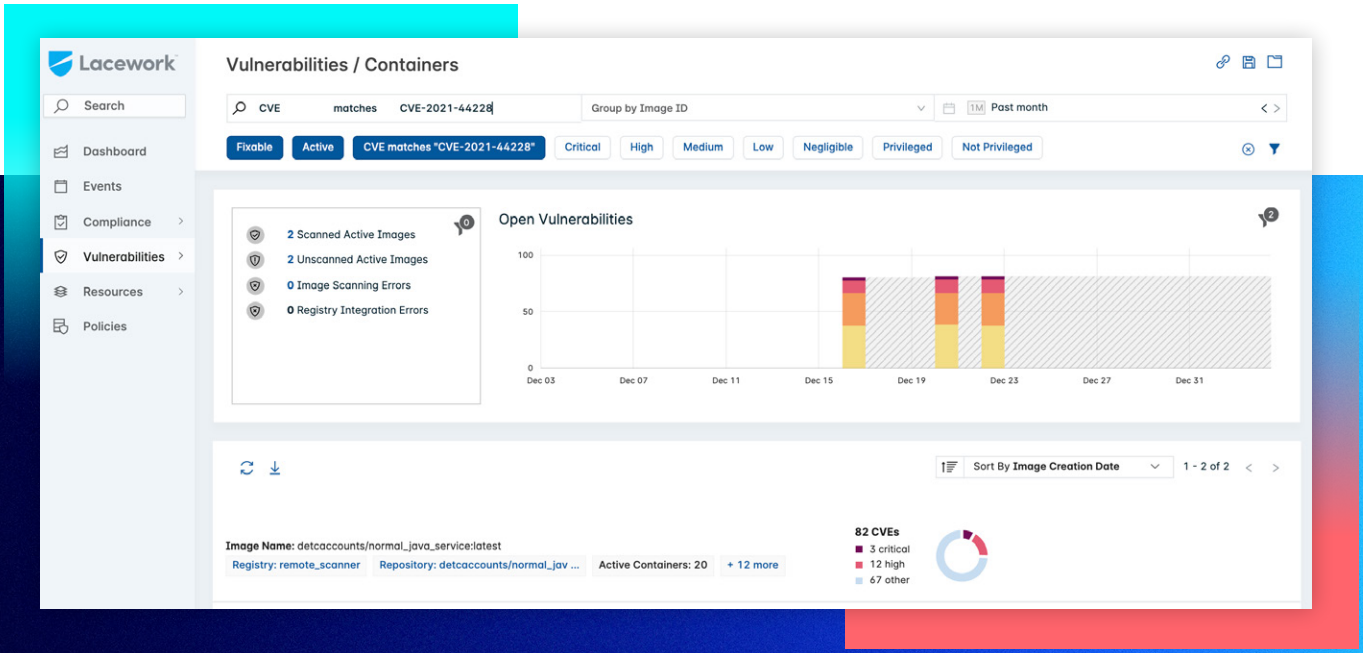


Figure 5: This screenshot shows a search for specific CVEs across all containers.

STEP THREE

Prioritize and remediate the affected systems

Once you have a list of the affected systems, it's time to start prioritizing, determining the course of action, and fixing. It can be overwhelming to patch vulnerabilities, so it's important to have an understanding of which systems to prioritize. Not every system has the same level of risk, so prioritizing will help make sure the systems with the most exposure, from a data or network perspective, will get first priority.

One factor of this vulnerability to keep in mind: Log4j is a library, not a standard software package. For some software that is impacted by this vulnerability, it will be up to the vendors to patch and push updates out to their customers.

How to do it

Start by focusing on the highest priority systems. For example, include systems with PCI/PHI or with sensitive company or personally identifiable data. You could also prioritize based on active vs inactive containers. Once those are patched, move to the remaining systems that are externally accessible. Attackers will be scanning public IP ranges, so it's important to focus on our IP ranges open to the world like our web servers or load balancers. After these are fixed, you can move to the remaining systems to make sure the remediation process is complete. Some may require coordination between different teams and testing, so you can take other mitigating controls in the interim.

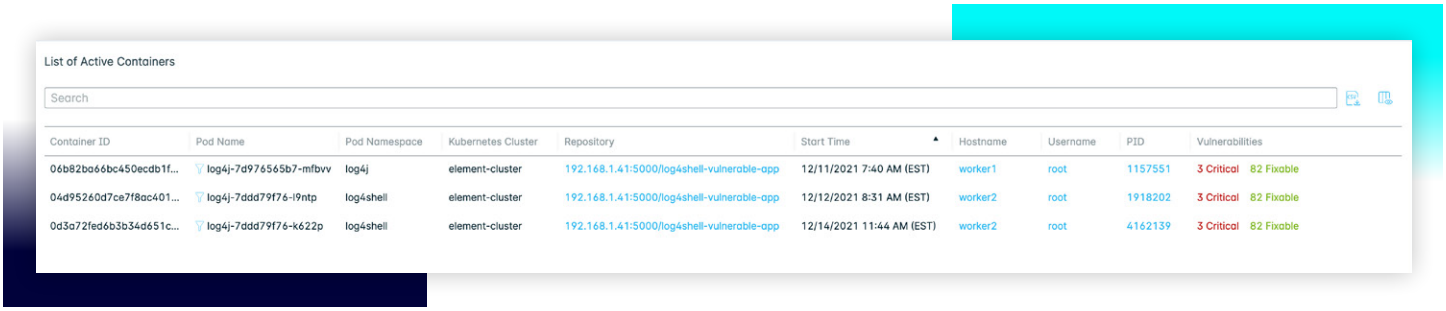


Figure 6: This screenshot shows active containers found with the Log4j vulnerability.

How Lacework helps

With Lacework, you can prioritize based on context about the systems. For example, active containers with critical vulnerabilities, including a specific CVE.

STEP FOUR

Continuously monitor your environment for exploit attempts from this event – and others that will follow

This one is so important, we decided it should be mentioned twice. The Log4j CVEs are some of the largest and most serious vulnerabilities discovered in recent years. Due to the pervasiveness of the library in modern software, we will likely see the effects of this continuing over the next year. Since this is a difficult vulnerability to patch, it's critical to continuously monitor your environment to watch for systems that were missed during patching or 3rd party software in your environment that hasn't been patched.

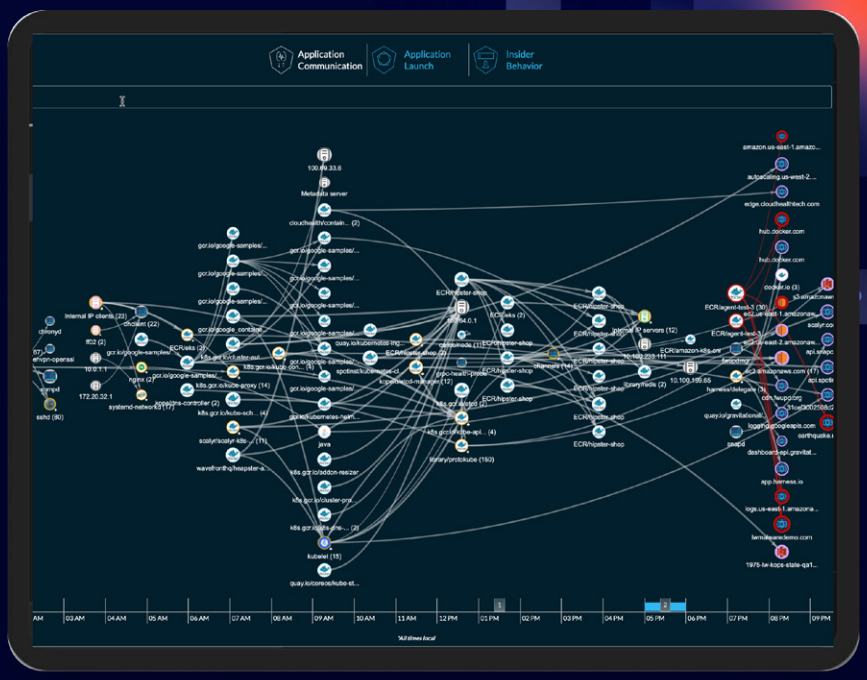
Lacework can help with the continuous monitoring using Polygraph and proactive vulnerability scanning to ensure unusual activity is flagged and alerted for investigation. Scanning inline or via registry as a part of your development process with the Lacework scanner will help you monitor CVEs and ensure that you are always using the most secure libraries. Lacework's machine learning will continually assess your cloud environment and monitor for activity indicating an attacker exploiting this vulnerability, along with any other suspicious activity happening in your environment. With continuous visibility into activity across your environment, you can not only better defend against Log4j attacks, but you can also better detect future threats. With Lacework, you can uncover anomalous behavior even before a vulnerability is known, widely publicized, and patched — helping security teams find potential exploits and take action faster.

Conclusion

The Log4j vulnerability is an unfolding situation that we are closely watching. To reduce risk to your business, it's critical to tie together a complete view of vulnerable systems with continuous monitoring of your environment for signs of compromise. We will update the Lacework blog on a regular basis, including new research from Lacework Labs. For the latest information and resources, please visit lacework.com/log4j.

Whether you're a Lacework customer or not, we're here to help with our free Cloud Care Rescue Program. Get access to:

- **Free 14-Day Cloud Threat Hunting Assessment**
  To help you quickly handle Log4j, our cloud security experts will work with you to find all vulnerable systems across your entire cloud and container environments and continuously monitor for active signs of compromise
- **Complimentary Coverage Booster for Customers**
  Existing customers can temporarily expand coverage of Lacework across your cloud environments to aid in vulnerability discovery and continuous monitoring.
- **24/7 Rescue Hotline support at 888-599-9519**



Ready to chat?

Request a demo