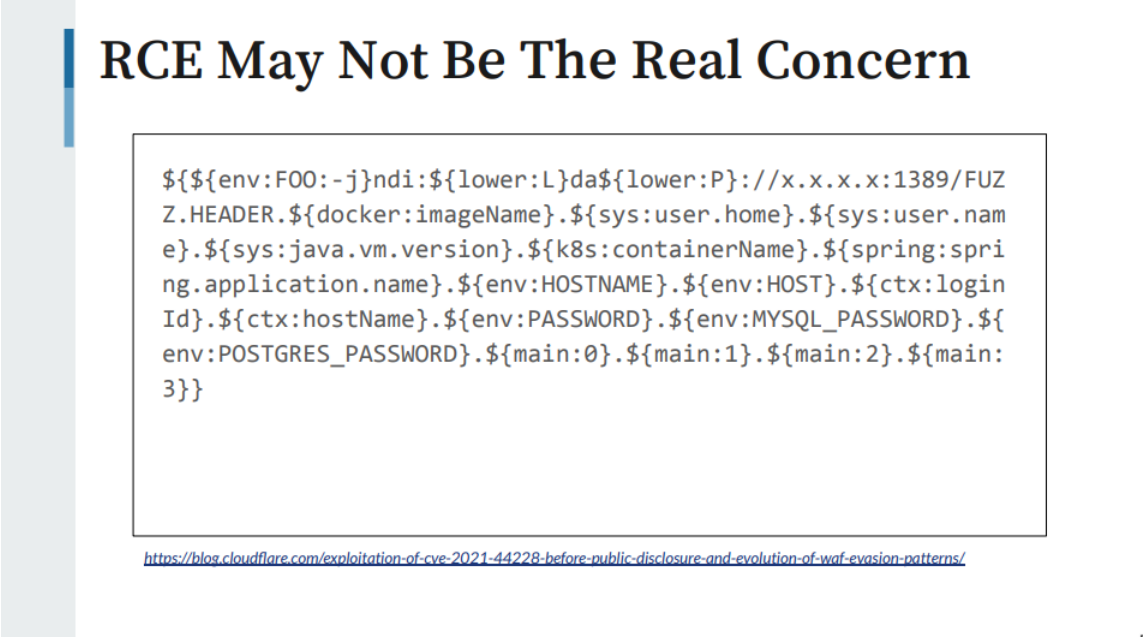
CS615 Meetup Report

Event: How cloud services help to mitigate Log4Shell and its friends
Venue: Virtual through meetup.com - https://www.meetup.com/PhillyJUG/events/284542274/
Event Date: 4/12/22
Guest Speaker: Keith Gregory from Chariot Solutions

This week, I attend a virtual meetup titled “How cloud services help to mitigate Log4Shell and its friends” hosted by the group Philly JUG (Java Users Group). The main discussion at the meetup is how it is possible to mitigate vulnerabilities such as Log4Shell using cloud infrastructure solutions such as AWS. At the event, guest speaker Keith Gregory from Chariot Solutions discussed some history about the software and vulnerability and explained some methods to secure your servers from attacks.

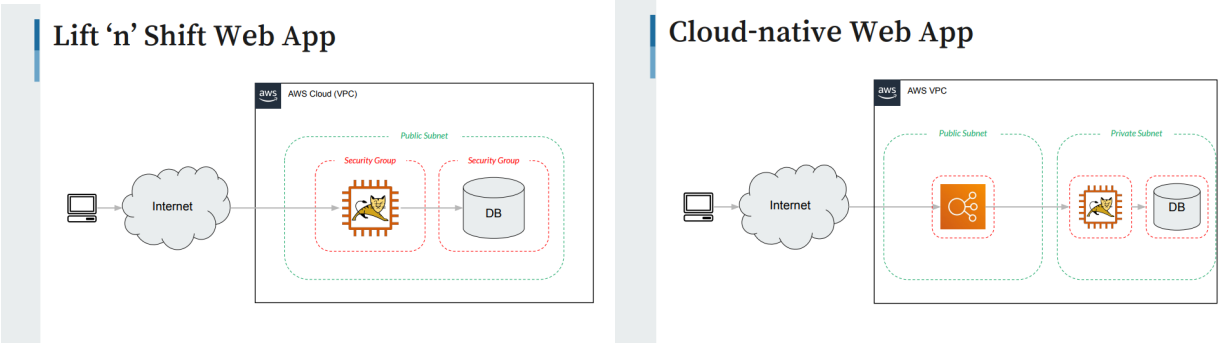
I chose this event as I found the Log4Shell vulnerability to be very interesting, as it allows users to execute code on a remote server without being granted access by a system administrator. Thus, the topic of preventing exploits of this nature is integral to system administrators who are trying to keep their services secure, and inevitably will be an issue if you are in charge of any public-facing services. The following is a summary of what was discussed at the meetup.

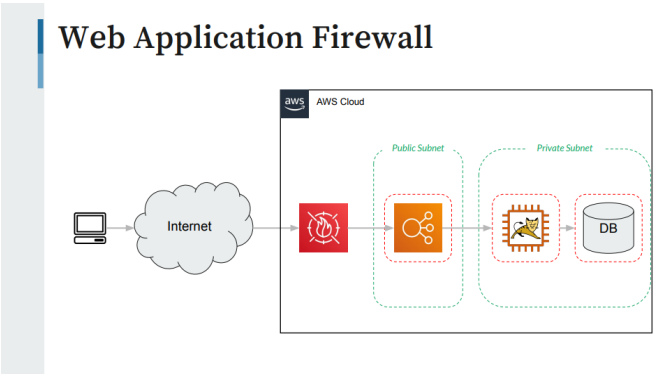
Just as discussed in class, Log4Shell originated from the Apache Log4j package and is detailed in CVE-2021-44228. This vulnerability allowed for people to run unauthorized code on a target's system due to a lack of sanitization when logging data. While remote code execution was a major concern, this vulnerability allowed for attackers to extract data from the server, such as passwords and keys stored in the host's environment:



The first step to mitigating exploits like this is to guard the entrance. AWS accomplishes this by providing users with a virtual private cloud, which contains subnets and security groups that can be configured to help prevent attackers from getting in. For example, a user can limit access to certain servers based on ports or even limited to only other machines within the same security group or subnet. Ultimately, a system administrator wants to limit the number of entrances, which can be done by exposing a machine on a public subnet to talk with the internet, but it also talks with a private subnet. This setup allows for servers that may be running vulnerabilities such as Log4Shell to be locked away without direct public access, but can still

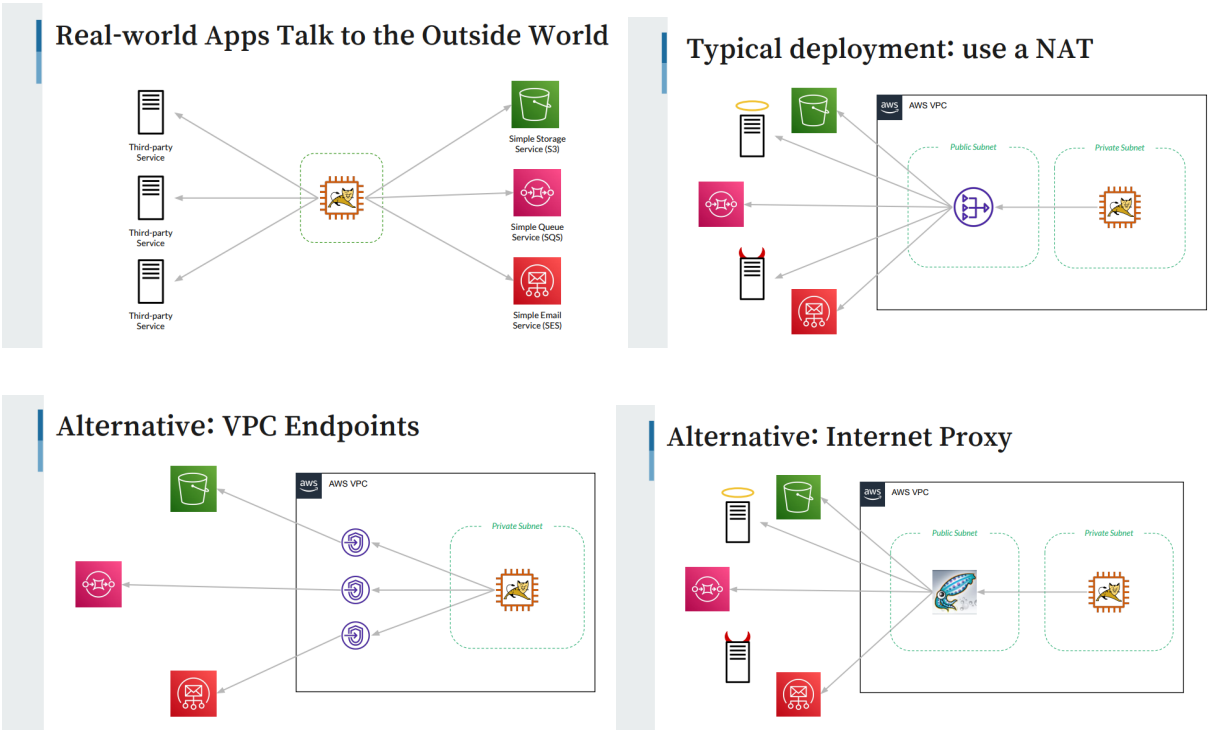
communicate to the outside world through the machines hosted on the public subnet. This however is not always feasible, as with modern infrastructure the servers that you know are good and can access your subnets may change due to the volume the servers need to handle (similar to how something like Google will use many IP addresses for the same site). “This is something that would have worked ten years ago, but not today”.





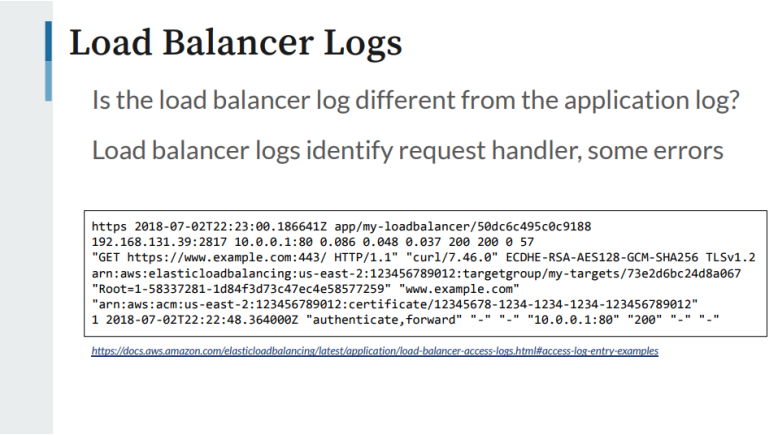
The second step to mitigating these exploits is controlling the blast radius. A system administrator can leverage cloud solutions such as AWS to limit the roles that each server can play. For example, certain servers can be set to a state where it only has permissions to send data to external sources but only receive incoming requests from authorized servers or users. By doing this, it creates a wall between your servers and malicious users. This can also be applied by location, where in the case of AWS an administrator can configure a server to be limited to only accept requests from certain data center regions. Additionally, another method of defense

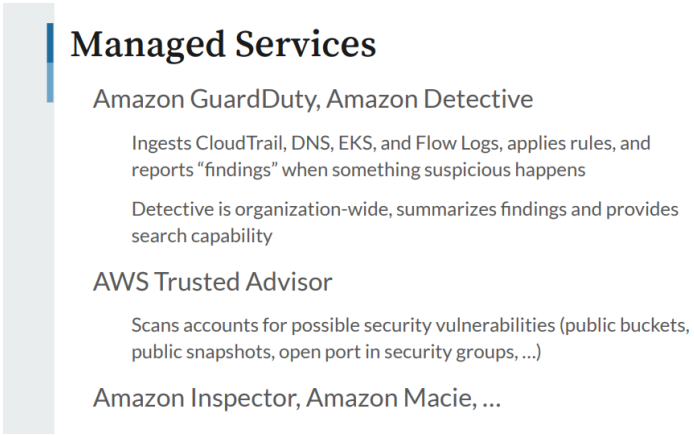
would be to make use of a proxy, allowing for splitting the services across multiple machines, and even networks, to prevent a vulnerability from affecting other machines that would normally be on the same network in a non-cloud hosted solution.



The third step to mitigating these exploits is properly monitoring of your servers' log files. An advantage of using most cloud services is that by default they will log all activity for their own statistics, and often expose these logs to their customers. By cross-checking your own logs, system administrators can check for discrepancies in a wide range of segments, ranging from shell execution due to faulty logging software to access attempts from unauthorized addresses on ports that would interface with software that can be exploited. AWS has services such as GuardDuty and Detective which can be used to assist system administrators in identifying potential malicious behavior. As we recently discussed in class, monitoring system

logs can be a powerful way to diagnose an issue or rogue user, and this vulnerability is a real-world example of such a use case.





What I learned from participating in this event were specific methods to secure your infrastructure using services such as AWS. While I was familiar with some concepts such as subnets and security groups, I had never thought of using them in a way that would allow for systems to be secure from the outside world yet also able to serve public users. Additionally, I learned that while remote code execution was the most talked-about worry regarding Log4Shell,

the possibility to extract data from the environment is also something to be concerned about, and is another thing that system administrators will need to try and secure.