HORIZON3.ai

~~TRUST BUT~~ VERIFY

# Log4Shell Attack Scenarios

We have narrowed it down to two separate attack scenarios, both which lead us to the same conclusion: **we have work to do.**

## External Attack

Consider an attack scenario where the adversary is using an Internet connection to access a web application. As we speak, attackers are engaging in a 'spray and pray' campaign. In other words, they are indiscriminately targeting online hosts for indications of log4j2 use. This indiscriminate fire comes in a couple varieties:

### HTTP header manipulation and form field fuzzing:

- In the case of header manipulation, attackers are replacing information normally found in an http header with a string of characters which they hope will invoke the vulnerability if their target is using a vulnerable version of log4j.
- In the case of form field fuzzing, attackers are looking for web applications with user-input fields. In those fields attackers are submitting the same string used in http header manipulation to achieve the same effect. As part of their effort, attackers are setting up ad-hoc hosts to receive the callback from their victim hosts.

**In this scenario, they are using an LDAP server pre-loaded with an exploit the victim will ask for. Discovery can be initiated by a mass targeting campaign using http header manipulation where a jndi query is included in the http header.**

### Entering a jndi lookup string into an existing user input field:

- The attacker will not know if their 'intended' target is vulnerable. They are simply inputting a string to trigger a jndi lookup using HTTP header manipulation or form field fuzzing.
- If the target is using log4j2, all inputs will be logged as intended by log4j2.
- If the target is using a vulnerable version of log4j, the attacker's inputs will be executed. In this case, they have triggered a jndi query to their ad-hoc LDAP server. The object on the LDAP server is what is used to initiate the exploit. The attacker-controlled ad hoc LDAP server will serve the referenced object identified in the jndi query string.

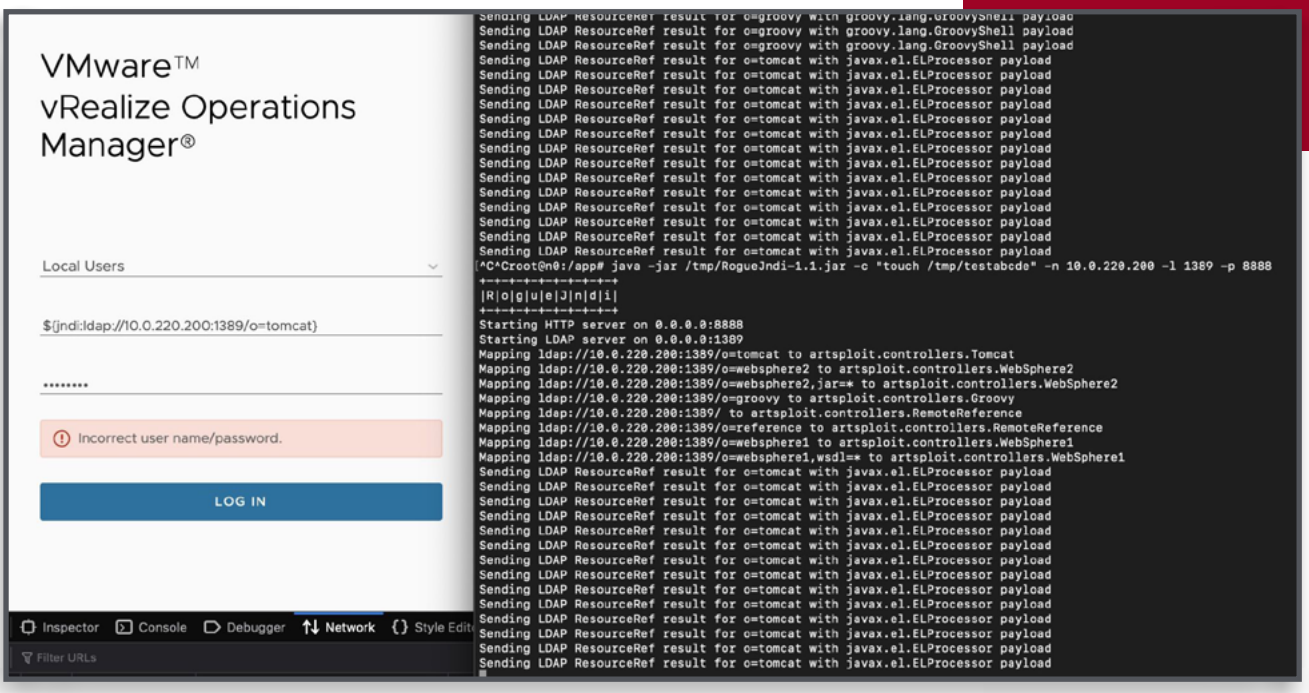
**In this scenario, the attacker initiated the victim to reference the exploit. The vulnerable application will load the attacker's exploit as intended, Java deserializes (downloads) the class and executes it… enabling the attacker to gain remote code execution (RCE) privileges on the host running the application.**

# Internal Attack

**Once an adversary has gained access to your network, they can leverage Log4Shell to achieve lateral movement and Remote Code Execution (RCE). The potential impact of RCE on your internal network is very, VERY high.**

This is where attackers achieve critical impacts. This is also where your crown jewels are used, often by critical Java-based third-party software components like WebSphere, CICS WebServices, Hadoop, Elastic, Atlassian, and numerous others — all of which may be vulnerable.

Moving and communicating laterally to inject malicious payloads and callouts from your enterprise is how an attacker establishes persistence and pervasiveness. You can run a NodeZero pentest to enumerate every host available and check for log4j2 java-based logging underlying your critical applications.




## Assume Compromise

The first step in mitigation is conducting a thorough analysis of your network infrastructure. That's why we pushed an update to NodeZero to detect and pentest Log4Shell in order to understand the depth of compromise.

Some vendors, like VMware and Cisco, have had to issue advisories for dozens of affected products. A variety of security tools have attempted to assist companies in remediating Log4Shell. Most of these tools stop at the point of detecting Log4Shell, with varying degrees of accuracy. NodeZero detects if an application is vulnerable by checking for DNS requests originating from the application. NodeZero is able to paste a JNDI string into vRealize, which then triggers a compromise.

# Putting NodeZero to work is simple:

- Set the scope of IP addresses, and NodeZero goes to work scanning and identifying your network hosts.
- NodeZero enumerates all hosts in scope and identifies which ones are using log4j2.
- NodeZero executes a general fuzzing of HTTP headers, paths, and query parameters to check for Log4Shell vulnerability.
- The NodeZero attack server will receive a request from you vulnerable hosts, which establishes a connection confirming the Log4Shell vulnerability.
- The vulnerable application will load NodeZero's exploit when Java deserializes (downloads) the object it requested.
- Once the object is received and executed, it enables NodeZero to gain remote code execution (RCE) privileges on the target hosting the vulnerable application.

**Customers can schedule a cooperative review with the Horizon3 AI Customer Success Team by emailing customer.success@horizon3.ai.**

## Ready to Learn More?

NodeZero is an Autonomous Penetration Testing as a Service (APTaaS) that helps organizations **find and fix attack vectors before attackers can exploit them**. It is safe to run in production and requires no persistent or credentialed agents.

**Sign up for your free demo today.**



HORIZON3.ai
TRUST BUT VERIFY