[CVE-2020-13924] Apache Ambari Arbitrary File Download Vulnerability

Posted to user@ambari.apache.org (list.html?user@ambari.apache.org)

Szabolcs Beki - Sunday, February 7, 2021 6:21:53 AM EST

Hello All,

Just about to post this vulnerability to Mitre that affects users of
Ambari 2.6.2.2 and earlier.
Malicious users can construct file names for directory traversal and
traverse to other directories to download files. Mitigation is to upgrade
to any later version.

Credit : threedr3am

Vulnerability details:

This vulnerability is mainly due to the use of "String requestURI =
httpRequest.getRequestURI();" in the authentication filter
(org.apache.ambari.server.security.authorization.AmbariAuthorizationFilter)：

```
@Overridepublic void doFilter(ServletRequest request, ServletResponse response,
FilterChain chain) throws IOException, ServletException {
HttpServletRequest httpRequest = (HttpServletRequest) request;
HttpServletResponse httpResponse = (HttpServletResponse) response;

String requestURI = httpRequest.getRequestURI();

SecurityContext context = getSecurityContext();

Authentication authentication = context.getAuthentication();

AuditEvent auditEvent = null;
....
}
```

Because when the web server processes the request, when accessing a
path like "/everyone-has-permission-path/..;/admin-has-permission-path",
the web server will return the resource "admin-has-permission- path",
but "httpRequest.getRequestURI()" in the filter will return the path
"/everyone-has-permission-path/..;/admin-has-permission-path", so in
the following code Will result in permission to pass the match：

```
@Override
public void doFilter(ServletRequest request, ServletResponse
response, FilterChain chain) throws IOException, ServletException {
...
if (authentication == null || authentication instanceof
AnonymousAuthenticationToken) {
...
}
if (authentication == null || authentication instanceof
AnonymousAuthenticationToken ||
!authentication.isAuthenticated()) {
...
} else if (!authorizationPerformedInternally(requestURI)) {
boolean authorized = false;

if (requestURI.matches(API_BOOTSTRAP_PATTERN_ALL)) {
authorized = AuthorizationHelper.isAuthorized(authentication,
ResourceType.CLUSTER,
null,
EnumSet.of(RoleAuthorization.HOST_ADD_DELETE_HOSTS));
}
else {
...
}

...
}
...
}
```

In fact, when I need to access the api under "/users.*", I only need
to use "/bootstrap/..;/users" to bypass certain authentication checks.

Of course, the APIs under "users.*" may require certain permissions to
access, but this is just an example, which means that in this way, you
will be able to bypass the authentication check to access other APIs
that require authentication to access.

Szabi