





com/thoughtworks/go/agent/service/ AgentUpgradeService.java	108	2	SECURITY	UNSAFE_HASH_EQUALS	 
com/thoughtworks/go/validation/ ChecksumValidator.java	40	2	SECURITY	UNSAFE_HASH_EQUALS	 

```
private void validateMd5(String currentMd5, CloseableHttpResponse response, String agentContentMd5Header, String what) {
    final Header md5Header = response.getFirstHeader(agentContentMd5Header);
    if (!"".equals(currentMd5)) {
        if (!currentMd5.equals(md5Header.getValue())) {
            jvmExit(jvmExit(what, currentMd5, md5Header.getValue()));
        }
    }

    HttpGet getAgentLatestStatusGetMethod() {
        return new HttpGet(urlService.getAgentLatestStatusUrl());
    }
}
"agent/src/main/java/com/thoughtworks/go/agent/service/AgentUpgradeService.java" 118L, 5265C 108,13
```

```
if (expectedMd5.equals(artifactMd5)) {
    checksumValidationPublisher.md5Match(effectivePath);
} else {
    checksumValidationPublisher.md5Mismatch(effectivePath);
}
}
}
"common/src/main/java/com/thoughtworks/go/validation/ChecksumValidator.java" 46L, 1746C 40,2
```

Unsafe comparison of hash that are susceptible to timing attack

An attacker might be able to detect the value of the secret hash due to the exposure of comparison timing. When the functions `Arrays.equals()` or `String.equals()` are called, they will exit earlier if fewer bytes are matched.

Vulnerable Code:

```
String actualHash = ... if(userInput.equals(actualHash)) { ... }
```

Solution:

```
String actualHash = ... if(MessageDigest.isEqual(userInput.getBytes(),actualHash.getBytes())) { ... }
```

References

CWE-203: Information Exposure Through DiscrepancyKey