**Report for SSE-exercise 4**

Name: Jianqi Zeng

Student Id: a1752399

Git repo (Including results): N/A

Repo 1:

|  |  |
| --- | --- |
| <https://android.googlesource.com/platform/packages/apps/Exchange> | 0d1a38b1755efe7ed4e8d7302a24186616bba9b2 |

Repo 2:

|  |  |
| --- | --- |
| <https://github.com/apache/tomcat> | 5496e193a89b8b8b3177e516358df2f07ab852b3 |

Repo 3:

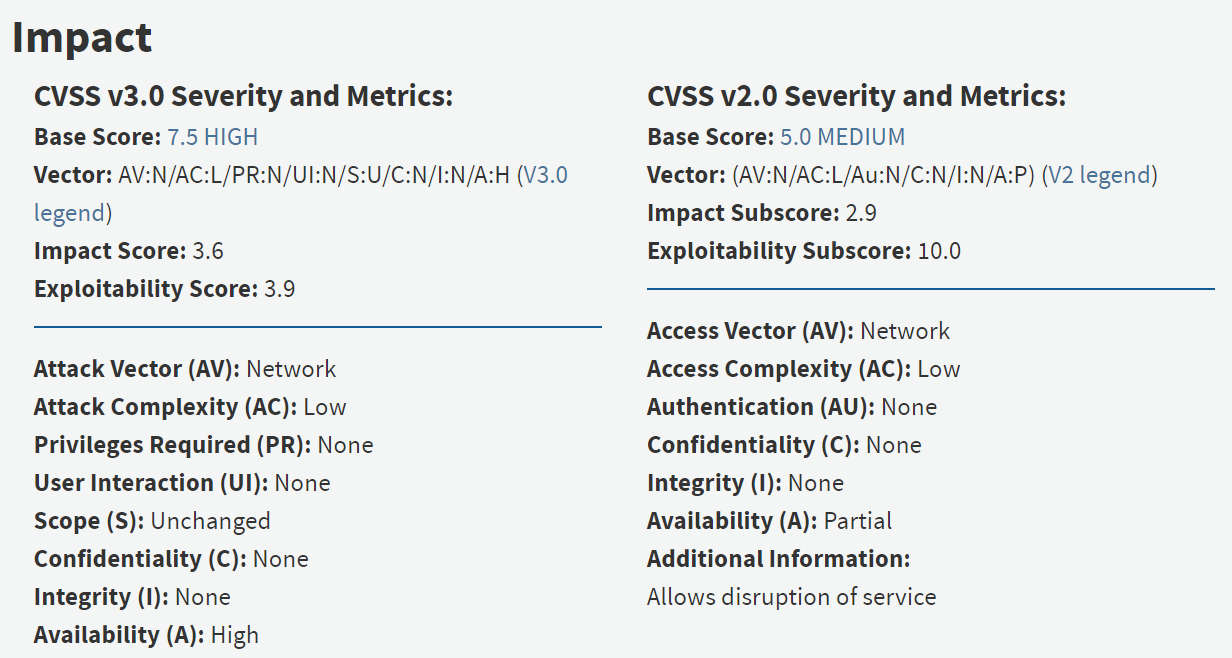
|  |  |
| --- | --- |
| <https://github.com/cloudfoundry-incubator/credhub> | 46ae8627a6887d0c810905585b40845193b9a9f8 |

**Identify each CWE:**

CVE-2017-5650:

* Improper Resource Shutdown or Release ([CWE-404](http://cwe.mitre.org/data/definitions/404.html))

Detail: In Apache Tomcat 9.0.0.M1 to 9.0.0.M18 and 8.5.0 to 8.5.12, the handling of an HTTP/2 GOAWAY frame for a connection did not close streams associated with that connection that were currently waiting for a WINDOW\_UPDATE before allowing the application to write more data. These waiting streams each consumed a thread. A malicious client could therefore construct a series of HTTP/2 requests that would consume all available processing threads.

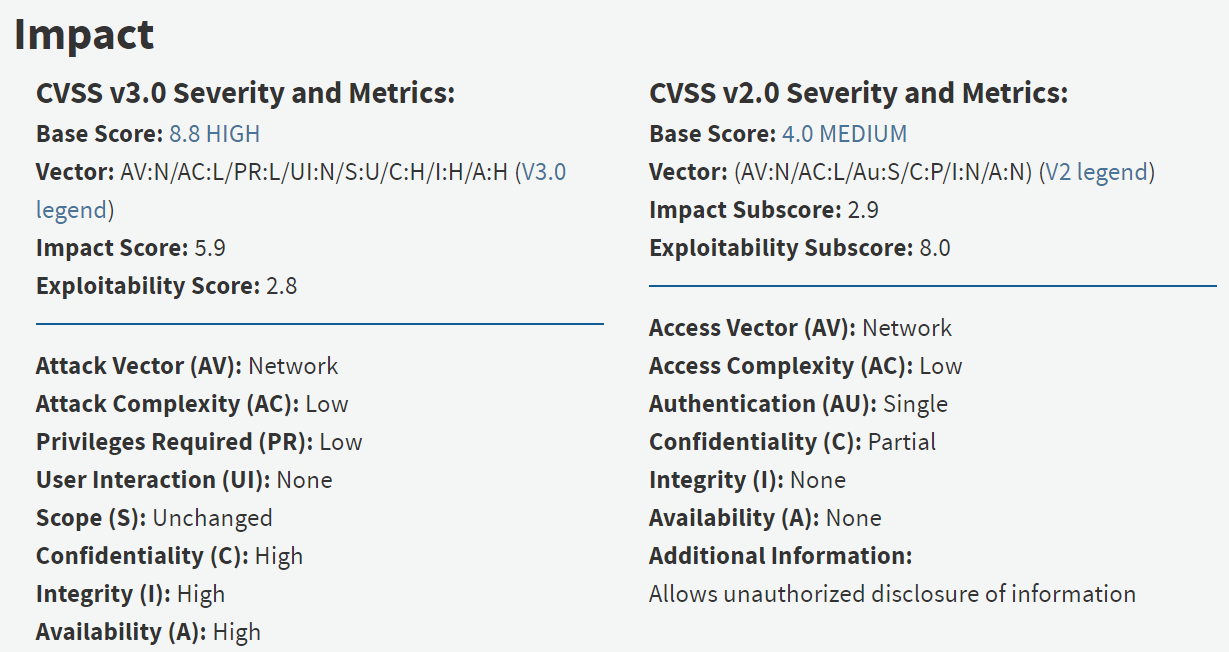


CVE-2017-8038:

* Insufficient Information ([NVD-CWE-noinfo](http://cwe.mitre.org/data/definitions/CWE-noinfo.html))

Detail:

In Cloud Foundry Foundation Credhub-release version 1.1.0, access control lists (ACLs) enforce whether an authenticated user can perform an operation on a credential. For installations using ACLs, the ACL was bypassed for the CredHub interpolate endpoint, allowing authenticated applications to view any credential within the CredHub installation.

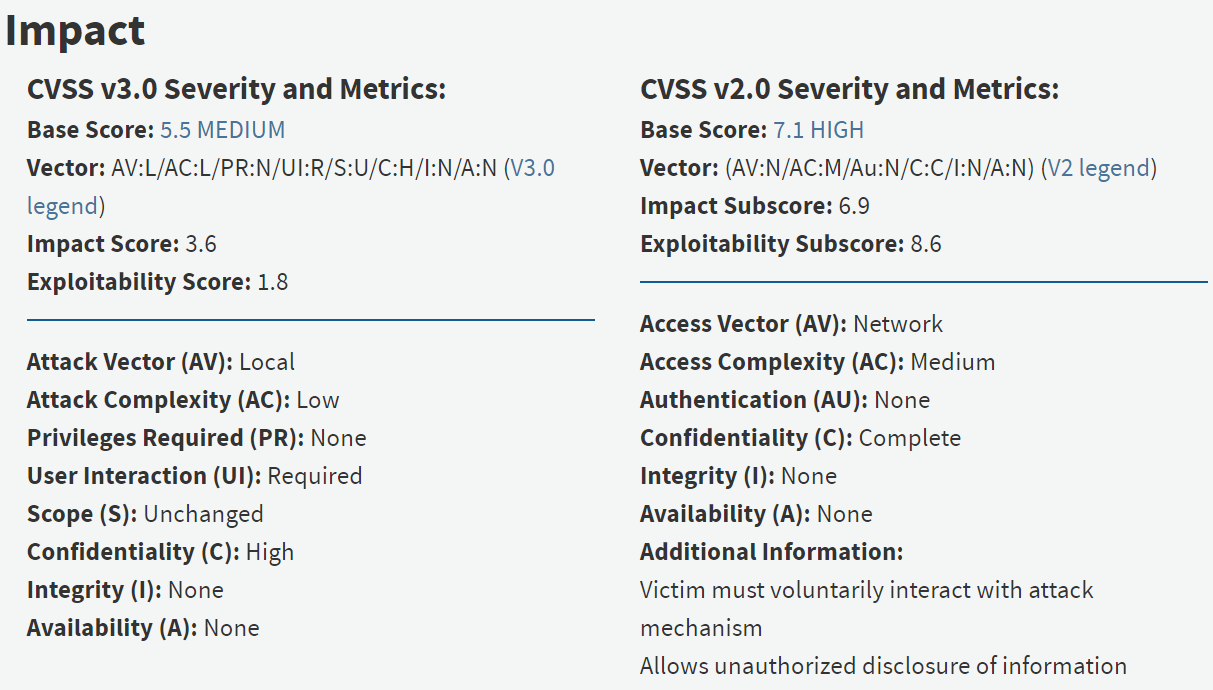


CVE-2016-2415:

**Vulnerability Type**:

Information Exposure ([CWE-200](http://cwe.mitre.org/data/definitions/200.html))

Detail: exchange/eas/EasAutoDiscover.java in the Autodiscover implementation in Exchange ActiveSync in Android 5.0.x before 5.0.2, 5.1.x before 5.1.1, and 6.x before 2016-04-01 allows attackers to obtain sensitive information via a crafted application that triggers a spoofed response to a GET request, aka internal bug 26488455.



Code explanation:

1. Check if it is the initial commit. First, I need to get all affected files by git dif and passing parameter “-U0”, fixing commit and previous commit.

Then I use git blame by passing ‘-w’, I got the worst commit which is VCC commit, and by using git show passing the worst commit (most frequent one) and affected files which got on the previous step, I can check if there is any other commit ahead of this one, if not then this will be the initial commit.

1. Determine if it is the same developer. I first use git shortog by passing parameter “-sne”, “all” get the all related commit authors, and then I loop the current fixing commit author in the commit authors to see if match, if match then the fixing commit author and the VCC author is the same developer.
2. Determine how long in between fixing commit and VCC commit. By using git log passing parameter “VCC\_commit”, “fixing\_commit”, “affected\_files”, “%a %b **%d** %H:%M:%S %Y” to format the date, we can get both commit date, and then we can see the time difference of two commits, I use variable date\_VCC to store the VCC commit timestamp and date\_Fixing\_commit to store the fixing commit timestamp, then date\_Fixing\_commit – date\_VCC commit we can get the time difference in timestamp, and then I use the parameter I mentioned previously to transfer the timestamp to day.
3. By git show, to see the relevant commits messages and analysis the d.

**CVSS:**

*The CVSS base score metrics for Version 2 are:*

Attack Vector (AV), Access Complexity (AC), Authentication (Au), Confidentiality Impact (C)\*, Integrity Impact (I)\*, Availability Impact (A)\*.

*The CVSS base score metrics for Version 3 are:*

Attack Vector (AV)\*, Attack Complexity (AC)\*, Privileges Required (PR)\*, User Interaction (UI)\*, Scope (S)\*, Confidentiality Impact (C)\*, Integrity Impact (I)\*, Availability Impact (A)\*.

