



Site Search

[Full Disclosure](#) mailing list archives[By Date](#) [By Thread](#)

List Archive Search



[SYSS-2021-062] Oracle Database - Weak NNE Integrity Key Derivation

From: Moritz Bechler <moritz.bechler () syss de>

Date: Fri, 10 Dec 2021 10:18:29 +0100

Advisory ID: SYSS-2021-062
Product: Database
Manufacturer: Oracle
Affected Version(s): 12.1.0.2, 12.2.0.1, 19c
Tested Version(s): 18c
Vulnerability Type: Inadequate Encryption Strength (CWE-326)
Risk Level: Medium
Solution Status: Fixed
Manufacturer Notification: 2021-03-17
Solution Date: 2021-08-07
Public Disclosure: 2021-12-10
CVE Reference: CVE-2021-2351
Author of Advisory: Moritz Bechler, SySS GmbH

Overview:

Oracle Database is a general purpose relational database management system (RDBMS).

The manufacturer describes the product as follows (see [1]):

"Oracle database products offer customers cost-optimized and high-performance versions of Oracle Database, the world's leading converged, multi-model database management system, as well as in-memory, NoSQL and MySQL databases. Oracle Autonomous Database, available on premises via Oracle Cloud@Customer or in the Oracle Cloud Infrastructure, enables customers to simplify relational database environments and reduce management workloads."

To protect the client/server communication, a proprietary security protocol "Native Network Encryption" (NNE) is used.
A TLS-based alternative can optionally be configured.

NNE's integrity protection mechanism deliberately weakens the key used for computing per-packet message authentication codes (MACs).

Vulnerability Details:

When analyzing the protocol details, SySS found out that depending on the selected hash algorithms, one of two key generation schemes is used. Both are seeded with material from the established session key. However, even for the AES-based key generator, which is used when modern cryptographic primitives are selected, the session key is truncated to 40 bits.

For more details on the protocol and MAC computation, refer to our paper [4].

Brute-force cracking of that key, for example if only integrity but no encryption is enabled, is likely possible and allows malicious manipulation of transmitted database commands or data.

Proof of Concept (PoC):

The initialization of the key generator, as originally implemented, can be described with the following Python code, where SK is the established session key, and the initialization vector (IV) was exchanged in clear text during NNE negotiation.

```
mk = SK[0:5] + b'\xFF' + b'\x00' * 10
self.m = AES.new(mk, AES.MODE_CBC, iv=IV[0:16])
self.ms = b'\x00'*32
self.ms = s = self.m.encrypt(self.ms)
self.m = AES.new(s[0:16], AES.MODE_CBC, iv=s[16:32])

k1 = s[0:5] + b'\xB4' + s[6:16]
self.s2c = AES.new(k1, AES.MODE_CBC, iv=s[16:32])
self.s2cs = b'\x00' * 32

k2 = s[0:5] + b'\x5A' + s[6:16]
self.c2s = AES.new(k2, AES.MODE_CBC, iv=s[16:32])
self.c2ss = b'\x00' * 32
```

A per-packet key "k" is then generated like

```
self.c2ss = k = self.c2s.encrypt(self.c2ss)
```

and appended to the packet data as well as hashed using the selected hash algorithm.

Solution:

Update the Oracle Database servers and clients to the patched versions.
Enforce usage of a secured protocol version by setting the following options:

```
SQLNET.ALLOW_WEAK_CRYPTO_CLIENTS=FALSE (server-side)
SQLNET.ALLOW_WEAK_CRYPTO=FALSE (client-side)
```

Or use TLS-based transport security instead of Native Network Encryption.

More information:

<https://www.oracle.com/security-alerts/cpujul2021.html>
<https://support.oracle.com/rs?type=doc&id=2791571.1> (customer account required)

Disclosure Timeline:

2013-03-02: Vulnerability discovered
2021-03-17: Vulnerability reported to manufacturer
2021-07-20: Initial patch release by manufacturer
2021-08-07: Final patches released by manufacturer
2021-12-10: Public disclosure of vulnerability

References:

- [1] Product website for Oracle Database
<https://www.oracle.com/database/>
- [2] SySS Security Advisory SYSS-2021-062
<https://www.syss.de/fileadmin/dokumente/Publikationen/Advisories/SYSS-2021-062.txt>
- [3] SySS Responsible Disclosure Policy
<https://www.syss.de/en/responsible-disclosure-policy>
- [4] Paper "Oracle Native Network Encryption"
https://www.syss.de/fileadmin/dokumente/Publikationen/2021/2021_Oracle_NNE.pdf

Credits:

This security vulnerability was found by Moritz Bechler of SySS GmbH.

E-Mail: moritz.bechler () syss de
Public Key: https://www.syss.de/fileadmin/dokumente/PGPKeys/Moritz_Bechler.asc
Key ID: 0x768EFE2BB3E53DDA
Key Fingerprint: 2C8F F101 9D77 BDE6 465E CCC2 768E FE2B B3E5 3DDA

Disclaimer:

The information provided in this security advisory is provided "as is" and without warranty of any kind. Details of this security advisory may be updated in order to provide as accurate information as possible. The latest version of this security advisory is available on the SySS website.

Copyright:

Creative Commons - Attribution (by) - Version 3.0
URL: <http://creativecommons.org/licenses/by/3.0/deed.en>

Sent through the Full Disclosure mailing list
<https://nmap.org/mailman/listinfo/fulldisclosure>
Web Archives & RSS: <http://seclists.org/fulldisclosure/>

[By Date](#) [By Thread](#)

Current thread:

[SYSS-2021-062] Oracle Database - Weak NNE Integrity Key Derivation *Moritz Bechler (Dec 10)*

Site Search

Nmap Security Scanner

Ref Guide

Install Guide

Docs

Download

Nmap OEM

Npcap packet capture

User's Guide

API docs

Download

Npcap OEM

Security Lists

Nmap Announce

Nmap Dev

Full Disclosure

Open Source Security

BreachExchange

Security Tools

Vuln scanners

Password audit

Web scanners

Wireless

Exploitation

About

About/Contact

Privacy

Advertising

Nmap Public Source License