New Oracle Security Threat

Troy Ligon

Stealth Password Cracking Vulnerability

Esteban Martinez Fayo – *AppSecInc.com*

http://arstechnica.com/security/2012/09/oracle-database-stealth-password-cracking-vulnerability/

RISK ASSESSMENT / SECURITY & HACKTIVISM

Oracle Database suffers from "stealth password cracking vulnerability"

Weakness makes it trivial for attackers to crack Oracle Database user passwords.

by **Dan Goodin** - Sept 20 2012, 2:05pm EDT

BIG DATA HACKING 16

A weakness in an Oracle login system—used in the company's databases which grant access to sensitive information—makes it trivial for attackers to crack user passwords and gain entry without authorization, a researcher has warned.

The issue has been dubbed the "Oracle stealth password cracking vulnerability," by the researcher who discovered it, and the problem stems from a session key the Oracle Database 11g Releases 1 and 2 sends to users each time they attempt to log on, according to a report published Thursday by Threatpost. The key leaks information about a cryptographic hash used to obscure the plaintext password. The hash, in turn, can be cracked using off-the-shelf hardware, free software, and a variety of attack methods that have grown increasingly powerful over the past decade. Proof-of-concept code exploiting the weakness can crack an eight-character alphabetic password in about five hours using standard CPUs.

SUNCOAST ORACLE USERS' GROUP

https://threatpost.com/en_us/blogs/flaw-oracle-logon-protocol-leads-easy-password-cracking-092012

September 20, 2012, 9:53AM

Flaw in Oracle Logon Protocol Leads to Easy Password Cracking

by Dennis Fisher

Share Like 10 +1 4

18 Comments

Follow @DennisF

There is a serious vulnerability in the authentication protocol used by some Oracle databases, a flaw that could enable a remote attacker to brute-force a token provided by the server prior to authentication and determine a user's password. The attacker could then log on as an authenticated user and take unauthorized actions on the database. The researcher who discovered the bug has a tool that can crack some simple passwords in about five hours on a normal PC.

The vulnerability exists in Oracle Database 11g Releases 1 and 2 and is caused by a problem with the way the authentication protocol protects session keys when users try to log in. The first step in the authentication process when a client contacts the database server is for the server to send a session key back to the client, along with a salt. The vulnerability enables an attacker to link a specific session key with a specific password hash.

SUNCOAST ORACLE USERS' GROUP

https://threatpost.com/en_us/blogs/flaw-oracle-logon-protocol-leads-easy-password-cracking-092012

"This Session Key is a random value that the server generates and sends as the initial step in the authentication process, before the authentication has been completed. This is the reason why this attack can be done remotely without the need of authentication and also, as the attacker can close the connection once the Session Key has been sent, there is no failed login attempt recorded in the server because the authentication is never completed," said Esteban Martinez Fayo, a researcher at AppSec Inc., who discovered the flaw and will discuss it at the Ekoparty conference Thursday.

Editor's Pick

Oracle Releases Fix For Java CVE-2012-4681 Flaw

Oracle Warns Users About Privilege Escalation Bug in Database Server

Microsoft Publishes Workaround for Oracle Outside In Vulnerability

Threatpost Newsletter Sign-up

"Once the attacker has a Session Key and a Salt (which is also sent by the server along with the session key), the attacker can perform a brute force attack on the session key by trying millions of passwords per second until the correct one is found. This is very similar to a SHA-1 password hash cracking. Rainbow tables can' t be used because there is a Salt used for password hash generation, but advanced hardware can be used, like GPUs combined with advanced techniques like Dictionary hybrid attacks, which can make the cracking process much more efficient."

Fayo found the bug after noticing that there was an inconsistency in the way that clients and database servers handled failed log-in attempts. He found that log-in attempts with incorrect passwords were handled differently by the client than by the server and started looking more closely at why that was.

"Basically, I discovered that not all failed login attempts were recorded by the database. Looking closer at the issue, I located the problem in the way that one of the components of the logon protocol, the Session Key, was protected. I noticed that, in a certain way, the Session Key was leaking information about the password hash," he said.

Fayo said that Oracle has released a new version of the authentication protocol, version 12, which fixes this problem. However, he said that Oracle is not planning to fix the bug in version 11.1 of the protocol, and that even after applying the patch that includes the updated protocol, database servers are still vulnerable by default. Administrators need to change the configuration of the server in order to only allow the new version of the protocol.

SUNCOAST ORACLE USERS' GROUP

So How Would This Work?

1. Get the SALT (available through AUTH_VRF_DATA field)
2. Get the encrypted server session key (available through AUTH_SESSKEY field)
3. Brute force the AES 192-bit encrypted AUTH_SESSKEY to determine the SHA-1 password hash
4. Once you have the SALT and the SHA-1 hash value, brute force the password.

SUNCOAST ORACLE USERS' GROUP

So How Would This Work?

1. Get the SALT (field)
2. Get the en gh AUTH_SESSKE
3. Brute force the AES 192-bit encrypted AUTH_SESSKEY to determine the SHA-1 password hash
4. Once you have the SALT and the SHA-1 hash value, brute force the password.

Flaw Leaks Unencrypted version of this Key

SUNCOAST ORACLE USERS' GROUP

So How Would This Work?

1. Get the SALT (available through AUTH_VRF_DATA field)
2. Get the encrypted server session key (available through AUTH_SESSKEY field)
3. Brute force the AES 192-bit encrypted AUTH_SESSKEY to determine the SHA-1 password hash
4. Once you have the SALT and the SHA-1 hash value, brute force the password.

From SALT and a "guessed" password you get a SHA-1 hash. With that and the leaked Session Key, a brute force SHA-1 crack gives the decrypt key in about 5 hours for an 8-character password.

SUNCOAST ORACLE USERS' GROUP

So How Would This Work?

1. Get the SALT (available through AUTH_VRF_DATA field)
2. Get the encrypted server session key (available through AUTH_SESSKEY field)
3. Brute force the AES 192-bit encrypted AUTH_SESSKEY to determine the SHA-1 password hash
4. Once you have the SALT and the SHA-1 hash value, brute force the password.

Now 4. is moot, as it is the password that properly decrypted in 3.

SUNCOAST ORACLE USERS' GROUP

5 Hours? Really?

One AMD Radeon HD7970 GPU can average 8.2 billion password trys/sec

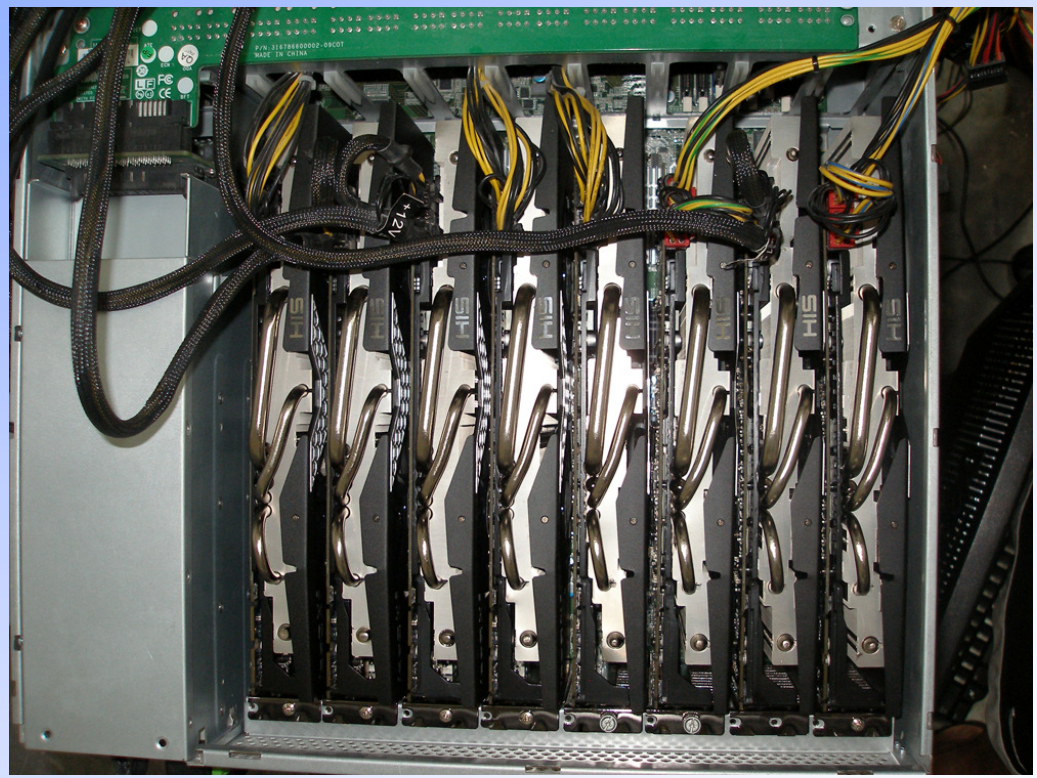
oclHashcat-plus can utilize multiple GPUs for exponential performance improvement

Rainbow tables can utilize pre-calculated values to cut even more time

SUNCOAST ORACLE USERS' GROUP

5 Hours? Really?

Here's an 8-Radeon card computer for about $12k that can brute force the entire 8-character namespace (upper/lower/digit/symbol) in 12 hours!!!



SUNCOAST ORACLE USERS' GROUP

Why is this so Insidious?

Wouldn't the account get locked due to too many failed login attempts?

SUNCOAST ORACLE USERS' GROUP

Why is this so Insideous?

Wouldn't the account get locked due to too many failed login attempts?

No!
You don't get locked because once you grab the AUTH_VRY_DATA and AUTH_SESSKEY, the rest is offline activity.

SUNCOAST ORACLE USERS' GROUP

How to Protect Against This?

How to Protect Against This?

Note that this is a flaw in O5LOGON protocol

O5LOGON came out with Oracle 11.1 (client and server)

SUNCOAST ORACLE USERS' GROUP

How to Protect Against This?

Upgrade to Oracle 12c

- or –

Go back to O3LOGON protocol

SUNCOAST ORACLE USERS' GROUP

How to Go Back to O3LOGON?

```
alter system set sec_case_sensitive_logon=FALSE scope=BOTH;

orapwd file=pwdSID.ora ignorecase=y

grant sysdba to USER1;

grant sysoper to USER2;
```

SUNCOAST ORACLE USERS' GROUP

So Now I'm Safe…Right?

So Now I'm Safe…Right?

WRONG!!!

David Litchfield, Accuvant Labs

Demonstrated a flaw that allows authenticated remote users to execute arbitrary SQL commands via vectors involving CREATE INDEX with a CTXSYS.CONTEXT INDEXTYPE and DBMS_STATS.GATHER_TABLE_STATS.

Affects: Oracle 8i thru 11.2.0.3

Fixed: July 2012 CPU

SUNCOAST ORACLE USERS' GROUP


SUNCOAST
ORACLE
USERS' GROUP

Stealth Password Cracking Vulnerability

Troy Ligon
tligon@soug.org

SUNCOAST ORACLE USERS' GROUP