

# Network Activity and Packet Analysis with Python

---

## Performing Packet Sniffing Actions with Scapy



**Sean Wilkins**

Network Engineer & Author

[swilkins@infodispersion.com](mailto:swilkins@infodispersion.com)

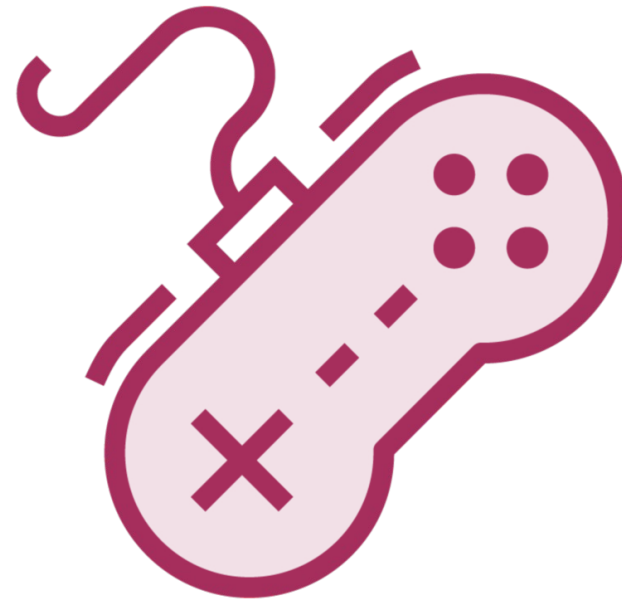
[www.infodispersion.com](http://www.infodispersion.com)



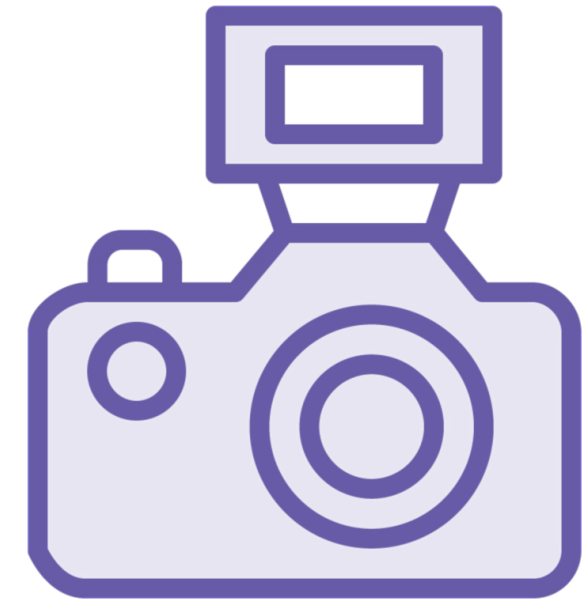
# Course and Module Overview



**Move from Socket to  
Scapy**



**Often engineers need  
tools with clear  
controls**



**This module focuses  
on packet captures**



# Overview



- **Setting the Stage**
- **Creating a Learning Environment**
- **Brief Introduction to Scapy**
- **Reviewing the Basics of Sniffing with Scapy**
- **Concepts Demonstration**



# Globomantics



**Let's set the conditions of our course's scenario**

**Globomantics is hiring you as one of their network security engineers**

**You are bringing experience with newer tools**

**Tools include: Python with Scapy**

**They have the ability to:**

- Be used at the CLI**
- Used in module form**



Scapy allows solution creation  
in multiple OS environments  
easier.





## Demonstration of use in a real environment

### Includes:

- Packet manipulation
- Port scanning & traceroute
- Identifying brute force attacks
- Connection hijacking
- Traffic replay

# Module Coverage Includes

**Sniffing: captures transmitted traffic**

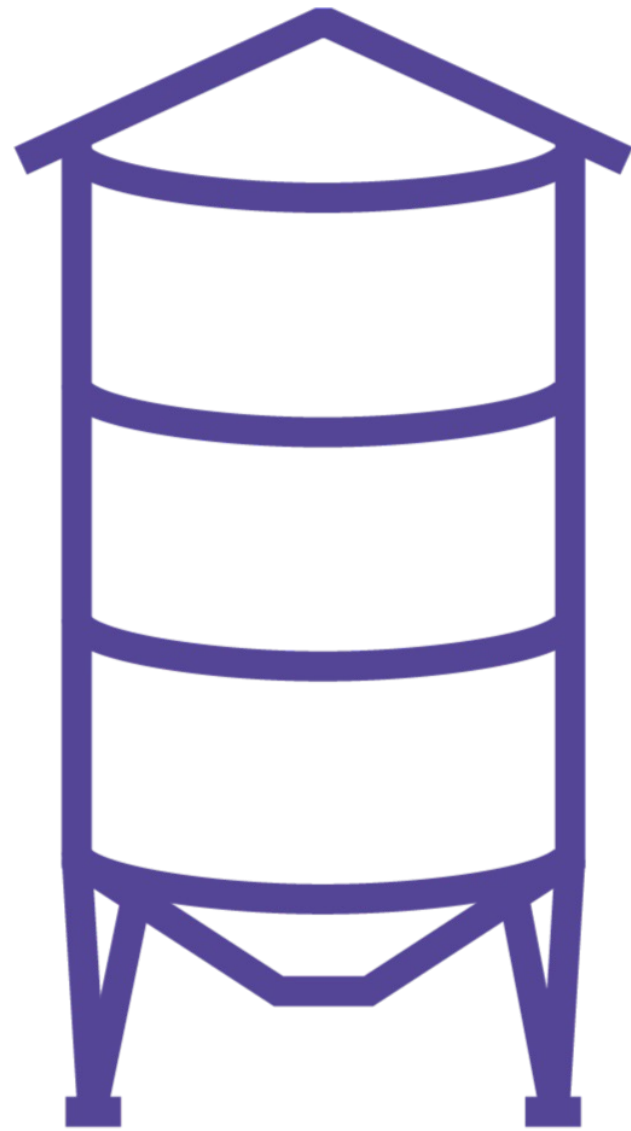
**Multiple options that provide sniffing**

**Python and Scapy**

**Can be done at the CLI**



# Module Coverage Includes How Sniffing:



**Collects information from a specific interface**

**Traffic can be filtered and parsed**

- Identifies potential attacks**
- Identifies source and destinations**

**Can be used at the CLI with Scapy**

**Can be used as a module to perform a specific action**



We will be using Kali Linux,  
Python 3.10.5, Scapy 2.4.5,  
iPython 8.4.0 and Microsoft  
Visual Studio Code





**Software is free**

**Easily setup in different environments**

**Kali Linux**

- Built on top of Debian Linux
- Focused on network security professionals
- Multiple tools come pre-installed
- Available at: <https://www.kali.org/get-kali/>



# Tasks to Perform

**Need to update  
Kali**

**sudo apt update**

**sudo apt full-  
upgrade  
commands**



Use the 'pip install  
scapy==2.4.5' command to  
update to Scapy





iPython is updated with the same command structure

Use 'pip install ipython==8.4.0'

Uses iPython as the CLI



# Installing Visual Studio Code

**`code.visualstudio.com`**

**Microsoft Visual Studio  
code repo**

**Standard Kali package  
managers can be used**

**Use 'apt install code'  
command**



# Scapy

**Built on Python**

**Created to make an easier to use  
framework**

**Can be used interactively and as a module**



# Scapy

**Has a large index  
of actions**

**Captures traffic**  
**Analyzes protocol  
fields**  
**Creates packets**

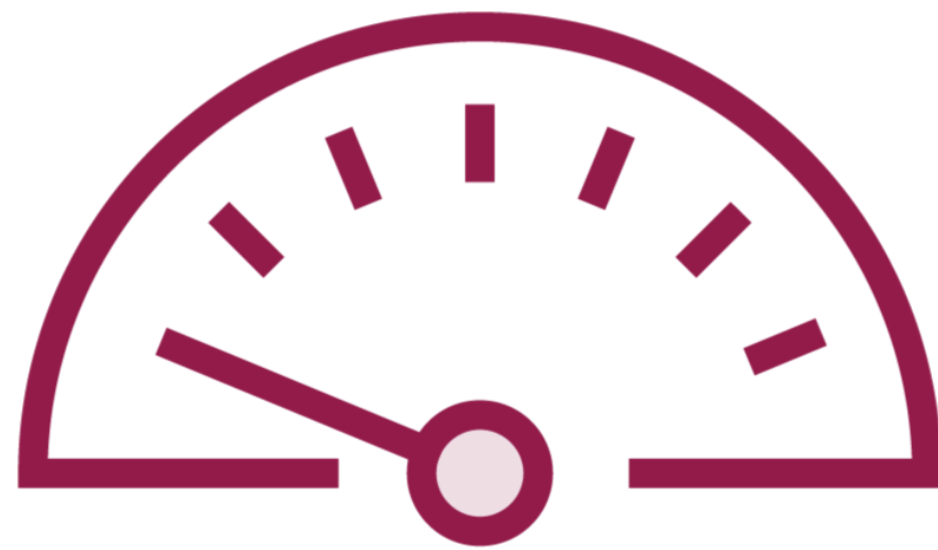
**Packets can be  
sent to different  
hosts**





Scapy was created to make life  
easier.





**Scapy is built on top of Python**

**Python has a lower level of performance**

**Don't use if high performance is needed**



Let's review the Scapy sniff  
command and how it can be  
used



# Scapy sniff command

**Can be used as a  
command style or  
as a function**

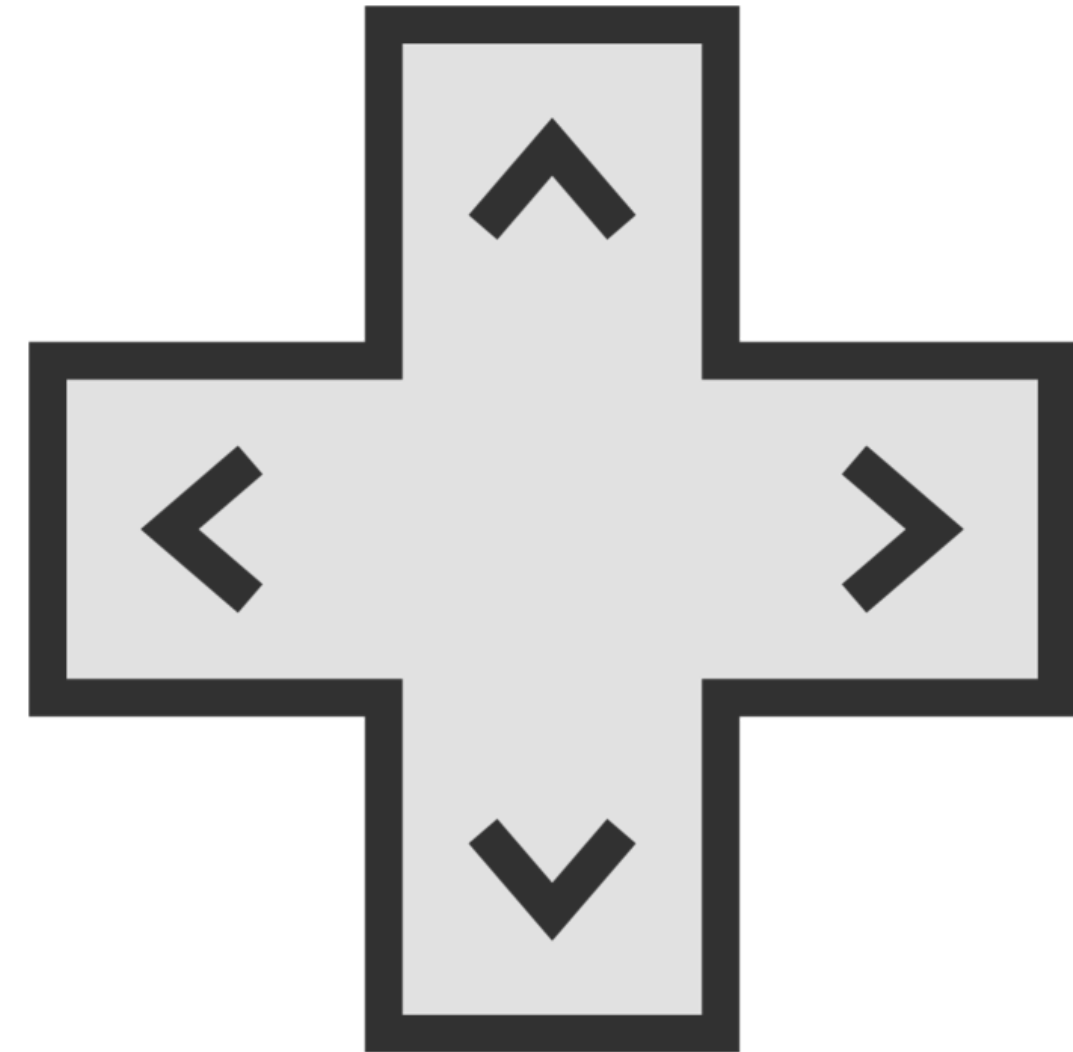
**Sniffs all traffic  
coming in or out**

**Will run in a loop  
until broken**



**Users want additional  
control over the packets**

**Multiple parameters  
available to alter behavior**



count, iface, timeout, filter,  
lfilter, prn, and offline are  
common parameters



count

**Dictates specific number of  
packets to capture**

**If not specified, will be  
unlimited**



# iface



**Specifies the interface(s) to capture from**

**Can specify multiple interfaces**

- Better to use multiple threads**

**There are issues but may be version specific**





# timeout

**Sets a specific  
time to capture a  
packet**

**Measure in  
seconds**

**Can limit the  
number of packets  
captured**



# filter and lfilter



**filter parameter uses Berkeley Packet Filters (BPF)**

**Uses keywords like:**

- src, dst, host, port**

**Operating system is filtering before Scapy does**

**PDF reference sheet: <https://idfo.in/3AZGahk>**

**Lfilter used when filter doesn't work or can't use BPF**

**Done within Scapy**



prn

**Specifies a function to run for each packet captured**

**Often matched with lambda command at CLI**

**If used in a python script, full functions are available**



# offline

**Sniff isn't limited to  
live packets**

**Can filter and  
parse on already  
captured files**

**Once imported,  
can perform  
multiple functions**



wrpcap

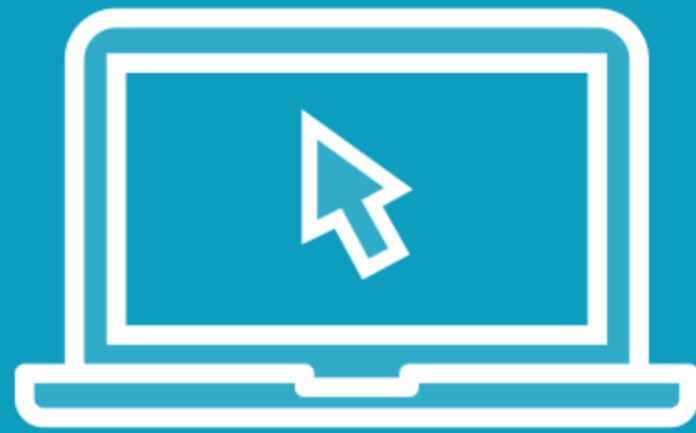
{y, x}

**Allows saving of live and offline loaded packets**

**Uses two variables**

- **String name to save into**
- **Variable that holds the packet list to be saved**

Demo



**Sniffing Using the count Parameter**

**Sniffing Using the timeout Parameter**

**Sniffing Using the iface and count Parameters**

**Sniffing Using the filter Parameter**

**Sniffing Using the lfilter Parameter**

**Sniffing Using the prn Parameter and lambda Function**

**Sniffing Using the offline Parameter**

**Saving Captures to File with wrpcap**



# Summary



- **Setting the Stage**
- **Creating a Learning Environment**
- **Brief Introduction to Scapy**
- **Reviewing the Basics of Sniffing with Scapy**
- **Concepts Demonstration**

