
Penetration Testing with
PYTHON
تست نفوذ پیشرفته با پایتون
محمد شریعتی مهر . بهمن ۹۵

نام کتاب : **تست نفوذ با پایتون**

منبع : **Learning Penetration Testing with Python**

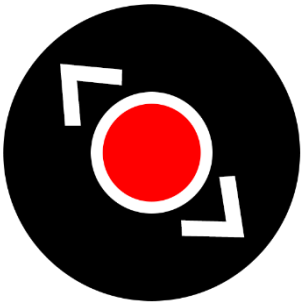
موضوع : **امنیت شبکه**

سطح آموزشی : **متوسط**

تاریخ انتشار : **بهمن 95**

تعداد صفحات : 440

نویسنده و مترجم : **محمد شریعتی مهر**



این کتاب به درد من می خوره ؟

اگر می خواهید چندتا اسکریپت آماده پیدا کنید و خیلی سریع به یک اسکریپت کیدی مبدل بشید این کتاب رو نخرید. اصلا به دردتون نمی خوره. اگر می خواهید نحوه توسعه ابزارهای تست نفوذ پایتون رو بر مبنای متدلوژی PTES یاد بگیرید این کتاب برای شما نوشته شده. در صورتیکه می خواهید نحوه توسعه درست ابزارهای تست نفوذ پایتون رو یاد بگیرید این کتاب برای شماست.

کتاب تست نفوذ پایتون چه ویژگی هایی داره؟

طراحی بر اساس متدلوژی PTES

آموزش نکات ضروری برای توسعه یک اسکریپت پایتون

آموزش توسعه اسکریپت ها در فازهای مختلف تست نفوذ

درج تصاویر اجرای فرایندها

شرح کامل جریان کدنویسی

معرفی کتابخانه های برتر پایتون در زمینه تست نفوذ

دارای فایل های تمرینی

پیش نیازها :

قبل از مطالعه این کتاب توصیه می شود در زمینه برنامه نویسی با زبان پایتون مهارت داشته باشید. همچنین توصیه می شود مقدمات موضوعات شبکه و امنیت شبکه و تست نفوذ را آموخته باشید. در صورتیکه در موارد یاد شده مهارت ندارید قادر به یادگیری کلی مباحث مطرح شده در کتاب خواهید بود ولی توصیه می شود پیش نیازها را رعایت کنید تا درک موضوعات برای شما ساده تر شود.

تغییرات :

کتابی که پیش روی شماست به صورت مستقیم از کتاب Learning Penetration Testing with Python نوشته کریستوفر دافی ترجمه شده است. فصل هشت از کتاب مرجع ترجمه به دلیل قدیمی بودن مباحث حذف شده و به جای آن مباحثی کاربردی تر از فصل های 8 و 10 کتاب پایتون کلاه مشکی جایگزین شده است. همچنین در فصل 6 کتاب نیز علاوه بر ترجمه کامل محتویات کتاب مرجع مباحثی کاربردی از کتاب تست نفوذ وب با پایتون اضافه شده است.

راهنمای مطالعه کتاب :

مباحثی که در کتاب مطرح می شود دارای فایل های تمرینی می باشند که این فایل ها ابزارها و اسکریپت های مورد نیاز بخش مربوطه هستند. توصیه می شود در حین مطالعه کتاب اسکریپت ها داخل ویرایشگر متنی باز کنید و موضوعات را دنبال کنید. صرف مطالعه کتاب نمی توان چیزی یاد گرفت. شروع به اسکریپت نویسی کنید و با وجود فایل های تمرینی سعی کنید خودتان یک بار مجدد اسکریپت ها را نوشته و عملکرد هر بخش از کد را تحلیل کنید.

خلاصه آنچه می خوانید :

متدولوژی PTES

ضروریات اسکریپت نویسی در پایتون

شناسایی اهداف و توسعه اسکریپت های اسکپی و انمپ با پایتون

توسعه اسکریپت برای حملات اعتبارنامه در پایتون

پایتون در بکارگیری سرویس ها

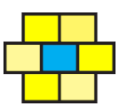
توسعه اسکریپت های تست نفوذ وب با پایتون

شکست پیرامون با پایتون

توسعه تروجان ویندوز با پایتون

توسعه اسکریپت های اتوماسیون گزارش دهی و وظایف با پایتون

شیوه نگارش ابزارهای استاندارد صنعتی



فصل یک : متدولوژی تست نفوذ

درک متدولوژی تست نفوذ

مروری بر تست نفوذ

آنچه تست نفوذ نیست

تعاملات مهندسی معکوس

هک کردن تست نفوذ نیست

متدولوژی های ارزیابی

استاندارد اجرای تست نفوذ

تعاملات قبل از شروع

تست نفوذ جعبه سفید

تست نفوذ جعبه خاکستری

تست نفوذ جعبه سیاه

جمع آوری اطلاعات

مدل سازی تهدید

آنالیز آسیب پذیری

بکارگیری

پس از بکارگیری

گزارش دهی

یک مثال از تست نفوذ

ابزارهای تست نفوذ

انمپ

متاسپلوییت

ویل

برپ سوییت

هایدرا

جان ریپر

شکستن پسوردهای ویندوز با ابزار جان ریپر

ابزار OCLHashcat

ابزار OphCrack

ابزارهای Mimikatz و Incognito

ابزار SMBexec

ابزار Cewl

ابزارهای Recon-NG و TheHarvester

ابزارهای FGDump و PWDump

ابزار نتکت

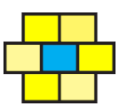
ابزارهای SysInternals

فصل دو : مقدمات اسکریپت نویسی در پایتون

مقدمات اسکریپت نویسی پایتون

تفاوت بین زبان های کامپایل شده و تفسیری

پایتون : بد و خوب



مفسر تعاملی پایتون در مقایسه با یک اسکریپت

متغیرهای محیطی و PATH

زبان های تایپ پویا

اولین اسکریپت پایتون

توسعه اسکریپت ها و شناسایی خطاها

کلمات رزرو شده, کلمات کلیدی و توابع درون ساخت

متغیرهای سراسری و محلی

فضای نام

ماژول ها و واردکردن آنها

فرمت دهی در پایتون

تورفتگی کد

متغیرهای پایتون

اشکال زدایی مقادیر متغیر

متغیر رشته ای

متغیر عددی

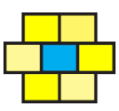
تبدیل متغیرهای رشته ای و اعداد

متغیر لیست

متغیر تاپل

متغیر دیکشنری

درک مقادیر پیش فرض و سازنده ها



ارسال یک متغیر به یک رشته

عملگرها

عملگرهای مقایسه

عملگرهای واگذاری

عملگرهای حسابی

عملگرهای منطقی و عضویت

عبارات ترکیبی

عبارت if

حلقه ها در پایتون

حلقه while

حلقه for

شرط break

هندلرهای شرطی

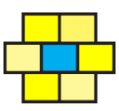
توابع

کاراکتر آکولاد در پایتون و دیگر زبان های برنامه نویسی

کامنت گذاری در پایتون

آرگومان ها و گزینه ها

اولین اسکریپت ارزیابی شما



فصل سه : شناسایی اهداف با انمپ, اسکپی و پایتون

شناسایی اهداف با Nmap , Scapy و پایتون

درک نحوه ارتباط سیستم ها

معماری فریم اترنت

لایه دو در شبکه های اترنت

معماری بسته آیپی

معماری هدر TCP

نحوه کار TCP

دست دهی سه مرحله ای TCP

معماری هدر UDP

نحوه کار UDP

مهارت انمپ

واردکردن محدوده هدف به انمپ

اجرای انواع گوناگون اسکن

اجرای اسکن اتصال کامل TCP

اجرای اسکن SYN

اجرای اسکن ACK

اجرای اسکن UDP

اجرای اسکن های ترکیبی TCP و UDP

عدم استفاده از اسکن سیستم عامل

انواع خروجی انمپ

خروجی Grepable

خروجی XML انمپ

موتور اسکریپت نویسی انمپ

کارآمدی در اسکن های انمپ

تشخیص جزئیات رابط شبکه با کتابخانه netifaces

کتابخانه انمپ برای پایتون

کتابخانه اسکپی برای پایتون

فصل چهار : حملات اعتبارنامه با پایتون

حملات اعتبارنامه با پایتون

انواع حملات اعتبارنامه

تعریف حملات آنلاین اعتبارنامه

تعریف حملات آفلاین اعتبارنامه

شناسایی هدف

ایجاد اسامی کاربری هدف

ایجاد و تایید اسامی کاربری به کمک Census

عملیات ایجاد اسامی کاربری

تست اسامی کاربری با SMTP VRFY

توسعه اسکریپت SMTP VRFY

فصل پنج : بکارگیری سرویس ها با پایتون

اکسپلوییت سرویس ها با پایتون

عصر تازه بکارگیری سرویس ها

درک زنجیره سازی اکسپلوییت ها

بررسی پسوردهای ضعیف, پیش فرض و شناخته شده

بدست آوردن دسترسی روت به سیستم

شکست هش های کپی شده لینوکس

تست همگام بودن اعتبارنامه های حساب کاربری

اتوماسیون بکارگیری با پایتون

فصل شش : تست نفوذ وب با پایتون

دسترسی به اپلیکیشن های وب با پایتون

شناسایی اپلیکیشن های زنده از طریق پورت های باز

شناسایی فایل ها و دایرکتوری های مخفی با پایتون

حملات اعتبارنامه با برپ سوییت

استفاده از twill برای مرور کد اپلیکیشن

چه زمانی باید از پایتون برای تست نفوذ وب استفاده کرد

چه زمانی از یک کتابخانه بخصوص استفاده کنیم

جمع آوری اطلاعات وب با رابط برنامه نویسی اپلیکیشن شودان

اسکریپت نویسی با استفاده از API جستجو گوگل پلاس

دانلود تصاویر پروفایل با API گوگل پلاس

ایجاد اسکرین شات از وبسایت ها با کتابخانه QtWebKit

فصل هفت : شکست پیرامون با پایتون

شکست پیرامون با پایتون

پیرامون جدید امنیتی

پروتکل های متن ساده

اپلیکیشن های وب

سرویس های ریموت رمزنگاری شده

شبکه های خصوصی مجازی

سرویس ایمیل

سرویس نام دامنه

سرویس UDP

پیوند بین حساب های کاربری و سرویس ها

شکست اینباکس با برپ سوییت

شناسایی مسیر حمله

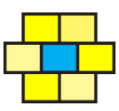
محدودیت های اسکن پیرامون

دانلود فایل های بک آپ از سرور TFTP

شناسایی نام فایل های بک آپ

شکست هش های Cisco MD5

کسب دسترسی از طریق وبسایت ها



حملات درج فایل

تایید یک آسیب پذیری RFI

بکارگیری میزبان ها از طریق RFI

فصل هشت : توسعه تروجان ویندوز با پایتون

توسعه تروجان ویندوز با پایتون

توسعه یک کیلاگر ساده با پایتون

اجرای اسکریپت کیلاگر

ایجاد اسکرین شات با پایتون

مانیتور فرایند با WMI

ایجاد exe با بسته Py2exe

تشخیص سندباکس (جعبه شنی) با پایتون

توسعه کیلاگر پیشرفته با پایتون

فصل نه : اتوماسیون گزارش ها و وظایف با پایتون

اتوماسیون گزارش ها و وظایف با پایتون

نحوه تجزیه و تحلیل فایل های XML برای گزارش ها

نحوه ایجاد یک کلاس در پایتون

توسعه یک اسکریپت پایتون برای تجزیه فایل XML انمپ

توسعه اسکریپت پایتون برای ایجاد صفحات گسترده اکسل

فصل ده : ماندگاری در ابزارها و اسکریپت های پایتون

ماندگاری در ابزارها و اسکریپت های پایتون

نحوه ایجاد لاگ با پایتون

تفاوت بین چندرشته ای و چند پردازشی

توسعه اسکریپت چندرشته ای در پایتون

توسعه اسکریپت چند پردازشی در پایتون

توسعه ابزارها با استاندارد صنعتی

فصل یازده : نصب پایتون و کتابخانه های مورد نیاز

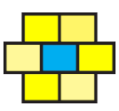
نصب پایتون بر روی ویندوز

تنظیم متغیرهای محیطی پایتون در ویندوز

نصب easy install در ویندوز

نصب pip در ویندوز

نصب کتابخانه های مورد نیاز در ویندوز



هشدار !

همه مطالب ارایه شده در این کتاب به منظور آموزش متخصصان امنیتی و ارتقا سطح امنیت شبکه های رایانه ای ارایه شده است !

لذا مسئولیت هر نوع استفاده نادرست و نفوذ غیرمجاز به سیستم های رایانه ای با شخص خاطی خواهد بود و این کتاب صرفا جنبه آموزشی دارد.

نفوذ غیر مجاز به شبکه های رایانه **جرم** محسوب می شود و به همین منظور در این کتاب از محیط تست ایزوله استفاده می شود.

بخشی هایی از کتاب به صورت تصادفی

تشخیص جزئیات رابط شبکه با کتابخانه netifaces

در فصل دوم اسکریپتی را به شما معرفی کردیم که با استفاده از آن می توانستید جزئیات یک رابط شبکه را پیدا کنید. این اسکریپت مستقل از هر نوع کتابخانه پایتون بود ولی نکته مهم این است که با استفاده از آن می توانستید بر اساس رابط شبکه تعیین شده جزئیات را استخراج کنید.

به جای این اسکریپت شما می توانید از کتابخانه netifaces به منظور جستجو در آدرس ها و کشف جزئیات استفاده کنید. اسکریپت یاد شده از تعدادی تابع به منظور انجام وظیفه تعیین شده استفاده می کند. این توابع عبارتند از :

get_networks , get_addresses , get_gateways , get_interfaces

اولین تابع یعنی get_interfaces همه رابط های شبکه مرتبط را برای سیستم تعیین شده پیدا می کند :

```
def get_interfaces():
interfaces = netifaces.interfaces()
return interfaces
```

تابع دوم get_gateways دروازه های پیش فرض (DG) را شناسایی کرده و آنها را در قالب یک دیکشنری پایتون بازگشت می دهد :

```
def get_gateways():
  gateway_dict = {}
  gws = netifaces.gateways()
  for gw in gws:
    try:
      gateway_iface = gws[gw][netifaces.AF_INET]
      gateway_ip, iface = gateway_iface[0], gateway_iface[1]
      gw_list =[gateway_ip, iface]
      gateway_dict[gw]=gw_list
    except:
      pass
  return gateway_dict
```

تابع سوم get_addresses آدرس های هر رابط شبکه را شناسایی کرده, که شامل مک آدرس, آدرس آیپی, آدرس برودکست, و ماسک شبکه می باشد. همه این جزئیات از طریق ارسال نام رابط شبکه که در مراحل قبل بدست آمده.....

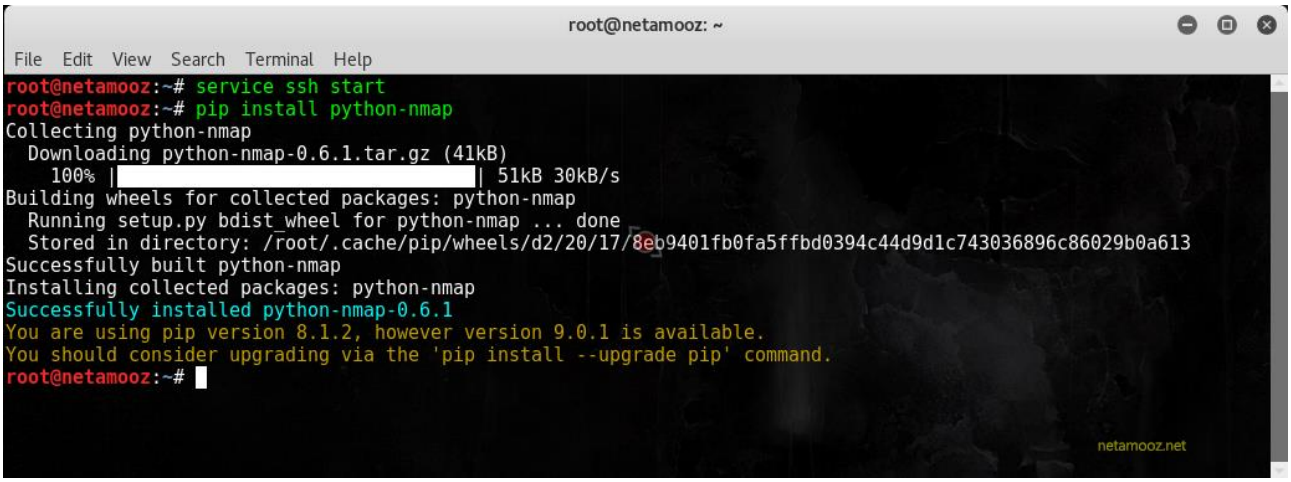
کتابخانه های انمپ برای پایتون

پایتون دارای کتابخانه هایی می باشد که به شما اجازه اجرای مستقیم اسکن های انمپ را می دهد. این کار را می توان از طریق شل تعاملی پایتون یا از طریق یک اسکریپت یا ایجاد ابزارهای حمله چندوجهی انجام داد.

برای مثال, می خواهیم از کتابخانه انمپ به منظور اسکن سیستم لوکال کالی لینوکس خودمان بر روی پورت SSH استفاده کنیم. قبل از شروع کار اطمینان حاصل کنید که سرویس ssh فعال می باشد. همچنین کتابخانه های انمپ را بر روی کالی نصب کنید :

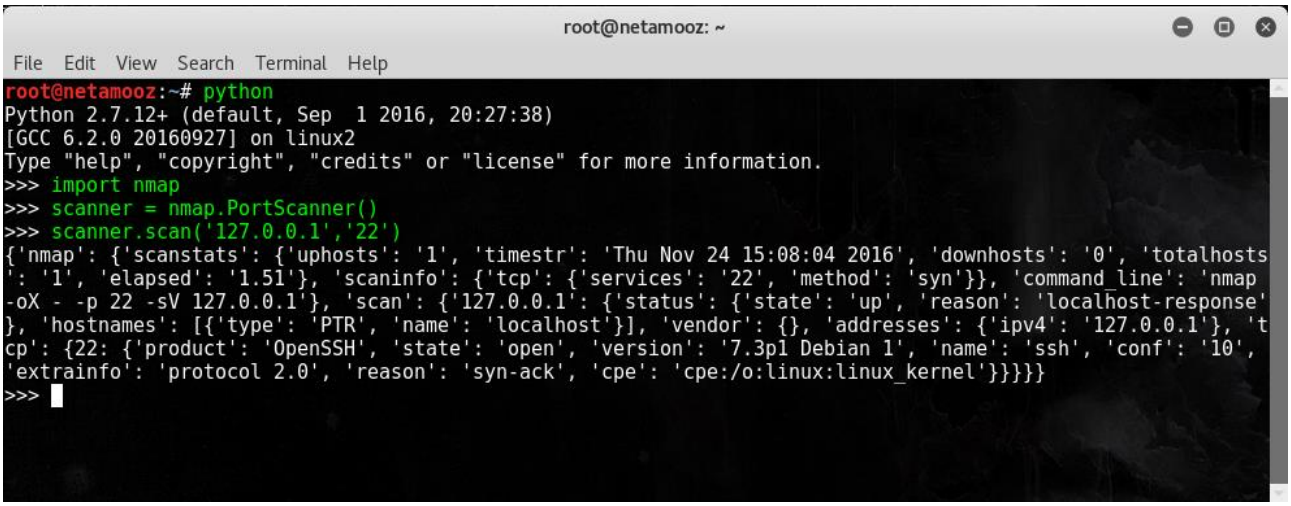
service ssh start

pip install python-nmap



اکنون می توانید یک اسکن را به صورت مستقیم و با استفاده از کتابخانه ها و وارد کردن آنها و اختصاص nmap.PortScanner() به یک متغیر اجرا کنید. در ادامه از این متغیر معرفی شده به منظور اجرای اسکن ها می توان استفاده کرد.

برای مثال می خواهیم یک اسکن را درون مفسر تعاملی پایتون انجام دهیم. مثال زیر اسکن پورت 22 کالی با استفاده از کتابخانه های انمپ را نشان می دهد :



همانطور که در تصویر بالا مشاهده می کنید, نتیجه یک دیکشنری از دیگر دیکشنری ها می باشد که هرکدام را در صورت نیاز می توان فراخوانی کرد. اجرای یک اسکن از طریق شل تعاملی کمی بیشتر کار می برد.

ما می توانیم اسکریپت هایی ایجاد کنیم که آرگومان هایی را برای اسکن میزبان هایی خاص به همراه پورت وارد شده اسکن کند. از آنجایی که در اسکریپت ما آرگومان های اسکن از خط فرمان دریافت می شود, ابتدا بایستی کتابخانه sys را وارد اسکریپت خود کنیم و به دلیل اینکه از طریق کتابخانه های انمپ اسکن را انجام می دهیم, بایستی کتابخانه nmap را نیز به اسکریپت خود وارد کنیم.

به یاد داشته باشید که در حین واردکردن کتابخانه ها درون اسکریپت خود از کنترل های شرطی استفاده کنید تا در صورت عدم نصب کتابخانه بر روی سیستم راهنمایی لازم به شخص برای نصب و وجود اشکال داده شود :

```
import sys
try:
    import nmap
```

```
except:
    sys.exit("[*] Ketabkhane Nmap ra nasb konid: pip install python-nmap")
```

زمانیکه کتابخانه ها وارد شد, برای اسکریپت بایستی آرگومان های مورد نیاز را طراحی کنیم. ما برای اسکریپت خود حداقل به دو آرگومان نیاز داریم. به این معنا که در صورتیکه کمتر از دو آرگومان وجود داشته باشد اسکریپت با شکست مواجه خواهد شد. به یاد دارید که آرگومان اول نام اسکریپت می باشد در نتیجه به سه آرگومان نیاز داریم به صورت زیر آرگومان ها را ایجاد می کنیم.

ایجاد اسامی کاربری

اولین گام در این فرایند, دانلود فایل صفحه گسترده اکسل می باشد. به این منظور به آدرس زیر رفته و فایل را دریافت کنید. به دلیل تحریم برای دریافت امکان دریافت با آیپی ایران وجود ندارد. فایل را می توانید از فایل های تمرینی دریافت کنید. برای دریافت از طریق کنسول از دستور wget استفاده کنید.

wget http://www2.census.gov/topics/genealogy/2000surnames/Top1000.xls

فایل را درون اکسل باز کنید و نحوه فرمت بندی آن را بررسی کرده تا بدانیم چگونه می توان اسکریپتی برای استخراج اطلاعات مورد نظر توسعه داد.

همانگونه که مشاهده می کنید یازده ستون در این صفحه گسترده وجود دارد. از این یازده ستون تنها دو مورد برای ما اهمیت دارد. یعنی ستون نام (name) و ستون رتبه (rank) ستون نام شامل نام خانوادگی که ما برای لیست خود استفاده خواهیم کرد و رتبه هم نرخ رخداد این اسامی در ایالات متحده امریکا می باشد. قبل از اینکه یک تابع برای تجزیه و

تحلیل این فایل ایجاد کنیم, بایستی چاره ای برای دریافت داده ها و ورودی آن به اسکریپت بیندیشیم.

کتابخانه argparser به شما اجازه می دهد تا گزینه ها و آرگومان های خط فرمان را به سرعت و موثر توسعه دهید. کتابخانه xlrd به منظور آنالیز فایل صفحه گسترده اکسل استفاده شده و کتابخانه string به منظور توسعه لیستی از کاراکترهای حروف الفبا استفاده می شود.

کتابخانه os نوع سیستم عاملی که اسکریپت از آنجا اجرا می شود تایید کرده تا فرمت نام فایل به صورت داخلی قابل رسیدگی باشد. در نهایت کتابخانه collections ابزاری برای سازماندهی داده های درون حافظه و گرفته شده از فایل صفحه گسترده اکسل فراهم می کند.

تنها کتابخانه ای که به صورت بومی در پایتون موجود نیست xlrd می باشد که قبل از استفاده بایستی با استفاده از دستور pip install xlrd نصب گردد.

```
#!/usr/bin/env python


import sys
from collections import namedtuple
import string
import argparse
import os
try:
    import xlrd
except:
    sys.exit("[*] lotfan Ketabkhane xlrd ra nasb konid : pip install xlrd")
```

توسعه اسکریپت SMTP VRFY

از آنجایی که متاسپلوییت و دیگر ابزارهای حمله به وقفه های زمانی برای ایجاد نشست و تاخیر در هر تلاش اهمیتی نمی دهند, برای تست SMTP VRFY نیاز به ایجاد یک اسکرپت سفارشی SMTP VRFY داریم. همانطور که قبلا هم گفتیم در 80 درصد اوقات ابزارهای موجود کافی هستند ولی برای یک تستر حرفه ای سازگاری با همه موقعیت ها یک نیاز اساسی محسوب می گردد.

کتابخانه هایی که استفاده کردیم را قبلا می شناسید. برای کنترل رابط شبکه از کتابخانه socket و برای کنترل وقفه های زمانی از کتابخانه time استفاده می کنیم.

```
#/usr/bin/env python
import socket, time, argparse, os, sys
```

تابع اول فایل ها را خوانده و آنها را به لیستی برای تست اسامی کاربری تبدیل می کند.

```
def read_file(filename):
    with open(filename) as file:
        lines = file.read().splitlines()
    return lines
```

.....

اتوماسیون بکارگیری با پایتون

این تمرین بکارگیری نسبتا ساده بود ولی می توان بخشی از آن را با استفاده از رویه فراخوان متاسپلوییت(MSFRPC), اتوماسیون کرد. اسکریپتی که خواهیم ساخت از کتابخانه nmap برای اسکن فعال پورت 445 بهره گرفته و سپس لیستی از اهداف را برای تست ایجاد کرده تا نام کاربری و رمزعبور از طریق آرگومان به اسکریپت برای تست ارسال شود.

اسکریپت از همان ماژولی که در بالا استفاده کردیم یعنی smb_enumusers_domain برای شناسایی جعبه تست بهره می گیرد. قبل از شروع شما بایستی کتابخانه Spiderlabs msfrpc را نصب کنید.

اسکریپتی که ایجاد می کنیم از کتابخانه netifaces برای شناسایی این موضوع که کدام آدرس آیپی به رابط شبکه شما تعلق دارد استفاده می کند. سپس پورت smb 44 را بر روی آدرس آیپی تعیین شده اسکن می کند. در ادامه آدرس آیپی متعلق به رابط شبکه شما را حذف کرده و در نهایت اعتبارنامه های تعیین شده را با استفاده از ماژول smb_enumusers_domain تست می کند.

در همین حین کاربر وارد شده به سیستم را تایید می کند. خروجی این اسکریپت علاوه بر پاسخ در لحظه ای که نمایش داده می شود, در دو فایل ایجاد می گردد. یک فایل لاگ که حاوی همه پاسخ ها دریافتی بوده و فایلی دیگر که آدرس های آیپی برای همه میزبان هایی که دارای سرویس SMB هستند.

شما می توانید فقط یک فایل ریسورس متاسپلوییت ایجاد کرده و از آن به جای اسکریپت استفاده کنید ولی زمانیکه قصد تست سازمانی با میلیون ها آدرس آیپی را دارید این فرایند دیگر قابل مدیریت نخواهد بود.

شبیه همه دیگر اسکریپت ها ابتدا بایستی کتابخانه های مورد نیاز را وارد کنیم. کتابخانه های مورد نیاز به همراه تست وجود آنها بر روی سیستم به صورت زیر انجام می پذیرد.

```
import os, argparse, sys, time
try:
    import msfrpc
except:
    sys.exit("[!] Lotfan Ketabkhane msfrpc ra az filehaye tamrini nasb konid")
try:
    import nmap
except:
    sys.exit("[!] Lotfan Ketabkhane Nmap ra nasb konid : pip install python-nmap")
try:
    import netifaces
except:
    sys.exit("[!] Lotfan Ketabkhane netifaces ra nasb konid : pip install netifaces")
```

جمع آوری اطلاعات وب با رابط برنامه نویسی اپلکیشن شودان

شودان یک موتور جستجو آسیب پذیری می باشد. شما با ارایه یک نام و یک آدرس آیپی و یا حتی یک شماره پورت می توانید اطلاعات مطابق موجود در پایگاه داده را بیرون بکشید. همین موضوع موجب شده تا شودان به یکی از موثرترین منابع در هوش اطلاعات زیرساخت ها مبدل شود.

در واقع شودان گوگل دیوایس های متصل به اینترنت می باشد. شودان دایما اینترنت را اسکن کرده و نتایج خود را درون یک پایگاه داده عمومی ذخیره می کند. گرچه این پایگاه داده از طریق وبسایت شودان قابل جستجو هست, ولی نتایج و سرویس های بدست آمده محدود هستند مگر اینکه از طریق API شودان یا همان رابط برنامه نویسی اپلیکیشن شودان اقدام نمایید.

در زمان نگارش این مطلب شما می توانید از طریق ایمیل خود عضو بخش توسعه دهندگان شودان شده و یک کلید API رایگان دریافت کنید. این موضوع شاید در آینده تغییر کند.

https://account.shodan.io/register

پس از عضویت یک ایمیل فعال سازی برای شما ارسال شده و پس فعال سازی کلید API در صفحه نمایش داده می شود. هرچند این کلید موقتی است و برای استفاده مداوم بایستی پلن های شودان را سفارش دهید.

در اینجا اسکریپتی را به شما معرفی می کنیم که با استفاده از API شروع به جستجو آسیب پذیری ها می کند. اسکریپت در بخش فایل های تمرینی کتاب موجود است. اسکریپت را درون ویرایشگر باز کنید تا با هم مرور کنیم.

اسکریپت ما به چه نحوی کار می کند؟

پس از وارد کردن کتابخانه های مورد نیاز ابتدا رشته های ایستا خود را درون کد تعیین می کنیم که شامل کلید API , آدرس سایت هدف و ... می باشد :

```
import shodan
import requests
SHODAN_API_KEY = "Klid API Shoda ra inja Vared Konid"
target = 'www.sitehadaf.com'
dnsResolve = 'https://api.shodan.io/dns/resolve?hostnames=' + target
+ '&key=' + SHODAN_API_KEY
```

گام بعدی ایجاد شی API می باشد :

```
api = shodan.Shodan(SHODAN_API_KEY)
```

توسعه یک کیلاگر ساده با پایتون

ضبط کلیدهای وارد شده توسط سیستم هدف یکی از قدیمی ترین تکنیک های استفاده شده توسط آزمونگرهای نفوذ می باشد. استفاده از این روش همچنان رایج است چرا که ضبط اطلاعات حیاتی وارد شده توسط کاربر, اطلاعاتی همچون اعتبارنامه ها و گفتگوها بسیار ارزشمند هستند.

کتابخانه Pyhook یک کتابخانه ارزشمند است که شما را قادر ساخته تا رخدادهای صفحه کلید را به دام اندازید.

این کتابخانه از تابع بومی ویندوز با نام SetWindowsHookEx کمک گرفته و به شما اجازه داده تا یک تابع تعریف شده توسط را نصب کرده و برای رخدادهای ویندوز در زمان نیاز آن را فراخوانی کنید. با رجیستر کردن یک Hook (قلاب) برای رخدادهای کیبورد, ما قادر خواهیم بود تا همه کلیدهای وارد شده توسط کاربر هدف را به دام اندازیم.

فراتر از همه ما می خواهیم بدانیم که کدام فرایند در حال اجرا این کلیدها را وارد می کند تا از این طریق قادر به تشخیص کلیدهای وارد شده نام کاربری و رمزها و دیگر اطلاعات ارزشمند باشیم.

کتابخانه Pyhook بیشتر این وظایف را در سطح پایین برنامه نویسی سیستم برای ما انجام می دهد. وظیفه ما ایجاد منطق اصلی ایجاد کیلاگر است. فایل تمرینی Keylogger.py را باز کنید و بخش ابتدایی کد را بررسی نمایید :

```
from ctypes import *
import pythoncom
import pyHook
import win32clipboard

user32   = windll.user32
kernel32 = windll.kernel32
psapi    = windll.psapi
current_window = None
```

ایجاد یک اسکریپت پایتون برای تجزیه فایل XML انمپ

کلاسی که برای فایل مثال تعریف خواهیم کرد بسیار ساده است. کلاس ما سه تابع خواهد داشت. یک تابع سازنده __init__ , یک تابع که داده های ارسالی را پردازش می کند و درنهایت یک تابع که که داده های پردازش شده را بازگشت می دهد.

ما کلاس را به نحوی پیاده سازی می کنیم که فایل XML انمپ و سطح درازنویسی را بپذیرد و در صورتیکه هیچ کدام از این ارسال نشدند مقدار پیش فرض 0 را در نظر بگیرید. تعریف کلاس در زیر و تابع __init__ را بررسی کنید :

```
class Nmap_parser:
    def __init__(self, nmap_xml, verbose=0):
        self.nmap_xml = nmap_xml
        self.verbose = verbose
        self.hosts = {}
        try:
            self.run()
        except Exception, e:
            print("[!] There was an error %s") % (str(e))
            sys.exit(1)
```

اکنون اقدام به تعریف تابعی که کار اصلی این کلاس را به انجام می رساند کرده. همانگونه که مطلع هستید, ما نیاز به ارسال هیچ متغیری به تابع نداریم, چرا که آنها درون self قرار گرفته اند.

در اسکریپت های بزرگتر ابتدای هر تابع وظیفه انجامی را با کامنت توضیح می دهند تا در مرور کد در آینده نیاز به خواندن صدها خط کد نباشد.

تابع run تست می کند که آیا قادر به بازکردن فایل XML می باشد یا خیر و سپس آن را با استفاده از تابع parse از کتابخانه etree فایل را درون یک متغیر بارگذاری می کنیم. سپس تابع متغیرهای ضروری اولیه را تعریف کرده و .....