

HACKING

PERIPHERAL PROTECTION

IT SECURITY MAGAZINE

VOL.17, NO.05

ETHICAL HACKING STARTER KIT

**OUR JOURNEY INTO BUILDING
AN UNDETECTABLE PYTHON RANSOMWARE**

**RASPBERRY-ARDUINO BLUETOOTH COMMUNICATION
AND BLUETOOTH ATTACKS**

EVIL TWIN ATTACK

XSS FLAWS AND FOCUS ON BLIND XSS

AND MORE...

HAKIN9

TEAM

Editor-in-Chief

Joanna Kretowicz

joanna.kretowicz@eforensicsmag.com

Managing Editor

Marta Sienicka

sienicka.marta@hakin9.com

Editors:

Marta Sienicka

sienicka.marta@hakin9.com

Marta Strzelec

marta.strzelec@eforensicsmag.com

Bartek Adach

bartek.adach@pentestmag.com

Michalina Szpyrka

michalina.szpyrka@eforensicsmag.com

Proofreader:

Lee McKenzie

Senior Consultant/Publisher:

Paweł Marciniak

CEO:

Joanna Kretowicz

joanna.kretowicz@eforensicsmag.com

Marketing Director:

Joanna Kretowicz

joanna.kretowicz@eforensicsmag.com

DTP

Marta Sienicka

sienicka.marta@hakin9.com

Cover Design

Hiep Nguyen Duc

Joanna Kretowicz

Publisher

Hakin9 Media Sp. z o.o.

02-676 Warszawa

ul. Bielawska 6/19

Phone: 1 917 338 3631

www.hakin9.org

BETATESTERS &

PROOFREADERS

Lee McKenzie

Hammad Arshed

Avi Benchimol

Amit Chugh

Kevin Goosie

Craig Thornton

Paul Mellen

Daniel W. Dieterle

Steve Hodge

Shanika B

David Molik

Girshel Chokhanelidze

Sumit Kumar Soni

Jaimandeep Singh

Elia Pinto

Danish Zia

Humberto A. Sanchez II

Jordan M. Bonagura

All trademarks, trade names, or logos mentioned or used are the property of their respective owners. The techniques described in our articles may only be used in private, local networks. The editors hold no responsibility for misuse of the presented techniques or consequent data loss.

DEAR READERS,

Starting a career as an ethical hacker can be a daunting task. The number of resources available on the internet is endless and it can be hard to find the best tutorial. To make things easier for you we prepared a special ethical hacking starter kit that contains ten tutorials about various parts of offensive security. Let's see what's inside!

We start with *Cross-Site Scripting (XSS) flaws and focus on Blind XSS*, where you will see how one of the most popular web app attacks works. What's more, the authors created their own safe vulnerable application environment to show progress.

Moving next, we have *Raspberry-Arduino Bluetooth communication and Bluetooth Attacks*, the authors focused on the offensive side showing how attacks such as Man-in-the-Middle, sniffing, and replay can affect BLE. After learning about vulnerabilities in the second part of the article, you will learn how to secure those devices.

Securing A Smart Home will show you how to hack wi-fi to breach security in your home. You will first study the risks of having a smart house but also its advantages, then you will see how to avoid being attacked. Finally, you will read about potential prevention methods.

The article about the *Evil Twin Attack* is a must-read. You will learn how to trick users into connecting to a fake Wi-Fi access point that mimics a legitimate network. Once a user is connected to an "evil twin" network, you as a hacker can access everything, from their network traffic to private login credentials.

Our journey into building undetectable Python ransomware is a special article where you will see what kind of methods can be used to create this dangerous type of malware. Of course, everything is done in a safe environment, and for educational purposes only.

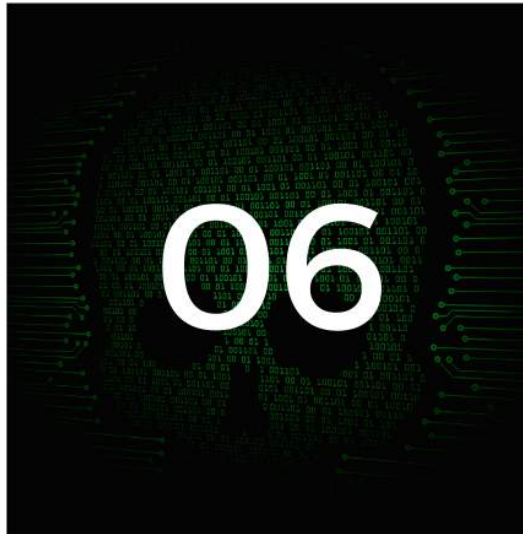
There's plenty more in this edition! You will learn what *AhMyth (Android Remote Administration Tool)* is and how to use it. For OSINT fans we have an amazing *Introduction to SpiderFoot*. If you are interested in password cracking we highly recommend *How To Install, Configure, And Use Hashcat In An Azure VM Step-By-Step*. You will have a chance to learn more about *CeWL*, a tool that takes the input of a URL and, via automated methods, recursively parses and collects specific parts of web pages that it finds. And finally, we have an amazing write-up about a *CTF Wargame - Mr-Robot*.

As you can see, this month's issue is packed with guides and tools that may come in handy! We hope you'll find something that suits your needs best! We would also like to send gratitude to our contributors, reviewers, and proofreaders, who helped us create this issue!

Stay safe and enjoy!

Hakin9 Editorial Team

Contents



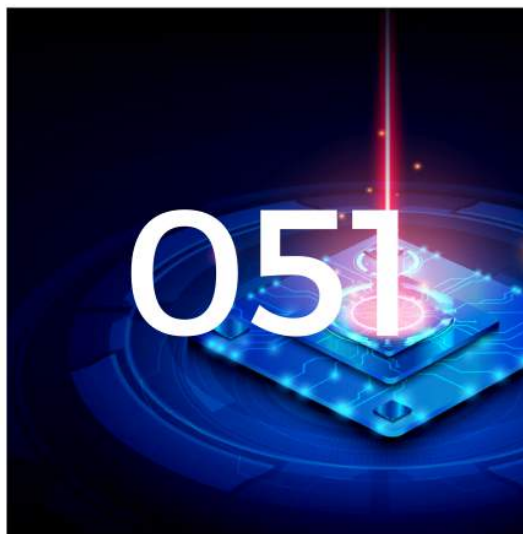
Cross-Site Scripting (XSS) Flaws and Focus On Blind XSS

by Roberto CHEMAMA,
Antoine PLANQUE



Raspberry-Arduino Bluetooth Communication and Bluetooth Attacks

by Olivier Pauchont,
Robin Hosking



Securing a Smart Home

by MOHAMMADY Arvin,
OGANEZOV Alexandre



Introduction to SpiderFoot

by Jeff Minataka

Contents



080

Evil Twin Attack

by Maya Sandra AIT YAHIA,
Elisa CAZERES



093

Our Journey Into Building an Undetectable Python Ransomware

by Barrault Victor,
Ornella Fabi



106

There's a Ghost in My Machine - The AhMyth Ghost

by Byron Gorman



118

How To Install, Configure, And Use Hashcat In An Azure VM Step-By-Step

by José Pablo

Contents



The CeWL Way To Scrape Webpages

by Gregory Haapaoja



CTF Wargame - Mr-Robot

by Gabriel Lazo Canazas

CROSS-SITE SCRIPTING (XSS) FLAWS AND FOCUS ON BLIND XSS



ROBERTO CHEMAMA

As soon as a new subject presents itself to him, Roberto dives in to understand its workings. He is passionate about cybersecurity and intends to constantly improve his skills in this field.

roberto.chemama@etu.u-paris.fr

<https://www.linkedin.com/in/roberto-chemama/>



ANTOINE PLANQUE

Antoine is curious to constantly learn new things about cybersecurity topics. He has a strong desire to practice more CTF and is considering certifications like the OSCP.

antoine.planque@etu.u-paris.fr

<https://www.linkedin.com/in/antoine-planque-a7387320b/>

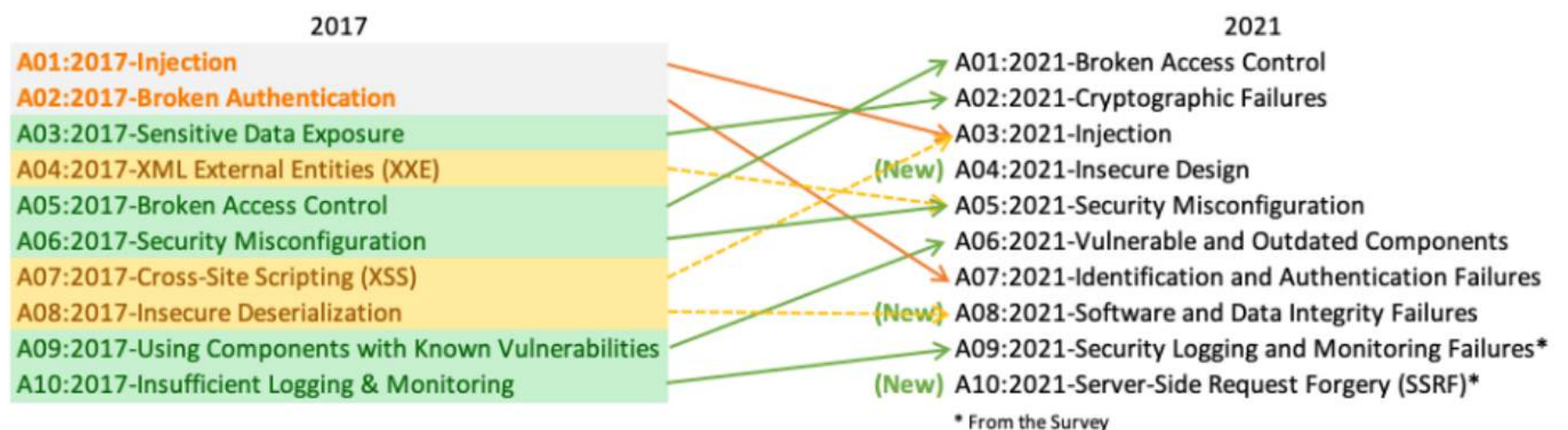


INTRODUCTION TO XSS ATTACKS

Applications that run in a Web environment involve many technologies both on the client side (type of browser, browser version, etc.) and on the server side (type of server, database management system, etc.) and therefore generate a large attack surface.

Indeed, the OWASP, an association that works on web application security, has published a list of ten major vulnerabilities that can be found in this environment. In 2021, this list was updated for the third time to reveal a categorization of these flaws.

The one we are interested in in this article is the XSS flaw that goes from the A07:2017-Cross-Site Scripting (XSS) nomenclature to A03:2021-Injection.



OWASP TOP 10 update in 2021

It is not by chance that the XSS flaw is now categorized as an injection flaw because it consists of injecting code fragments (HTML, JavaScript) into a Web page that interprets it as something legitimate.

There are three main types of XSS flaws: Reflected XSS, DOM XSS, Stored XSS.

REFLECTED XSS

Context of the flaw

This flaw concerns the parameters that are passed in the URL request. We often find it in phishing mails, leading the user to an illegitimate page to retrieve his personal information, for example.

Overview of the flaw

There is no interaction with the web server and the payload will not be stored in the database.

Impacts of the flaw

The two major impacts of this vulnerability are the redirection of the user to another web page and the ability to modify the interactions between users within the application.

DOM XSS**Context of the flaw**

This flaw concerns the parameters that are modified in the source code of the web page.

Overview of the flaw

There is no interaction with the web server and the payload will not be stored in the database since everything happens in the browser.

Impacts of the flaw

It will be possible for the attacker to modify the DOM (Document Object Model), which defines the structure of the pages interpreted by the browser and therefore the visual appearance.

Elements such as buttons or forms can also be impacted by this vulnerability, thus modifying the behavior of the latter.

STORED XSS**Context of the flaw**

In general, we find this flaw on the following sites:

- ➡ Chat application
- ➡ Forum
- ➡ Comment areas

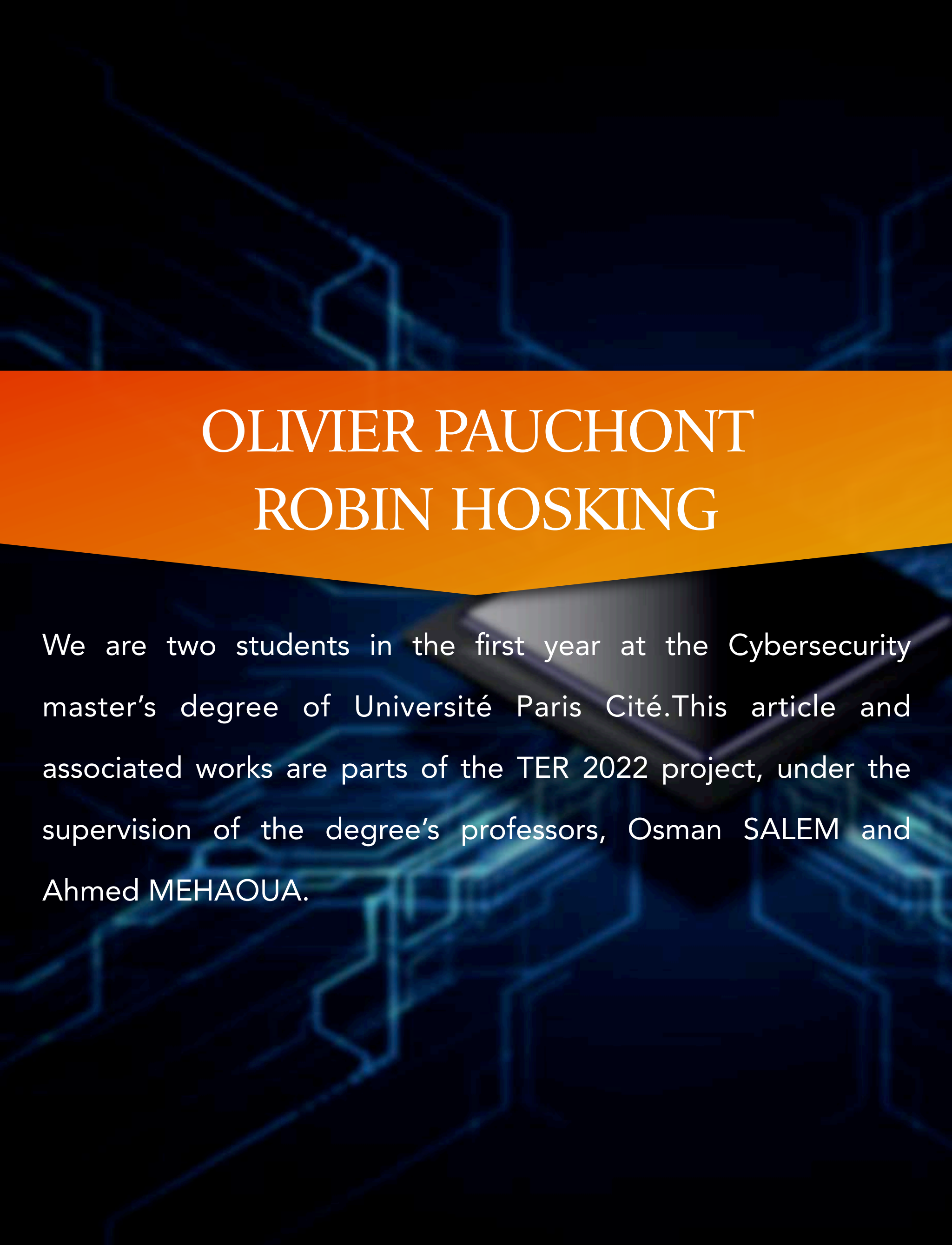
Overview of the flaw

There is an interaction with the web server and as the name suggests, it is persistent, stored in a database.

Impacts of the flaw

RASPBERRY- ARDUINO BLUETOOTH COMMUNICATION AND BLUETOOTH ATTACKS





OLIVIER PAUCHONT ROBIN HOSKING

We are two students in the first year at the Cybersecurity master's degree of Université Paris Cité. This article and associated works are parts of the TER 2022 project, under the supervision of the degree's professors, Osman SALEM and Ahmed MEHAOUA.

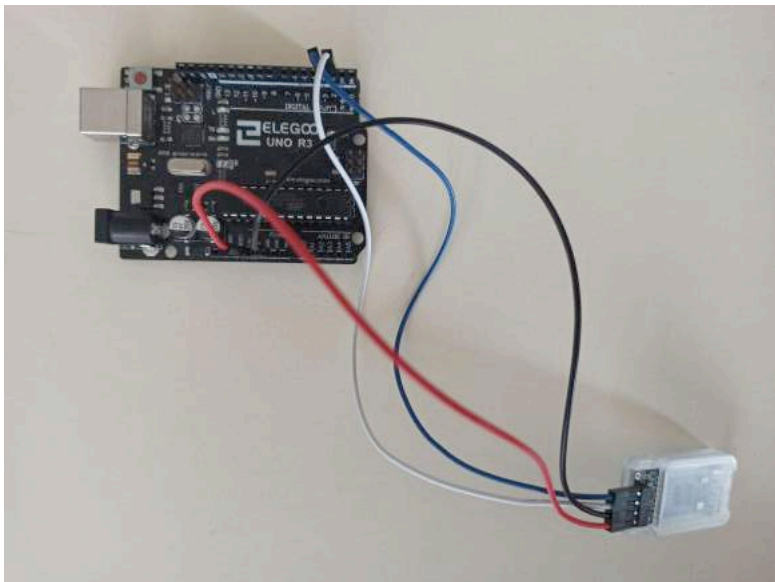
INTRODUCTION

Bluetooth Low Energy (BLE), also called Smart Bluetooth, is a well-established technology in the wireless devices market, used by many products and very convenient for users. However, by its nature and its many uses, it may be vulnerable to attacks such as Man in the Middle, sniffing, and replay. Our goal in this article is to study the BLE, reproduce some of these attacks to illustrate the limits of BLE and try to get some interesting reflection about its use and the possible improvements in its security. We will use two different tools: the open source BtleJuice framework developed by Digital Security and a Bluetooth sniffer. The first part of the article is dedicated to the establishment of Bluetooth Low Energy communication, and in the second part we will discuss possible attacks.

To do so, we used a Raspberry Pi and an Arduino Uno microcontroller to exchange data via a BLE connection. Handling is not particularly complicated and does not need advanced knowledge in Unix systems.

Equipment used:

- ➔ Raspberry Pi 4B
- ➔ Arduino Uno
- ➔ Bluetooth Dongle: BLE 4.0 DSD-TECH HM10
- ➔ Sniffer BLE Nordic Semiconductor nRF52840
- ➔ Application nRF Connect for Mobile (v2.5.1)



Bluetooth Dongle HM-10 connected to the Arduino Uno

ARDUINO PROGRAM

We need a small program to configure the connection between the Arduino microcontroller and the Bluetooth dongle.

```
#include <SoftwareSerial.h>
SoftwareSerial HM10(2, 3); // RX = 2, TX = 3

#define baudRate 9600
#define pinLED 13

char appData;
String inData = "";

void setup(){
  Serial.begin(baudRate);
  Serial.println("HM10 serial started at 9600");
  HM10.begin(baudRate); // set HM10 serial at 9600 baud rate
  pinMode(pinLED, OUTPUT); // onboard LED
  digitalWrite(pinLED, LOW); // switch OFF LED
}

void loop()
{
  HM10.listen(); // listen the HM10 port
  while (HM10.available() > 0) { // if HM10 sends something then read
    appData = HM10.read();
    inData = String(appData); // save the data in string format
    Serial.write(appData);
  }

  if (Serial.available()) { // Read user input if available.
    delay(10);
    HM10.write(Serial.read());
  }
}
```

The Arduino code to manage the Bluetooth dongle

However, it was necessary to consider the value of the transmission speed of the HM-10 module, and the TX and RX digital ports.

In our example, the RX digital port will occupy slot 2 and the TX port will occupy slot 3.

To know the serial transmission speed of the HM-10 module, you have to look at the documentation of it.

8 AT Commands

Factory default setting:

Name: HMSoft; Baud: 9600, N, 8, 1; Pin code: 000000; Peripheral Role;

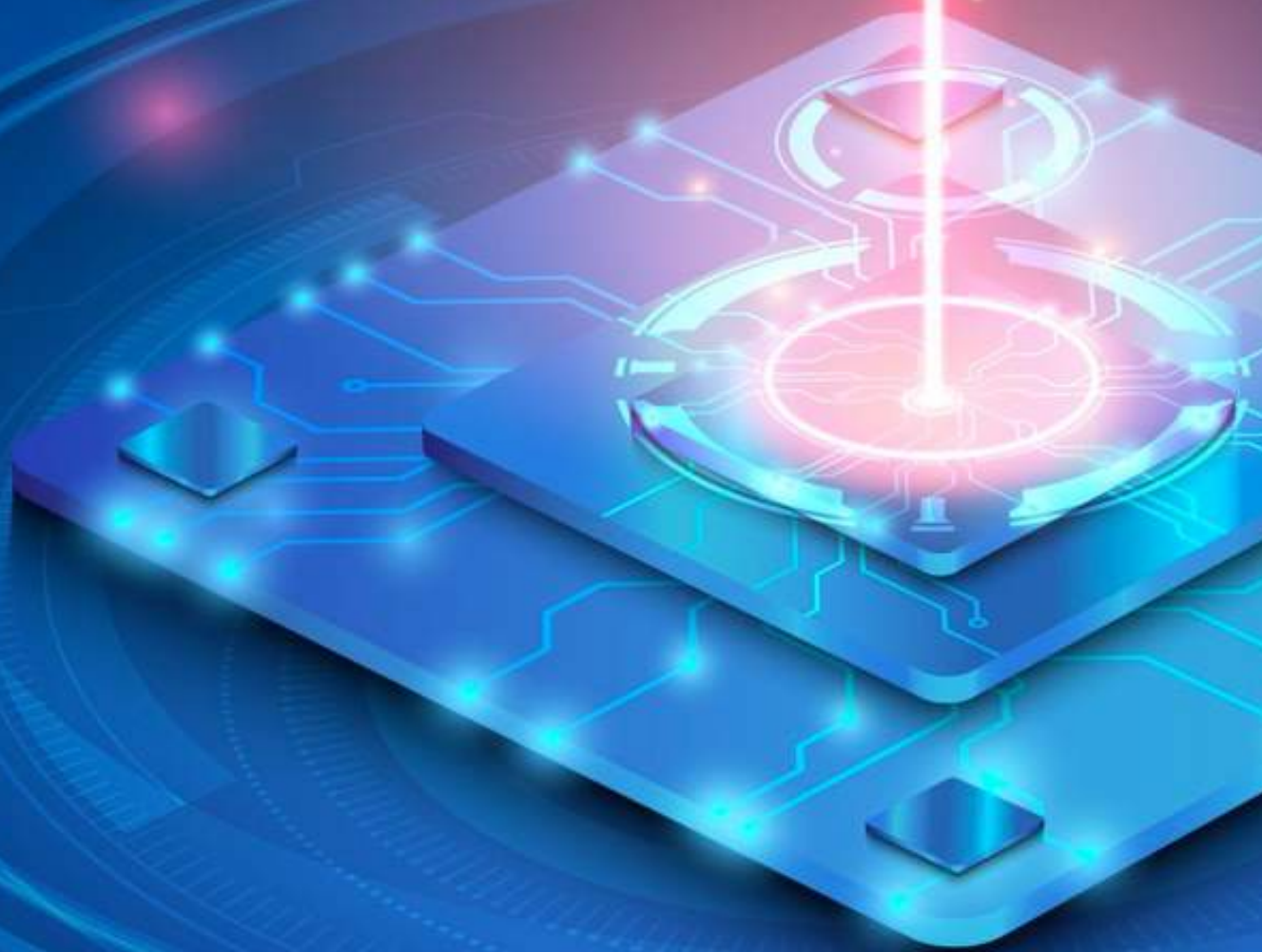
transmit mode.

AT Command format:

Uppercase AT command format. string format, without any other symbol. (e.g.

\r or \n).

SECURING A SMART HOME





MOHAMMADY ARVIN OGANEZOV ALEXANDRE

We are both first year master's degree students at Université Paris Cité in France. We both study cybersecurity. The idea of our project came from the fact that we wanted the Raspberry Pi to be the node of something but also to work on something that could potentially be useful.

INTRODUCTION

In a world where technological advances are increasingly numerous, there is an emergence of what is called "smart homes". It is a term that is beginning to be known by the public through its media coverage as in series and movies.

Perhaps some of you do not know what a smart home is. To begin with, smart houses are part of the home automation field. These are homes where the equipment is automated. It means that all actions can be programmed remotely thanks to your smartphone or tablet. For example, imagine that you come back home, tired after a long day of work and then when you arrive, the lights are already turned on, as well as the coffee machine and everything you needed. This is a considerable savings of time and energy for the owner.

However, the automation of a multitude of devices in a house inevitably involves risks, and that is precisely what this article is about. We will first study the risks that a smart house can set forth but also its advantages, then we will see how to avoid being attacked and finally, we will propose a solution to secure a smart house based on the study of human behaviour.

WHAT ARE THE ADVANTAGES AND RISKS OF OWNING A SMART HOME?

Advantages

Smart homes have many advantages, in fact, a smart home greatly improves the comfort of its owner. The automation of a home's equipment and facilities allows the owner to make his life easier. Take for example a person who is at work. This person knows how long it will take to get home, so he can already start the appliances to start cooking the meal, heating the house, etc., before he gets home so that he does not waste time. He can also decide to turn off a piece of equipment at any time because he may have forgotten to do so, which could have dramatic consequences.

A smart house allows you to save money. As we have seen before, we can activate or deactivate equipment remotely at any time, whether we are home or not. Therefore, it is a savings of energy but also of money. Since we can regulate our energy consumption, there is also an environmental impact.

Another advantage is that a smart home is more secure. Indeed, the risk of breaking and entering is almost null because we have several levels of security. For example, a door can be configured so that it opens only when the owner is facing the door. We can also program alarms and/or surveillance cameras to control them from afar, to be alerted in case of intrusion and to contact the security services quickly.

Nevertheless, this security that is sold to us is not totally infallible and, on the contrary, presents many risks.

Risks

A smart house can be subject to bugs, such as the fact that the smartphone, tablet, or any equipment used to control the house stops working. Then, how can we make the house obey? You will have to switch back to manual and wait for the system to work again or call a technician.

In addition to that, a smart house is dependent on the Internet because the equipment works and communicates with each other through the Internet. All it takes is for the Wi-Fi to stop working or for there to be a general outage at the Internet service provider for the house to stop responding for an indefinite period, and in this case, what would be the impact on the occupants?

Hacking the Wi-Fi

Let's see how you can hack the Wi-Fi of someone to show you how vulnerable a smart home can be.

First, we use the **iwconfig** command to check the wireless network configuration. We see that the Wi-Fi antenna is in “managed mode”, which is the default mode that allows it to connect to the Wi-Fi hotspot.

```
(kali@kali)-[~]
$ iwconfig
lo        no wireless extensions.

eth0      no wireless extensions.

wlan0     IEEE 802.11  ESSID:off/any
          Mode:Managed Access Point: Not-Associated Tx-Power=20 dBm
          Retry short limit:7 RTS thr:off Fragment thr:off
          Power Management:off
```

But we need it to be in “monitor mode” to capture all the Wi-Fi frames. To do so, we use the **sudo airmon-ng start wlan0** command.

```
(kali@kali)-[~]
$ sudo airmon-ng start wlan0

PHY      Interface  Driver      Chipset
phy0     wlan0      ath9k_htc   Qualcomm Atheros Communications AR9271 802.11n
          (mac80211 monitor mode vif enabled for [phy0]wlan0 on [phy0]wlan0mon)
          (mac80211 station mode vif disabled for [phy0]wlan0)
```

Here we see that the mode has effectively been changed to “monitor”:

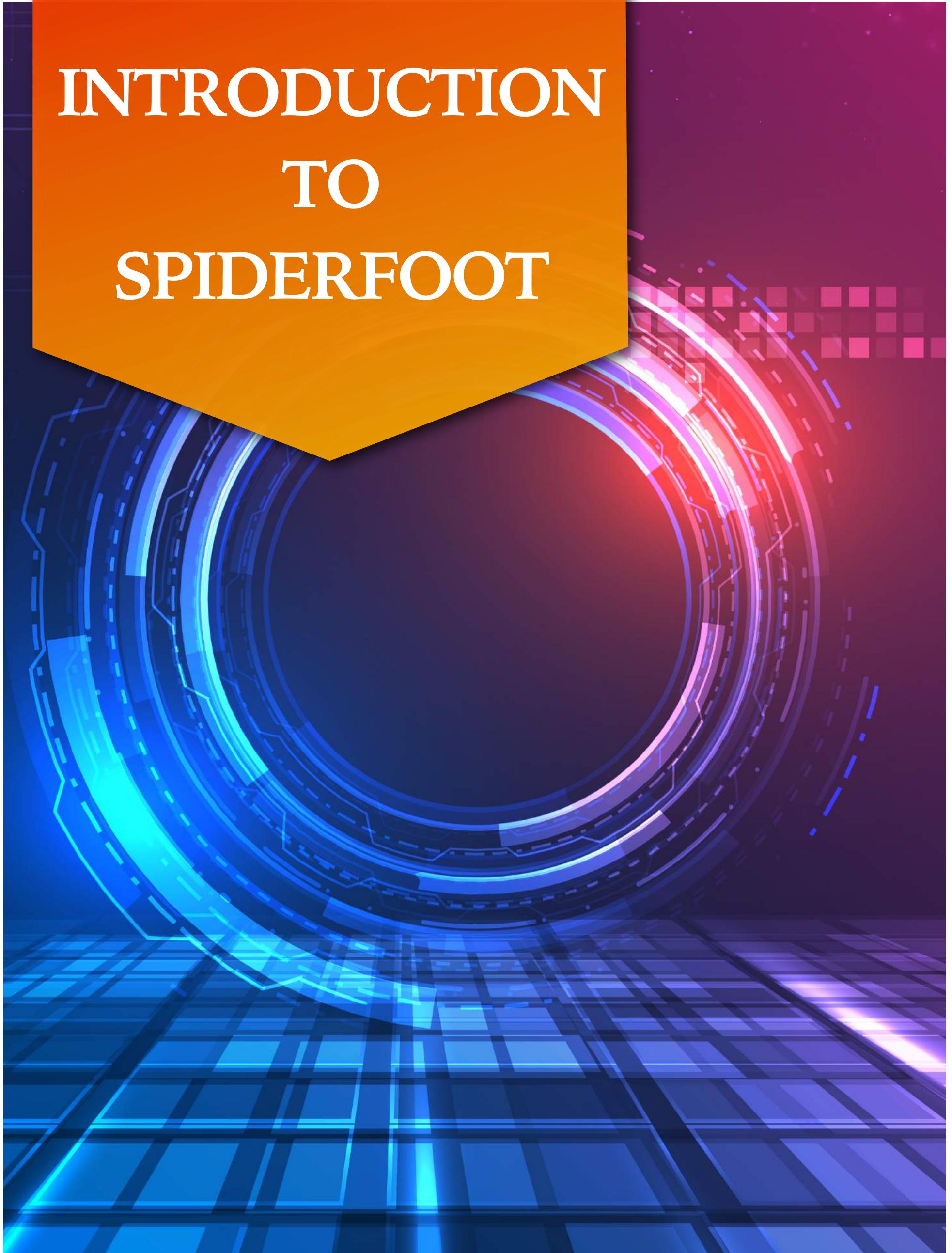
```
(kali@kali)-[~]
$ iwconfig
lo        no wireless extensions.

eth0      no wireless extensions.

wlan0mon  IEEE 802.11  Mode:Monitor Frequency:2.457 GHz Tx-Power=20 dBm
          Retry short limit:7 RTS thr:off Fragment thr:off
          Power Management:off
```

Now that this is done, we can choose which Wi-Fi we want to hack by using the command **sudo airodump-ng wlan0mon**:

INTRODUCTION TO SPIDERFOOT





JEFF MINATAKA

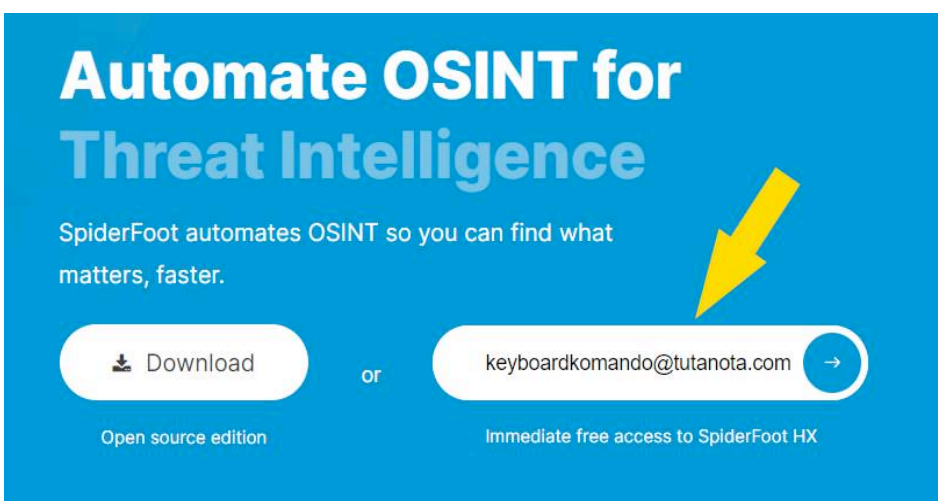
MCSI OSINT Practitioner, trained in CEH8 and CEH9, CISP, Metasploit certified, Accredited Configuration Engineer (ACE), and CWA certified. Over 20 years of experience in the IT industry. Online instructor for OSINT, ethical hacking, and network security. Has contracted courses for EC-Council and has written articles for Hakin9 and eForensics magazine.

keyboardkomando@protonmail.com

In this article, we will be taking a look at the tool SpiderFoot. This comes in two versions, an open-source edition on Github and also an online version. For this article, we will be using the online (Cloud) version as it contains more features and also simplifies getting started across whatever operating system that you may be using. As long as you have an internet connection and a browser, you should be able to use this tool.



So what is SpiderFoot? SpiderFoot is an automated OSINT (open-source intelligence) tool designed for asset discovery, threat intelligence, security assessment, and more. This is a great tool for a variety of situations. Given that this is an automated tool, this can save us a great deal of time finding new assets or verifying what we have already found.



Setting up your account is easy; head to the website and enter an email to set up your account (the link is in the **On the Web** section).

The free (Hobby) account is limited to three scans per month, 10 unique targets per scan, 1-hour scan duration, one user account, and 30-day retention. This will be plenty to do some light scans and will help give us an idea if we want to move to a paid tier.

Hobby	Freelancer	Business	Enterprise
Great for students, hobbyists and people new to OSINT wanting to explore the power of automation.	Penetration testers, investigators and bug bounty hunters who need to infrequently scan small to medium sized targets.	Businesses that need to perform scanning of medium to large targets and need attack surface monitoring.	Large enterprises or service providers needing higher scan volumes and integration with their own workflows and tooling.
3 scans per month 10 unique targets per scan 1 hour scan duration limit 1 user account 30 days scan data retention	15 scans per month 256 unique targets per scan 24 hours scan duration limit 1 user account 6 months scan data retention	50 scans per month 1024 unique targets per scan 72 hours scan duration limit 3 user accounts 12 months scan data retention	Custom scans per month Custom unique targets per scan Custom scan duration limit Custom user accounts Custom scan data retention
Features in Hobby ✓ Correlations ⓘ	Features in Freelancer ✓ Correlations ⓘ ✓ Investigations ⓘ ✓ Invasive modules ⓘ ✓ Customer e-mail support ⓘ	Features in Business ✓ Correlations ⓘ ✓ Investigations ⓘ ✓ Invasive modules ⓘ ✓ Customer e-mail support ⓘ ✓ Screenshotting ⓘ ✓ Attack surface monitoring ⓘ ✓ SpiderFoot HX API ⓘ	Features in Enterprise ✓ Correlations ⓘ ✓ Investigations ⓘ ✓ Invasive modules ⓘ ✓ Customer e-mail support ⓘ ✓ Screenshotting ⓘ ✓ Attack surface monitoring ⓘ ✓ SpiderFoot HX API ⓘ ✓ Firehoses ⓘ ✓ Custom modules ⓘ ✓ European hosting option ⓘ
Free!	\$79 monthly or \$948 \$749 yearly	\$249 monthly or \$2988 \$2399 yearly	Contact us

SpiderFoot HX <newuser@spiderfoot.net>

Verification Code: 917151

Tue, May 10 • 20:36

SHOW MORE ▼



Your SpiderFoot HX verification code is
917151

If you did not register for SpiderFoot HX, you
may safely ignore this email.

Registration came from IP address,
[REDACTED]

Within a few seconds to a couple of minutes, you should have an email from spiderfoot.net with your registration code. If you do not see it in your main mailbox be sure to check your spam folder.

The screenshot shows the SpiderFoot HX registration verification page. At the top is the SpiderFoot HX logo. Below it, the text reads: "Check your email (keyboardkomando@tutanota.com) for a 6-digit Verification Code and enter it below." A link to "contact support" is provided. There is a text input field for the "Verification Code" with a placeholder "6-digit Verification Code just emailed to you". Below this are three dropdown menus for "Which option best describes your business?", "What's your primary use case for SpiderFoot?", and "How did you find SpiderFoot HX?". At the bottom, there is a checkbox for "By proceeding you accept the Terms of Use & Privacy Policy." and a blue button labeled "Proceed to create account".

Navigate back to the site and enter in your verification code that was provided along with answering the questions below by clicking the down arrow and selecting an option on the menu.

EVIL TWIN ATTACK

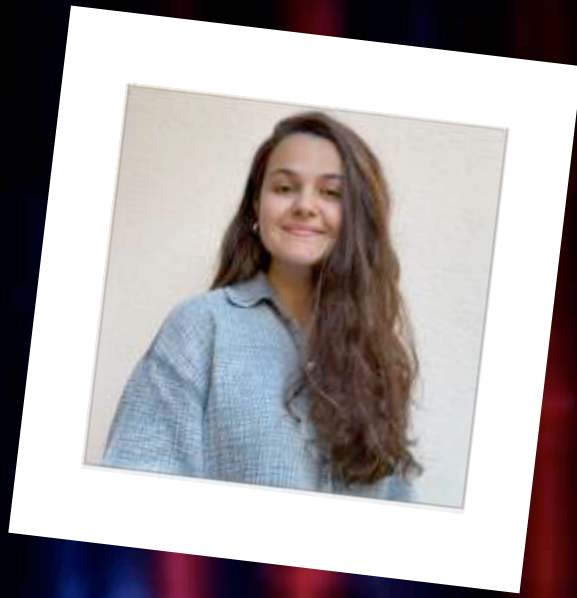




MAYA SANDRA AIT YAHIA

I am a student in cybersecurity at the University of Paris Descartes, in an internship in a company. I am interested in offensive and defensive cybersecurity.

LinkedIn: <https://www.linkedin.com/in/maya-sandra-ait-yahia-bb438b217/>



ELISA CAZERES

Interested in ethical hacking topics, I am preparing a master's degree in cybersecurity at the University of Paris while being an apprentice in a company.

LinkedIn: <https://www.linkedin.com/in/elisa-cazeres/>



INTRODUCTION

Many people use free Wi-Fi in order to connect to social networks, for example, or to an online bank. However, they aren't all aware of the risk of danger to which they are exposed. This may seem trivial but public networks are sources of vulnerabilities. In situations where a malicious person wants to steal data, connecting to these networks increases the likelihood of being the victim. Unfortunately, you also have to be careful with private networks and be aware of some unusual signals. In this article, we will describe an attack that can be performed using vulnerabilities and the non-vigilance of users.

WHAT IS THE EVIL TWIN ATTACK?

Evil Twin Attack is a kind of phishing used in wireless computer networks. It is an identity theft attack that consists of prompting users to connect to a fake Wi-Fi hotspot imitating a legitimate network. Once a user is connected to this fake access point, the hacker can access anything from network traffic to personal information, such as private login credentials.

There are several ways to accomplish the attack. This paper focuses on the stealing of a private network password with a captive portal. Knowing this password means that the hacker becomes a full member of the network and can do everything they want.

HOW TO PERFORM THE ATTACK:

Requirements:

- ☑ Kali Machine: mainly used for advanced Penetration Testing and Security Auditing.
- ☑ Airgeddon: this is a multi-use bash script for Linux systems to audit wireless networks.
- ☑ A configured access point: same SSID as the victim one (for the tutorial SSID is Livebox_ME).
- ☑ Network card: has a physical interface configured in monitor mode. For turning normal network interface cards into access points.
- ☑ A prepared captive portal: with languages such as HTML, CSS.

The Attack:

First of all, connect your network card (for us: wlan0). Then, we need to put it on monitor mode.

```
(root@kali)-[~]
└─$ airmon-ng start wlan0

Found 1 processes that could cause trouble.
Kill them using 'airmon-ng check kill' before putting
the card in monitor mode, they will interfere by changing channels
and sometimes putting the interface back in managed mode

  PID Name
  474 NetworkManager

PHY      Interface      Driver      Chipset
phy1     wlan0              rtl8192cu   Realtek Semiconductor Corp. RTL8192CU 802.11n WLAN Adapter
              (monitor mode enabled)
```

Figure 1. airmon-ng

You can find the mode with: `> iwconfig`

To have the list of every nearby access point, type the command: `> airodump-ng wlan0`

A first step is to retrieve some information, such as the MAC address or the SSID (and to know from which telecom operator it comes from, by looking at the name of the network “SFR...”, “Free...”. This information is important for the future captive portal). We need this information in order to create the “evil twin”. For that, perform a network scan with:

```
> airodump-ng wlan0
```

If your Kali machine doesn't have **airgeddon**, you need to install it.

Now, launch this tool: `> airgeddon`



Figure 2. Airgeddon script

As it is written, the script should only be used for educational purposes.

OUR JOURNEY INTO BUILDING AN UNDETECTABLE PYTHON RANSOMWARE

```
1 1001 01 00
00 01 1001 01 00110
101 00 01 100101 001101 00
01 00 01 1001 01 001101 00 01 100
01 001101 00 01 1001 01 001101 00 0
001101 00 01 100101 001101 00 01 1001
0 00 01 100101 001101 00 01 1001 01 001
1101 00 01 1001 01 001101 00 01 100101 0
00101 001101 00 01 1001 01 001101 00 01 10
0101 001101 00 01 1001 01 001101 00 01 10010
1 1001 01 001101 00 01 100101 001101 00 01 10
001101 00 01 100101 001101 00 01 1001 01 00110
101 00 01 100101 001101 00 01 100101 001101 00
01 01 001101 00 01 100101 001101 00 01 100101 0
01 100101 001101 00 01 1001 01 001101 00 01 10
01 001101 00 01 1001 01 001101 00 01 100101 001
001101 00 01 100101 001101 00 01 1001 01 001101 0
001101 00 01 1001 01 001101 00 01 100101 001101 00 01
001101 00 01 100101 001101 00 01 100101 001101 00 01 100101 001101
001101 00 01 100101 001101 00 01 1001 01 001101 00 01 100101
00 01 100101 001101 00 01 1001 01 001101 00 01 100101 001101 00
1 001101 00 01 100101 001101 00 01 1001 01 001101 00 01 100101 0011
1101 00 01 100101 001101 00 01 100101 001101 00 01 1001 01 001101
00 01 100101 001101 00 01 100101 001101 00 01 1001 01 001101 00 0
01 00 01 100101 0 00 01 1001 01 0011 01 100101 001101 0
01 001101 00 0 1101 00 01 10 00 01 100101
001101 0 001101 0 01 00 01 10 1 100101 0
00101 00 001101 00 1 100101
0 01 10 100101 00
01101 1 00 01 1
00101 0 01 001101 00
001101 00 1101 00 01 1001
101 00 01 100 01 00 01 0 1 001101 001 01 001101 0
0 01 100101 001101 00 01 1001 1101 00 01 100101 001101 00 01
1101 00 01 1001 01 001101 00 0101 001101 00 01 100101 00110
101 001101 00 01 100101 0011 01 1001 01 001101 00 01 10010
00 01 100101 001101 00 01 0001101 00 01 100101 00110
00 01 100101 001101 00 0 001101 00 01 1001 01 001
1 100101 001101 00 01 1 101 00 01 100101 001101
00 01 100101 001101 1 1 01 001101 00 01 100
001101 00 01 100101 001101 00
1 00 01 100101 001101 00 01 1 1101 00 01 1001 01 001101 00
00101 001101 00 01 1001 01 00 101 001101 00 01 1001 01 001
1001 01 001101 00 01 10010
1 00 01 100101 001101
00101 00
```



BARRAULT VICTOR ORNELLA FABI

The authors are both French and studying computer science and cybersecurité at University Paris Cité in France. As part of their studies, Victor works as a security analyst and Ornella works as an infrastructure engineer.

Contact us at :

victor.barrault@protonmail.com

<https://www.linkedin.com/in/barrault-victor>

ornella.fabi@outlook.fr

<https://www.linkedin.com/in/ornella-fabi>

INTRODUCTION

Ransomware is a growing threat on the internet today. In 2021, more than 30% of all businesses got hit by ransomware, and it cost them more than 20 billions dollar in losses. The strategy of ransomware is simple. Encrypt all the data of the user, and offer the key to decrypt it in exchange for a ransom. You can target a lot of usual computer users, or big companies to try to get a big ransom. In both cases, the goal is the same, but the program would be a lot different.

This article focuses on the technical and theoretical side of building ransomware in Python. We do not recommend building such a program as we wrote this article in order to better understand the train of thought of a malicious person. To reproduce the tools we built, you will need a virtual machine, or your own computer if you like taking risks. You will also need Python, gcc and clang installed, as well as at least pyinstaller.

Choosing Python to build a ransomware

Ransomware is not a new threat anymore and people have started to study them. Python is not a language we see a lot when studying their code. Most of the tools malicious people use are built-in C, C++, or more recently in Go.

They choose to do so for mostly two things:

First, with these languages, you are able to do low-level calls, building programs that interact directly with the system. These languages also give us the opportunity to edit the stack during runtime. That alone is a really great tool considering what we want to do: take advantage of the system doing things users do not really, without getting detected.

Secondly, these languages are also really easy to distribute, making an executable out of a C or Go source code is one command away, as these languages are made to easily build distributable software for Windows and Linux. An assailant could also choose to build their ransomware in C#, being particularly suited for Windows development. But C# is built on the Windows API, which makes it easy monitorable. Each API made by a C# program is evaluated by Windows Defender, and thus making a malicious program is far harder than making one in C or even in Python.

Is Python a great choice to build your ransomware?

We will start by explaining why you wouldn't choose Python to build your own ransomware: It has no low-level direct interface, no way to write straight to memory, as it is an interpreted language, and its bytecode is executed through a virtual machine. To distribute your ransomware, you have two possibilities: The victim has Python on their computer or you ship Python directly with your program.

The first option is nearly impossible as most of the population, and mostly the ones we want to focus on, do not have Python installed. So, we need to compile our Python code to a Windows portable executable (PE). Three problems emerge from that: First, we must ship Python in our executable, the packaged executable we need to send is around 10MB, which is big. Moreover, it is slow, as it needs to start the Python environment and be re-interpreted on the fly, and finally, it causes a lot of false positives in detection (we will discuss this later).

With all that being said, is there a use case for Python ransomware? Why would you choose such a language? Because building in Python is fast and easy. In fact, it is far easier to build a working program in Python than it is in C or Go. A young assailant could easily learn Python by themselves, but wouldn't learn C/C++ system programming without studying it in college. It's also really fast. You don't need to invest a lot of time into development to make a working program. Python also offers a vast number of modules; everything is prebuilt for you. For our ransomware, we heavily used the "cryptography" library, Pathlib and winreg to accomplish most of our tasks.

What does it take to build ransomware in Python?

We had to build our ransomware with the resources we had. As we are not a state group with millions of budgets, we do not have access to 0-days exploits and vulnerabilities, and we did not have a lot of time. So, all the choices were made thinking about making it simple to develop, and, even without using ultra-advanced techniques, we would still cause a lot of damage.

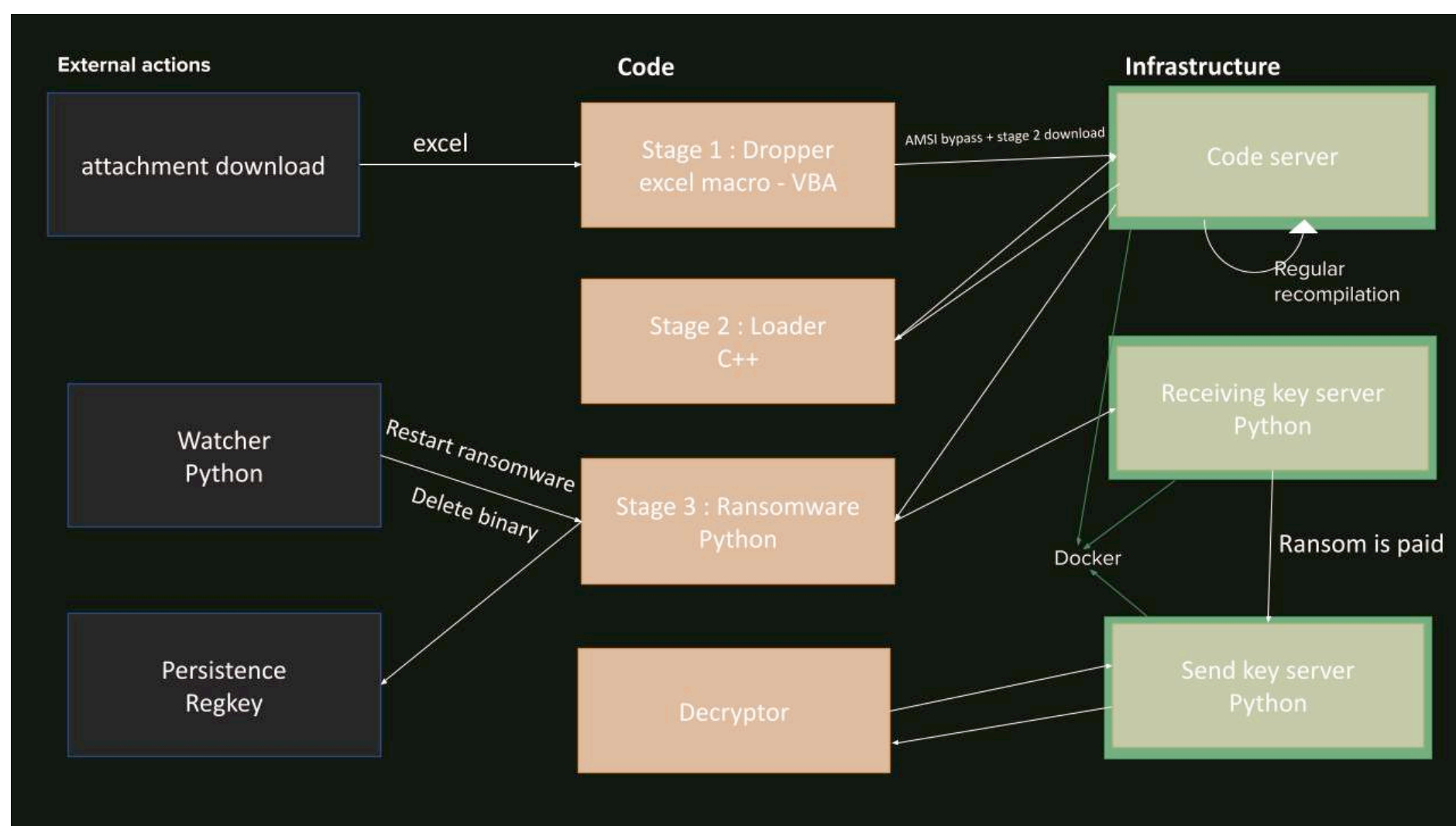
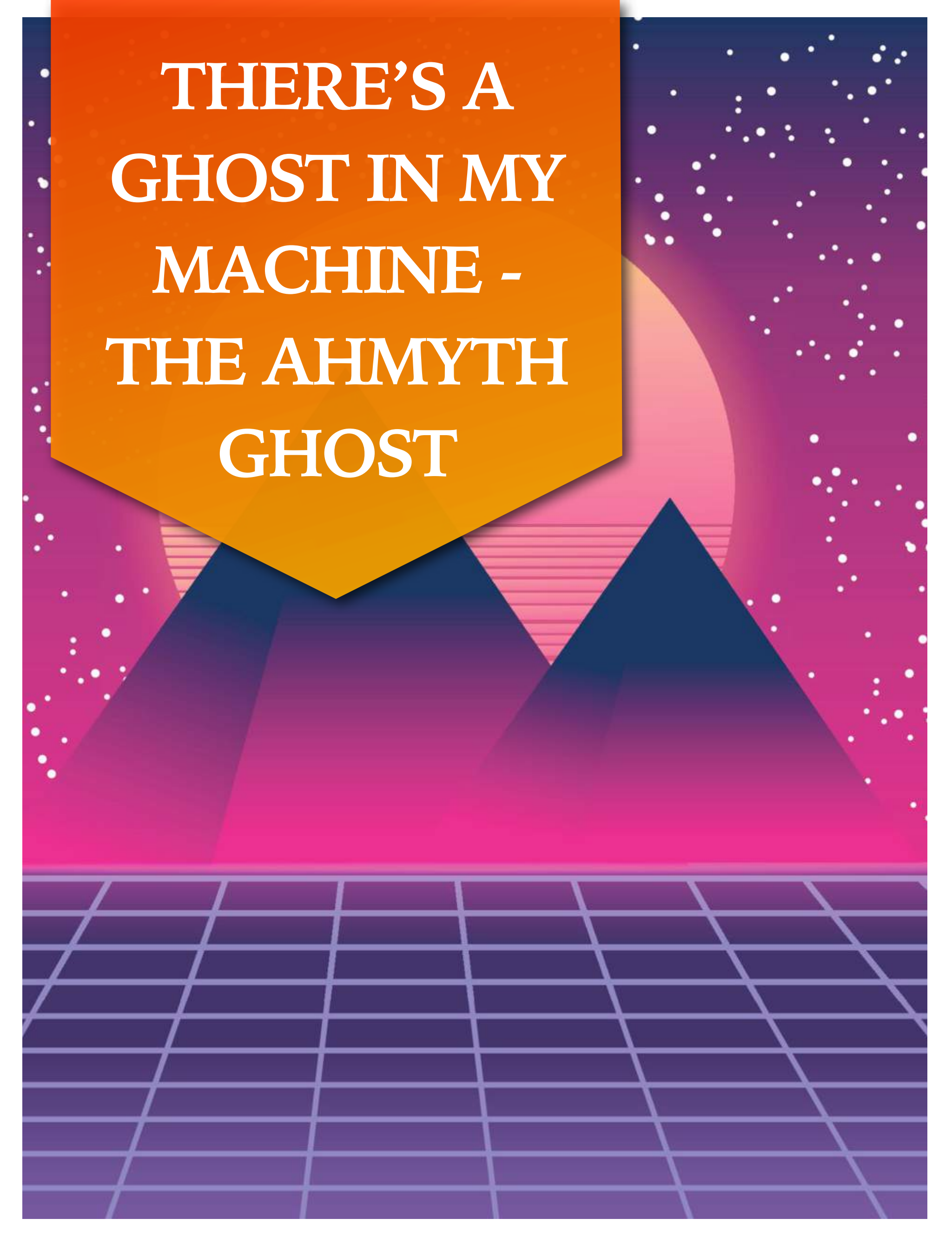


figure 1: General Architecture

Our ransomware is mostly deployed on the victim's computer through three steps: first, what we called the dropper, is an Excel document having a macro being downloaded, the second stage, the dropped, which downloads and starts the final payload.

Why did we choose to make this architecture?

First of all, because it is modular. We can change a piece of it without changing anything on the other pieces. That is a



THERE'S A GHOST IN MY MACHINE - THE AHMYTH GHOST

BYRON GORMAN

Current Position - Threat Hunter-intelligence /OSINT Specialist.

Company owner of - Octoberbluedevops & Codeblackglobal. Professional

Goals - Supply knowledge to those in the industry.

Relevant Achievement - securing systems for middle to small businesses.

byron.gorman@codeblackglobal.com

byron.gorman@octobeo0rbluedevops.com



We can hack into any Android across the internet if we have the ability to build a simple APK. The most important thing is to have the ability to monitor and control a device without the subject knowing. This can be helpful in several circumstances and can help people who wish to monitor their devices that have been distributed to employees and other people or to monitor someone for investigation purposes. No matter what the reason is, we will learn how to achieve this deployment. Cyber Ghosting means deploying and controlling a system without the end-user being aware that they have been compromised. So we are going to dig up our first ghost and find it a home to haunt.

“I Am The AhMyth Ghost!”

Let's get our ghosting tools together; we have some tools that we must install for our payloads which will allow us to start to deploy our ghost. Now let's download our program and clone it by issuing the following commands in your Kali Linux distribution:

- 🔹 Open our favorite Linux hacking deployment, Kali Linux
- 🔹 Open a web browser in Kali Linux and navigate to <https://www.github.com/Ahmyth>
- 🔹 Input the URL in the Kali terminal

The purpose of the tool is to automate the payload Creation Process.

- 🔹 Build an AhMyth APK Payload
- 🔹 Install the payload on the Victim Android Device
- 🔹 Take Control of the Victim's Android Device using the Victims Lab
- 🔹 Remain in the device unknown

HOW TO INSTALL, CONFIGURE, AND USE HASHCAT IN AN AZURE VM STEP-BY-STEP



JOSÉ PABLO

José Pablo is a Science and Technology enthusiast. He is the owner of Green Valley Business, S. L., an IT security solutions company. Also, he has been a lecturer in computer security, BigData, Artificial Intelligence and programming for several years.

You can contact him at <https://www.linkedin.com/in/jphernandeza>

INTRODUCTION

Whether it's as part of Red Team activities, ethical hacking, or a security assessment, the issue is that we'll need to break passwords at some point.

There are many tools that allow us to crack hashes, such as John the Ripper, etc. But when the going gets tough, it's time to break out the big guns and, in this case, the almost unanimous winner is Hashcat, a free and open source tool that at the time of writing is running version 6.2.5 with a lot of good features, among which we can highlight:

- ☑ Multi-OS (Linux, Windows and OsX)
- ☑ Multi-platform (CPU, GPU, APU and anything that can be programmed with OpenCL)
- ☑ Multi-hash (350+ types of hashes)
- ☑ Supports distributed cracking networks.
- ☑ And many more that we can see in his page (<https://hashcat.net/hashcat>)

On the other hand, the real potential of Hashcat comes out when using machines with really high performance. We are talking about machines equipped with GPUs such as the nVidia Tesla V100, a graphics card that in its 32GB version is advertised on Amazon.com for over \$12,000.00.

It is not something that most of us can afford, so another way to enjoy all that power without hurting our pocket is by renting a cloud service that has this type of device. Again, we will have a range of cloud services that rent virtual machines with this equipment: Google Cloud Services, Amazon Web Services or Microsoft Azure. The latter is the service we will use on this occasion.

WHAT IS A HASH?

Without going into the mathematical foundations on which hash algorithms or hash functions are based, we will say in a very simple way that a hash function is an operation that, from an input data stream (a string of characters, a document, a file, etc.) generates an alphanumeric string of characters at the output.

Output of the hash function has two important characteristics:

- ➡ The result is of fixed length for each algorithm, which means that, regardless of the length of the input, the output will always have the same length.
- ➡ The algorithm is one-way, i.e., having any input and applying the corresponding algorithm, we will obtain an output, but there is no algorithm that applied to the resulting output we obtain again what was put in the input.

Therefore, the only way to 'break' a hash is to apply the algorithm to input elements (by dictionary, brute force) and check if any of the input elements matches the output element we have, in which case we can say that we have 'broken' that hash.

AZURE SERVICES

Create one account.

Now we will see how to create a Microsoft Azure account. If you already have one, you can skip this point and go directly to how to configure it to use the advanced virtual machine services with the nVidia V100 GPU.

When accessing <https://portal.azure.com> with our preferred browser, we will be redirected to the login page, as shown in image 1.

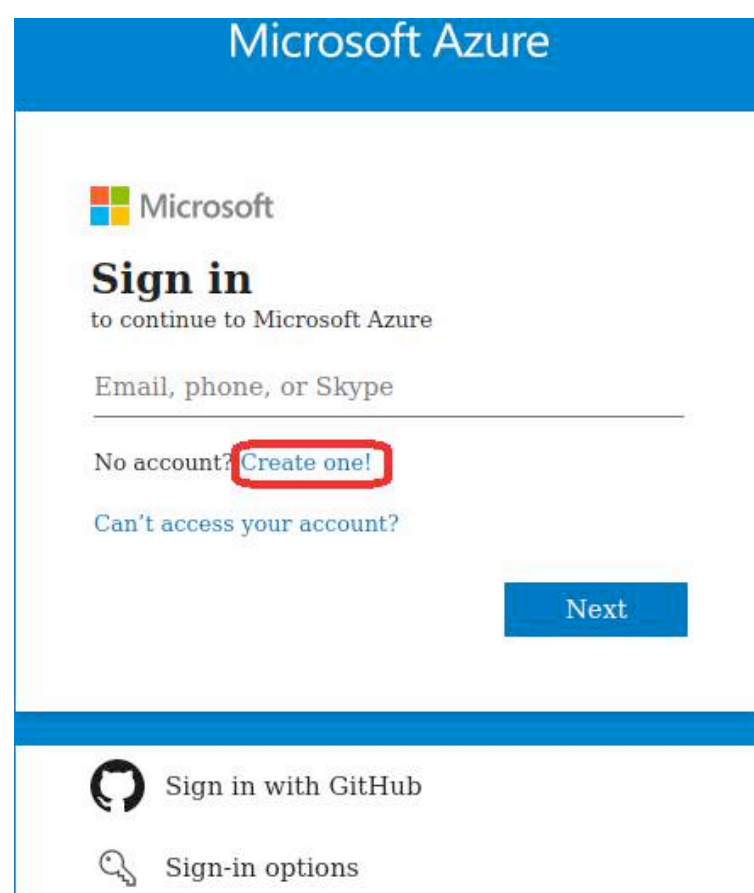


Image 1

If we click on Create one! It will even give us the possibility to get a new email address in the domains @outlook.com and @hotmail.com.

Once we have created the account, and solved the different crossword puzzles so that Azure can make sure we are not a robot, we will have the Azure welcome page.

THE CEWL WAY TO SCRAPE WEBPAGES



GREGORY HAAPAOJA

Greg is an Information Security Analyst and Penetration Tester at OCD Tech, joining the team in January of 2019. Prior to joining the team, he specialized in business management. Greg started at OCD Tech working with a globally significant financial services organization developing privileged access management tools and processes for Windows, Active Directory, and other specialized applications. Since joining OCD Tech's Advisory Services team, his current projects include penetration testing, IT general controls reviews, vulnerability assessments, wireless assessments, and social engineering. Greg is OCD Tech's resident password cracker and is responsible for the development and maintenance of password cracking hardware, software, and associated processes, in order to support OCD Tech's penetration testing practice, as well as ongoing information security research.

LinkedIn: <https://www.linkedin.com/in/gregoryhaapaoja/>

OCD Tech: <https://ocd-tech.com/>

As a beginner to the “hacking” industry, there are many tools you encounter where you have to determine this: how effective can this tool be against my target? If CeWL has come onto your radar, you have likely been researching ways to scrape information off of a website, maybe for open-source intelligence reasons, maybe you want to build a wordlist of common words the target uses, or maybe you just enjoy analyzing information (if you like hacking, I am sure you like this at minimum). This tool can be used for a multitude of reasons, and below we will dig into the most common use case, as well as additional functionality of the tool.

So, What Is CeWL?

CeWL, an acronym for Custom Wordlist Generator and pronounced as “cool”, is a tool that takes the input of a URL and, via automated methods, recursively parses and collects specific parts of webpages that it finds. Parsing and collecting, often termed “spidering” and “scraping,” are the main functions of this tool. These terms will be actively used during this guide, so if they are new to you, it is time to become comfortable with them.

One of my first roles in penetration testing was as a dedicated “password cracker”. Before taking on my full responsibility of on-site and hands-on testing, I was essentially deemed as the “guy in the van”, positioned outside the targeted network of my fellow penetration testers. As my team would gather password hashes, whether they were wireless, Active Directory, or some unknown hash, I would be tasked with attempting to crack them as quickly as possible to assist my team in escalating their privileges. Of course, there are large wordlists out there and I could always brute-force and pray, but lucky me, I stumbled upon CeWL.

“Live off the land”, a common phrase used by penetration testers and hackers alike, is a methodology used to narrow one’s efforts to utilizing what is right in front of you. In the story above, this is where I found myself. With CeWL, I was able to scrape our client’s webpage and build a list of words that, when effectively combined, actually cracked the client’s (weak) password.

For example reasons, I will rename the client I was working with at the time and call them “The Wardrobe Institute of Narnia”. Now, since these words are actively used on their website, as well as the year of “50”, the year the company was founded, I was able to create a dictionary cracking attempt using CeWL that was able to crack the password “WardrobeNarnia50”. This is a password that would not have been found in my common wordlists and would likely not have been brute-forced in this century.

Password cracking is just one example of what CeWL can be used for. In the sections below, I will detail where you can find and install this tool, what commands can be run with this tool to form output that could assist you with a scenario like the one above, and other scenarios that this tool may also prove to be handy in.

First off, CeWL can be found and installed from GitHub at this location: <https://github.com/digininja/CeWL>, or by simply running “apt install cewl” on most hacking-based distributions.

CTF WARGAME - MR-ROBOT





GABRIEL LAZO CANAZAS

ECPPT, CPTE, C|EH, LA ISO 27001, ISFS ISO 27002

Systems Engineer, specialized in Cybersecurity, IT Forensics and Pentesting, with many industry certifications. Teacher for many institutions and speaker at the Mundo Hacker Day Conference (Madrid) as well as other Latin-American cybersecurity events.

He has also written and contributed with the Mundo Hacker: Books & TV series, cybersecurity technical books for the Spanish RAMA Multimedia Editorial and a cybersecurity TV reality show for DiscoveryMax, TVE and Canal 13 - Colombia. Currently works as CEO of EnHacke, a Peruvian cybersecurity firm offering services to top companies of the main industries.

You can follow him here:

LinkedIn: <https://www.linkedin.com/in/gabslazo/>

Twitter: @gabsitus

Blog: www.enhacke.com/blog

In this article, we are solving one fun as hell wargame that has been created as a CTF type of challenge. The rules are simple:

- ☑ BOOT2ROOT: This means the ultimate goal of this wargame is to gain root privileges
- ☑ Get the Keys: There are three keys hidden inside the OS that are used to show progress and also to let you know you are on track
- ☑ SO FILE for the MR-ROBOT Wargame can be downloaded here:
<https://mega.nz/file/jFxOURpb#XGhTIO1rOiTqxaugBuIp7jBHDBMuNcoIt5cTG2Gd42E>
- ☑ Not a rule but in order to have a fully functional attack - victim lab, both virtual machines should be configured as BRIDGED
- ☑ There are different ways to get the same outcomes, so keep an open mind and enjoy the ride
- ☑ Both machines need to be on a bridged environment

Our strategy or approach will be:

- ☑ Reconnaissance & Network Discovery
- ☑ Information gathering & enumeration
- ☑ Attack & Exploitation
- ☑ Privilege escalation

I will be using open source tools exclusively so you can reproduce the whole scheme. Let the games begin:

RECONNAISSANCE & NETWORK DISCOVERY

To get targets, it is always recommended to inspect our own network. A quick ifconfig will reveal our local IP and what network we should be hearing about.

```
└─(root@kali) - [~]
└─# ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.0.96 netmask 255.255.255.0 broadcast 192.168.0.255
```

We now know we are on the 0 subnet, we want to scan for the entire network (minus the broadcast address).

```
└─(root@kali) - [~]
```