REMNUX

Ivan Marchany

CYB 630 INCIDENT RESPONSE MANAGEMENT Dr. Knapp

Table of Contents:

What is REMnux?

REMnux is a Reverse Engineering and Malware Analysis toolkit that analyzes malware in a safe environment. REMnux is an Ubuntu-based Linux distribution that contains many tools for analyzing malware on Linux and Windows. This tool is an open source and is widely used by reverse engineers and malware analyst for forensic investigation and critical security incidents. As a Digital Forensics/Incident Response standpoint, REMnux is an essential toolkit to analyze malware during an incident and can perform static or dynamic analysis in a sandbox environment. The distro is lightweight which mean it runs smoothly on your virtual software. This tool can be used during a containment phase of the IR lifecycle.

Main Features:

REMnux has several features that are included on a cheat sheet once is installed. In the cheat sheet, it includes a documentation of all the tools that have been tested by the developers and are useful for examining malware. They made a list because it makes it easier to know what tool would be the proper one to use when an incident is timed. Most of the tools are command-line based and some of them are applications that run in the GUI. According to REMnux documentation, the tools are broken down into several categories:

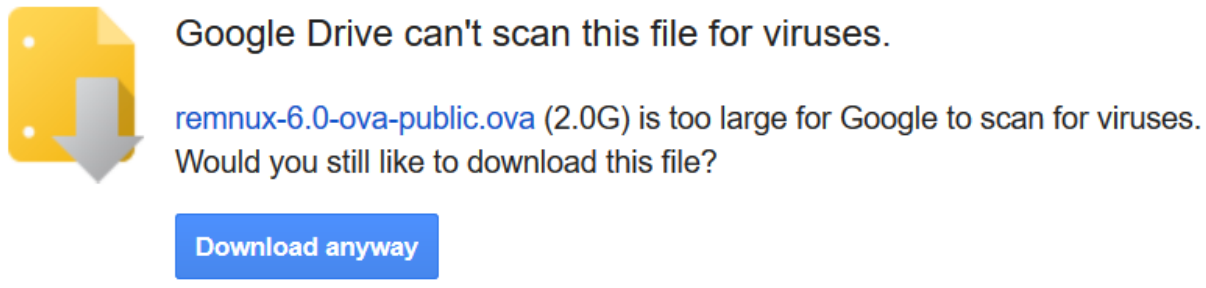
- Examine browser malware
- Analyze malicious document files
- Extract and decode suspicious artifacts
- Handle laboratory network interactions
- Review multiple malware samples
- Examine the properties and contents of suspicious files
- Investigate Linux and Windows malware
- Perform memory forensics

Tool Demonstration

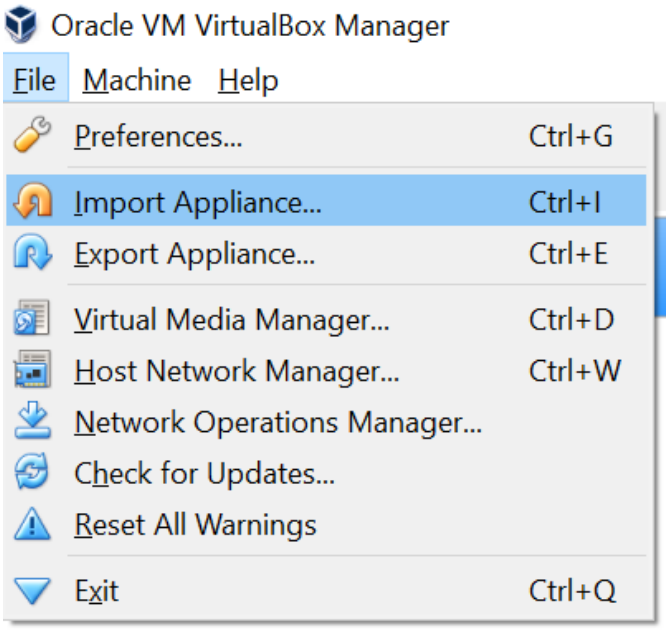
In this section, I will demonstrate some of the tools in the list that are useful as a forensic analyst. I'll be going over some tools not covered in class and try to use at least one tool per category. Before I demonstrate the tools, I want to cover briefly on how to install and configure Remnux as your malware lab.

Download and Install Remnux

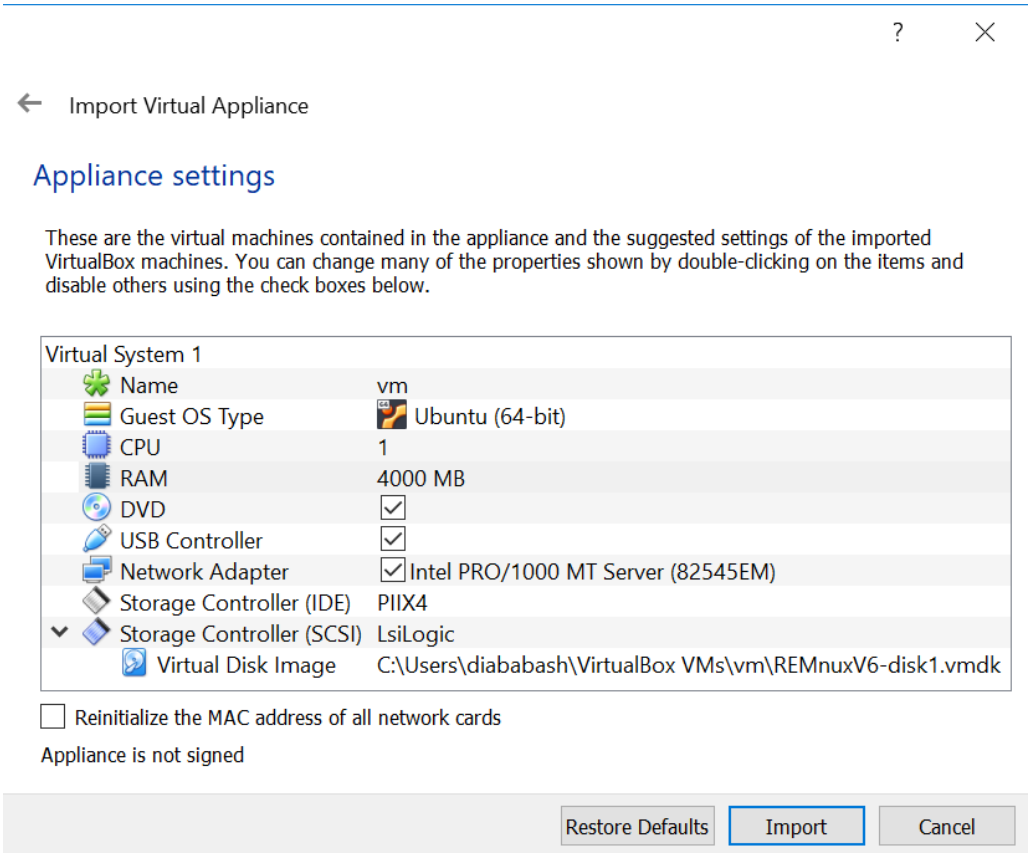
You can download REMnux from their website which takes you to this link: https://docs.google.com/uc?id=0B6fULLT_NpxMampUWlBCQXVJZzA&export=download. Unfortunately, this file cannot be scanned in VirusTotal because the file is over 256MB. However, my Bitdefender anti-virus did not detect malware, so I assume this file is safe.



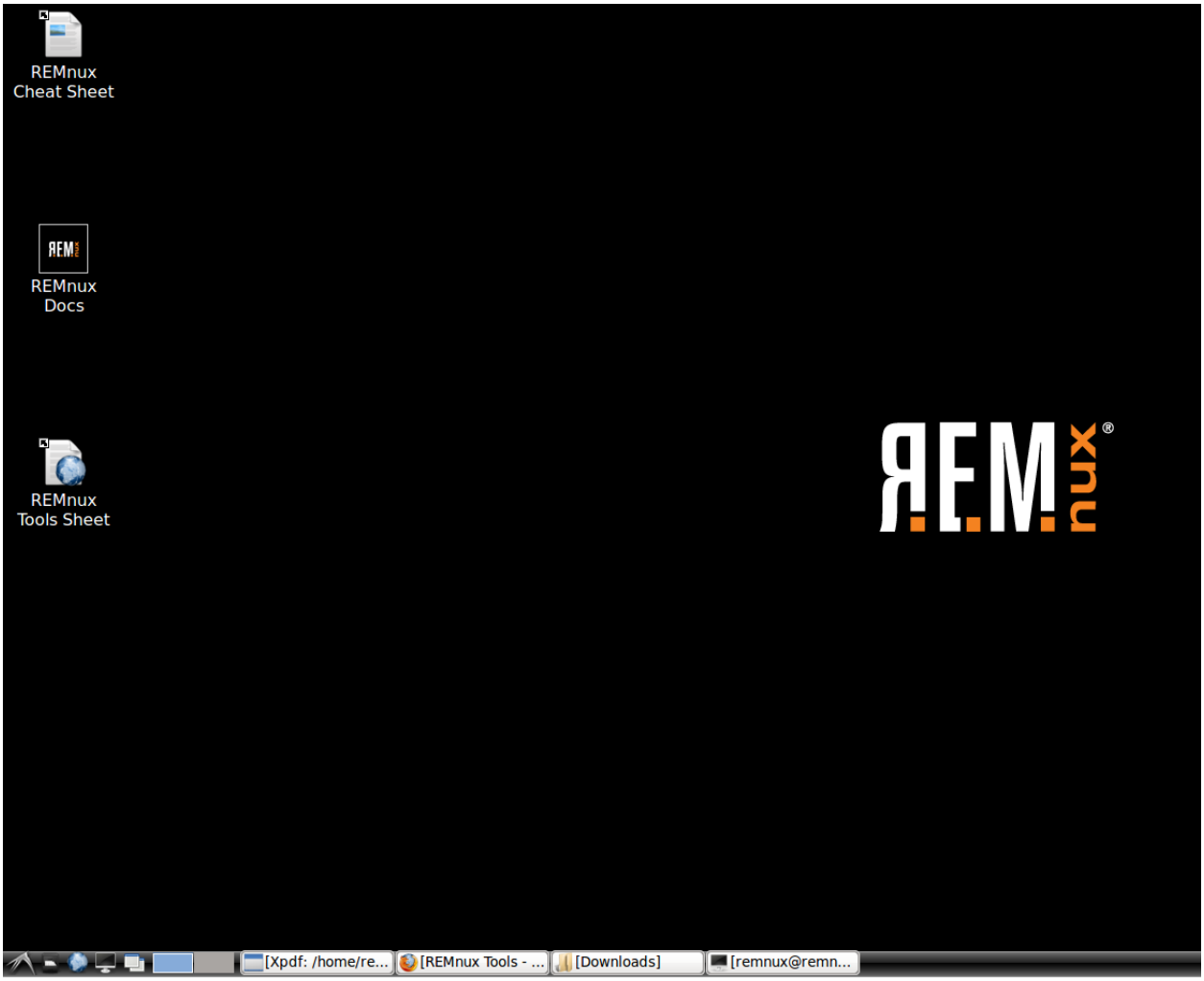
The file remnux-6.0-ova-public.ova is a virtualization format that is meant to run in a virtual machine. If you are using VirtualBox, you click on the file and then click import appliance.



In the appliance settings, search for your REMnux file and then configure it as your virtual machine.



Once the virtual appliance is installed, you may run REMnux and it will take you straight to the Desktop GUI view.

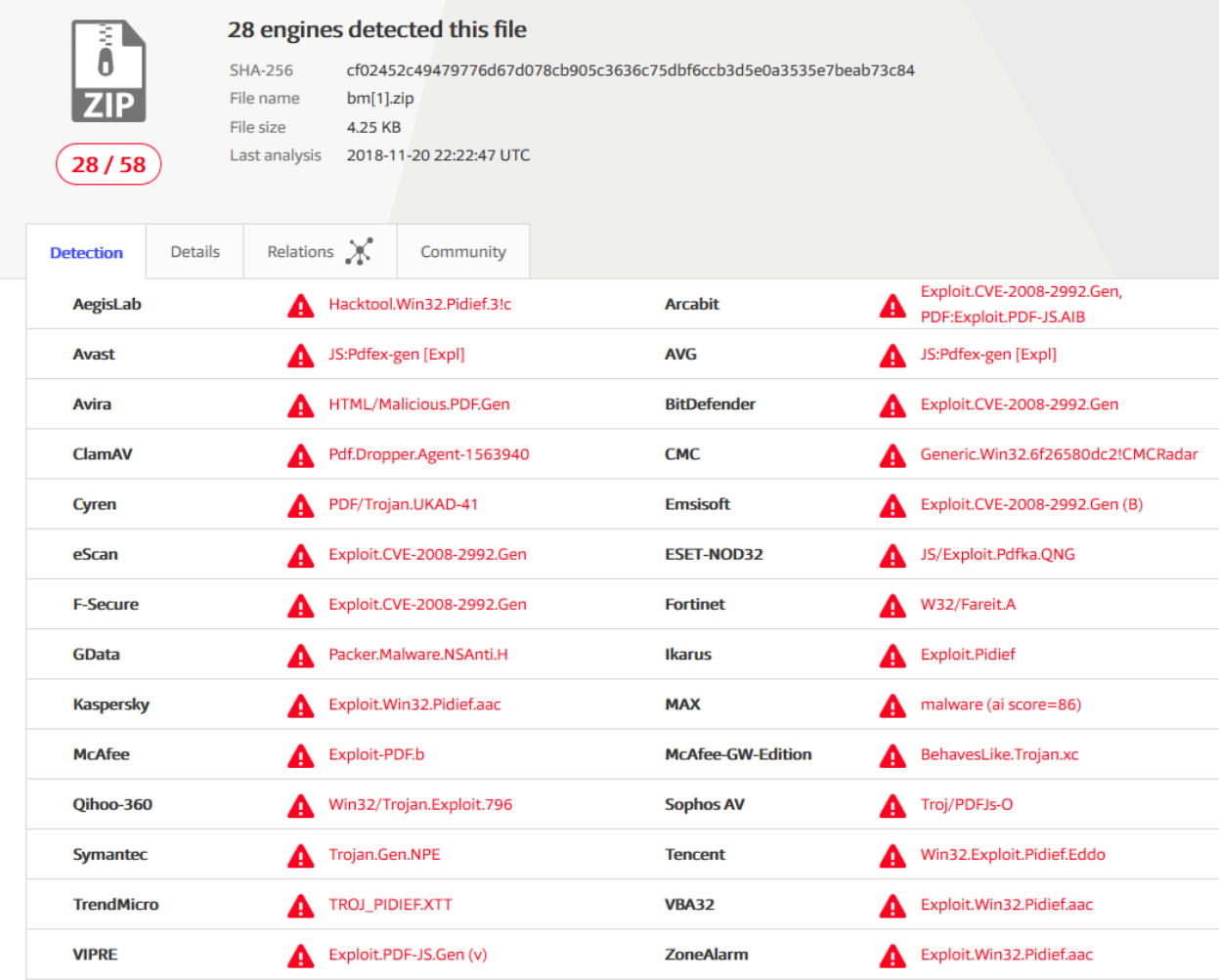


This is the Desktop GUI and from here you can start using the tools by looking at the REMnux Cheat Sheet.

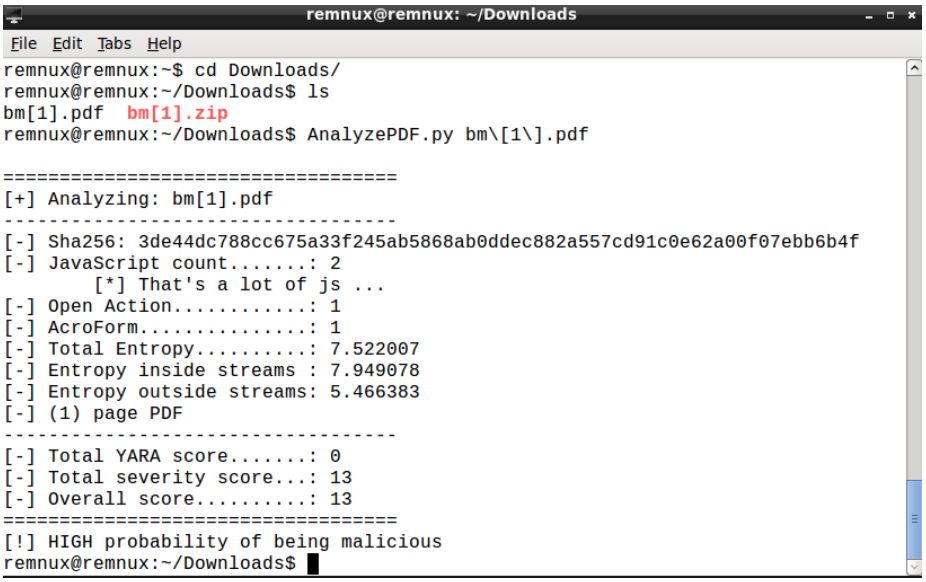
AnalyzePDF.py

The first tool I want to start with is AnalyzePDF.py. This tool is a command that examines a malicious PDF file and finds provide an overview of the PDF to determine if is infected or not.

The first thing you do is research on Google and find samples of infected PDF. I found one in MalwareDomainList website via zip file. When I downloaded, I checked in VirusTotal and here are the results:



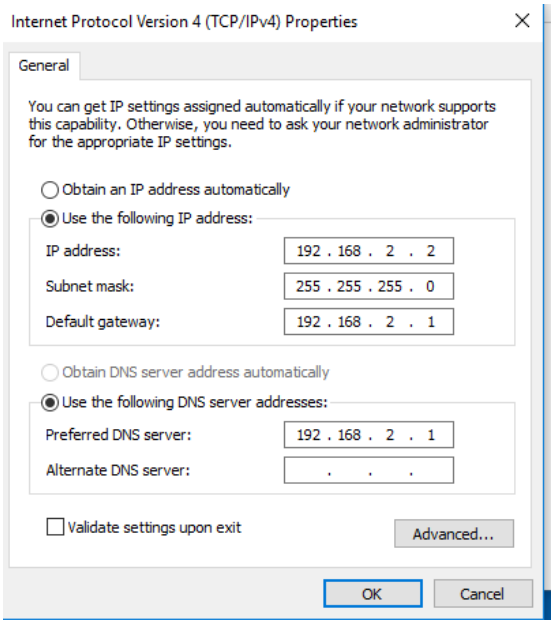
This file can execute malware if you open the pdf so once I downloaded it, I switched the network adapter to host-only adapter so is not connected to the internet or to my host machine. I opened the Downloads folder and unzipped the file. I went to the terminal, typed the AnalyzePDF.py command and here is the output:



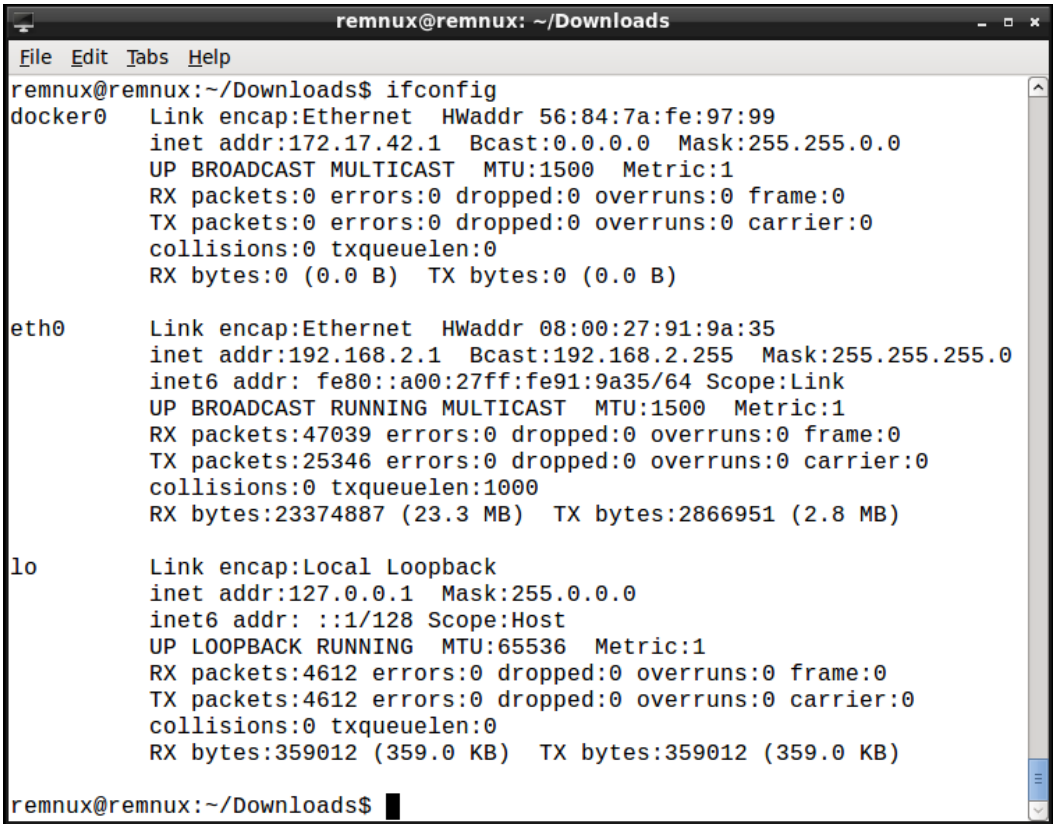
This PDF file indicates that the probability of this file is highly malicious. It outputs a sha256 hash to validate and detected two JavaScript code. This command can be useful for investigating a suspect's drive that contains pdf files and may be malicious. It is also useful before you open any files during the investigation.

Fakedns & Inetsim

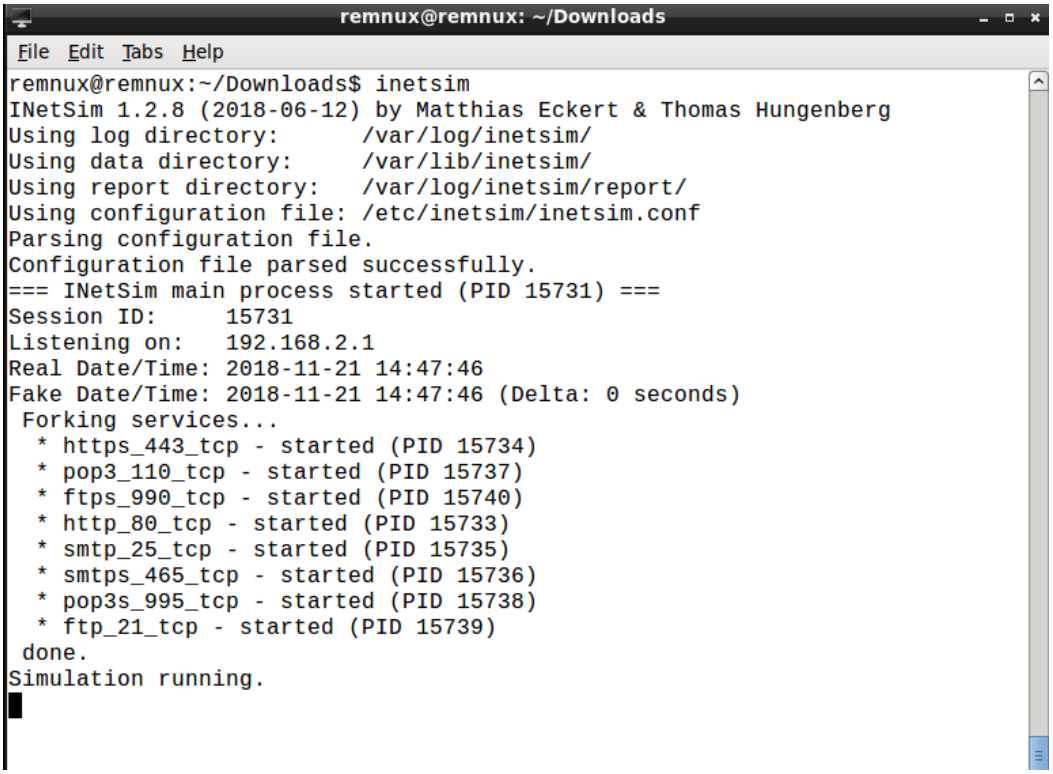
The next tool demonstration is how to use fakedns and inetsim. Fakedns acts a DNS server on REMnux and inetsim simulates network services in a lab environment. It requires to have a Windows machine that points to Remnux as a router. Once they can talk to each other, then it would get all the request that Windows use for Internet. The first screenshot is configuring your network configuration on Windows.



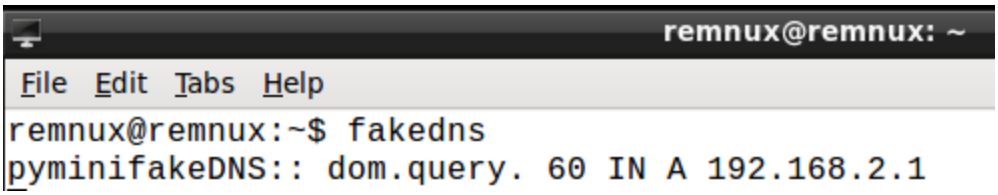
Once the Windows machine is configured, then you configure the REMnux network settings.



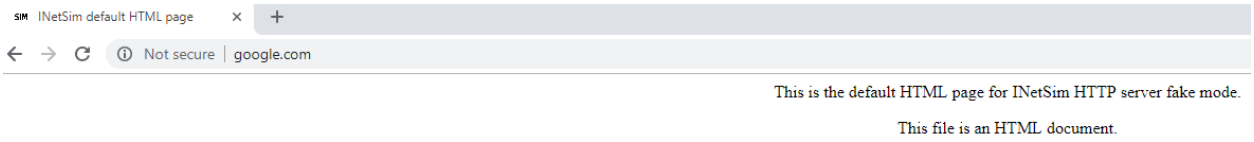
Once is configured, you can perform the command inetsim to start simulating the network services.



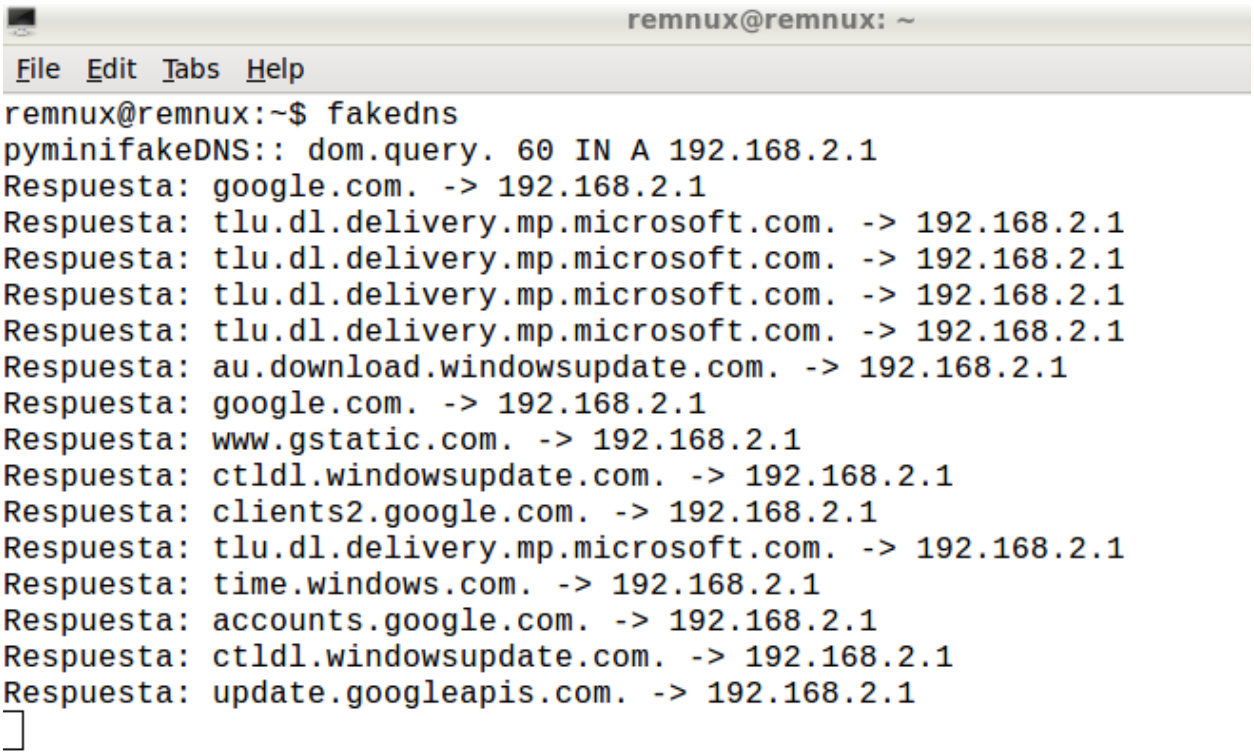
Once the simulation is running, you can run the command fakedns.



Once you see this output, go back to your Windows machine and start browsing the web like typing www.google.com



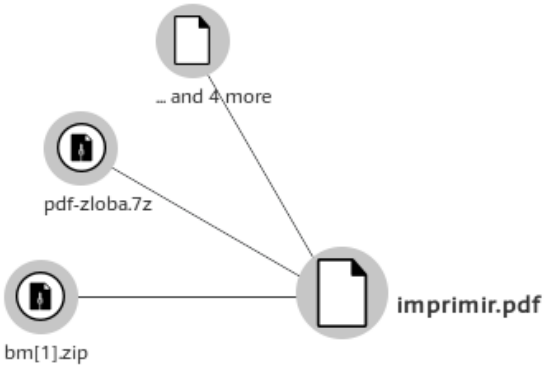
In Google Chrome, when you search for google.com it returns a message indicating that you a going through fake HTTP server. If you go back to Remnux, you see the results after you entered google.com.

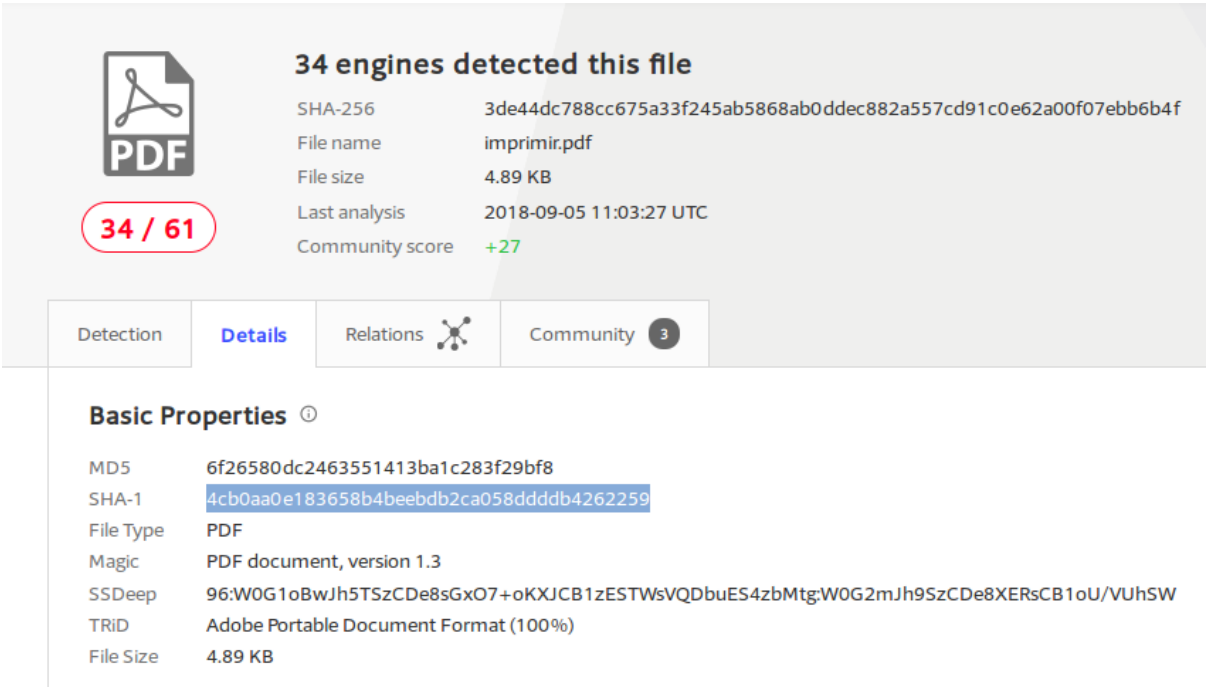


I'm not sure why it outputs in Spanish because this VM is configured in English. Respuesta means response for the DNS query. DNS queries run in port 53 UDP. These responses are what outputs from the Windows machine. While analyzing the responses, www.gstatic[.]com seems to be odd. According to HowToRemove, gstatic is probably a virus that redirects to gstatic[.]com. The Windows machine is not connected to the Internet so at least the virus will not be spread across my network. As an Incident Response perspective, this tool can be useful to trick the victim's machine and you can analyze what the user is trying to do. This information is valuable for evidence to collect and analyze what the user was trying to visit.

Vt-Tool.py

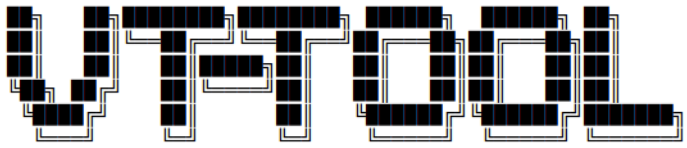
The next tool I want to demonstrate is the vt-Tool.py command. This command allows to determine the name of the malware by querying to VirusTotal. I can use the same malicious PDF file hash to determine the name of the malware and what category falls into. To identify the category, you must include the hash of the malicious file. To make it easier, I went back to VirusTotal and submit the malicious PDF. Under the details, you find different types of hashes. I copied the SHA-1 hash. The file name has relations to bm[1].zip which is the original PDF file I analyzed for the first command tool.





Once I copy the hash, then I paste it in the terminal and execute the vt-Tool.py command.

```
File  Edit  Tabs  Help
remnux@remnux:~/Downloads$ vtTool.py -hash 4cb0aa0e183658b4beebdb2ca058ddddb4262259
```



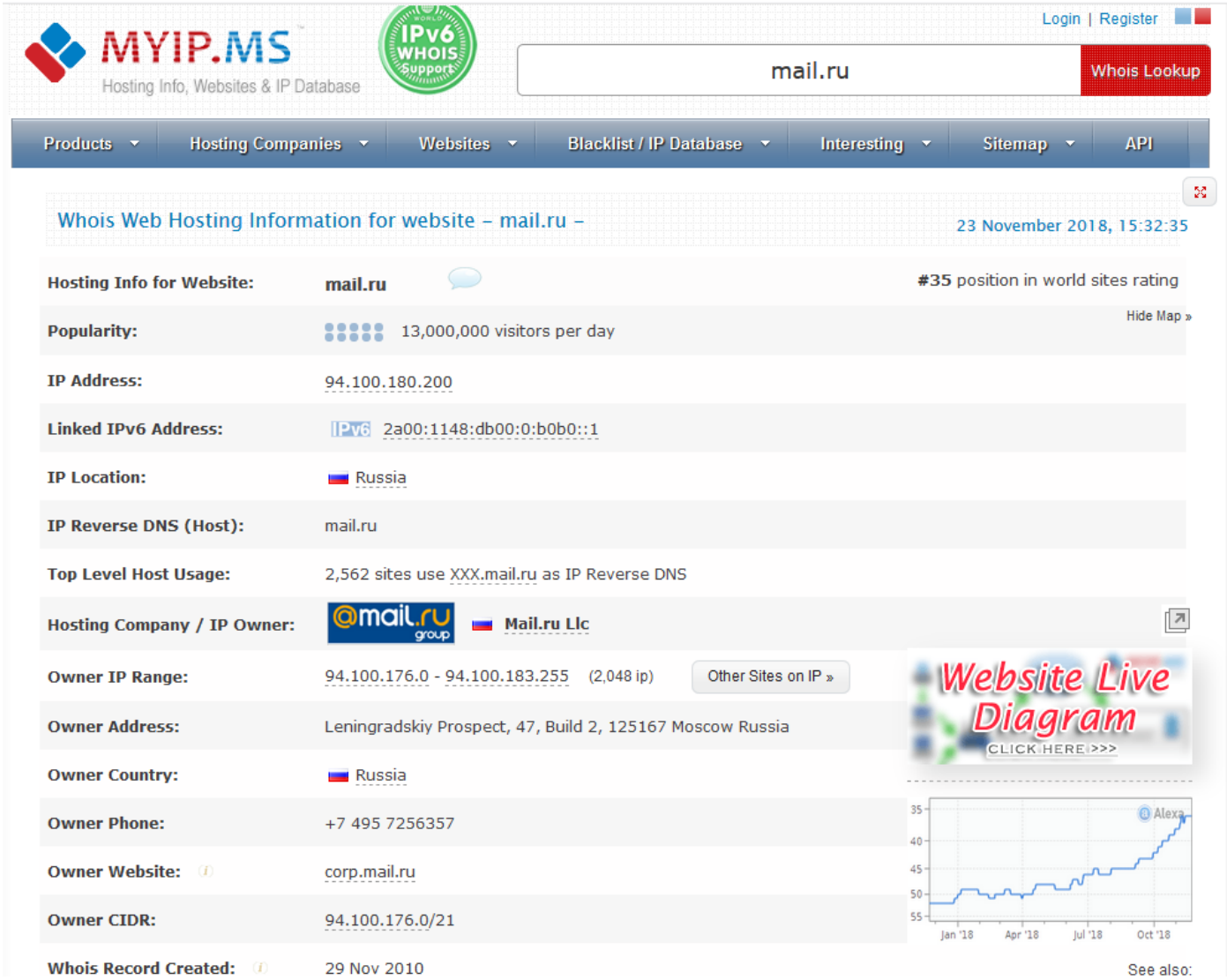
```
v0.9.3
 Find the name of the evil
 Robby Zeitfuchs, Mark Lawrenz
 Copyright (c) 2013-2015


--- scanner malware classification ---
  ransomware: 0
     dropper: 1
     exploit: 17
  downloader: 0
    riskware: 1
     rootkit: 0
        worm: 0
      trojan: 8


--- average detection rate ---
count hashes: 1
totalscanner: 61
   positives: 34



--- scanner malware family determination ---
Most frequent word: pidief (count=10)
Second most frequent word: expl (count=2)


remnux@remnux:~/Downloads$
```

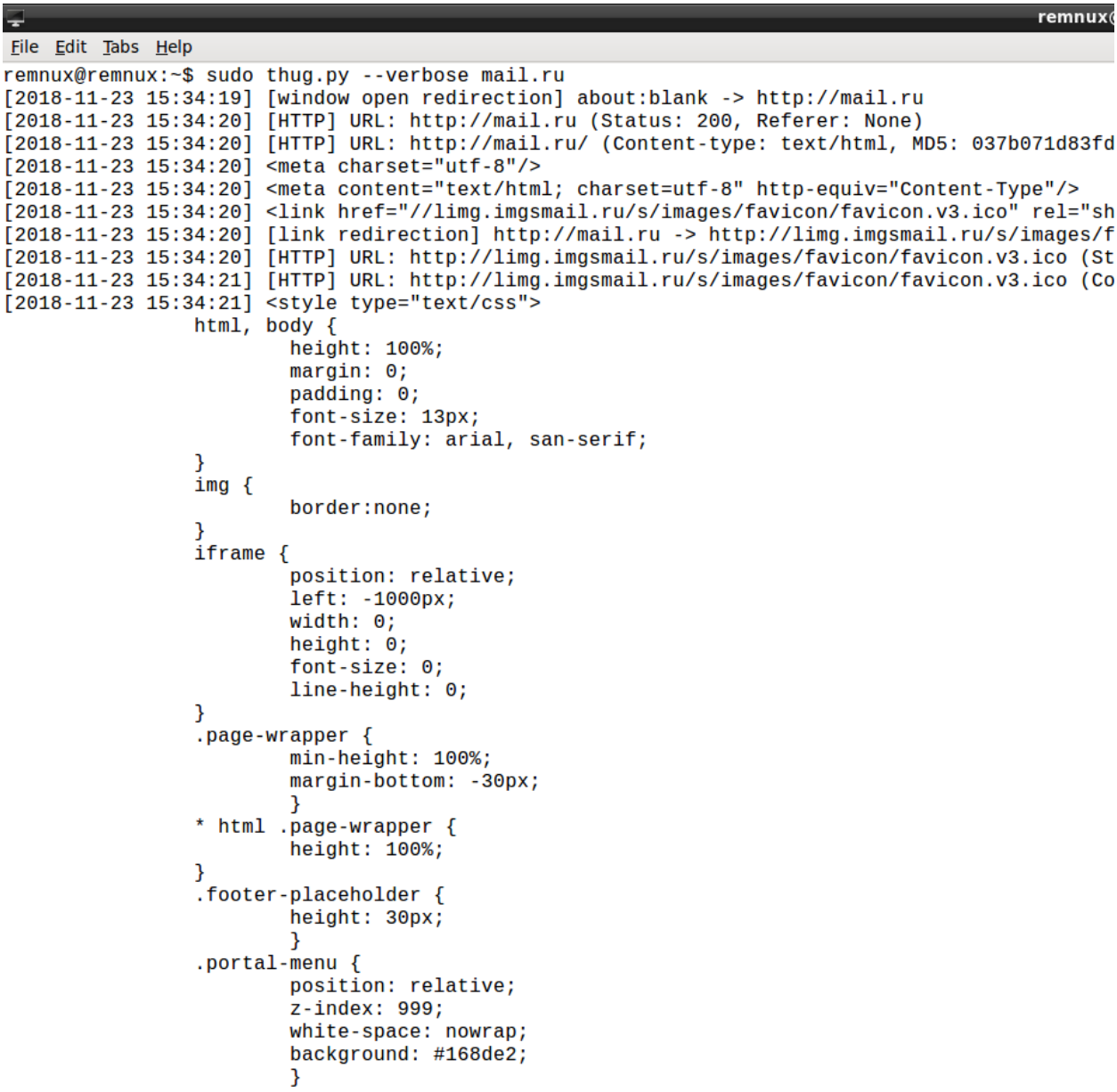
Most of the results indicates the PDF as an exploitation file. 61 detection software was scanned and 34 came positives. The most common word is pidief so it assumes that is the name of the malware. Most of the detections scanned as and exploitation for CVE-2008-2992.

This vulnerability is associated with Adobe util.printf() Buffer Overflow. According to NIST, (November 4, 2008) “Stack-based buffer overflow in Adobe Acrobat and Reader 8.1.2 and earlier allows remote attackers to execute arbitrary code via a PDF file that calls the util.printf JavaScript function with a crafted format string argument, a related issue to CVE-2008-1104.” The vulnerability score is 9.3 which is a high impact if is executed. As a Forensic Analyst, being able to identify this vulnerability is crucial so you can notify the client and make sure patching is in place immediately.

Thug.py

The last tool I want to demonstrate is the thug.py command. This tool is used to analyze suspicious websites and investigate lines of codes to identify signs of malware. You may analyze any website you want if you are curious to see a specific website that is strange. I’ll be analyzing a safe website, but it communicates with malware. The domain is called Mail.ru which is hosted in Russia.

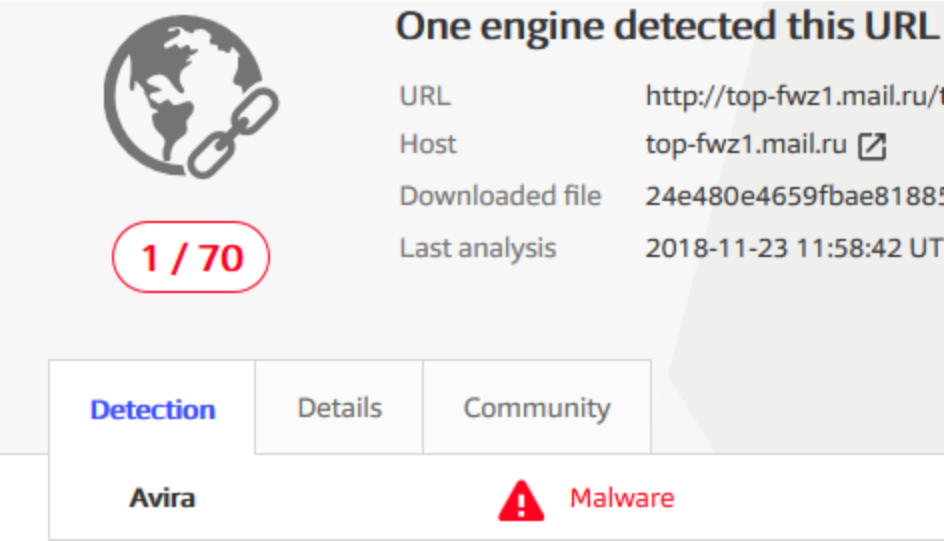

MYIP.MS
Hosting Info, Websites & IP Database
IPv6 WHOIS Support
Login | Register
mail.ru
Whois Lookup
Products
Hosting Companies
Websites
Blacklist / IP Database
Interesting
Sitemap
API
Whois Web Hosting Information for website – mail.ru –
23 November 2018, 15:32:35
Hosting Info for Website: mail.ru
#35 position in world sites rating
Hide Map »
Popularity: 13,000,000 visitors per day
IP Address: 94.100.180.200
Linked IPv6 Address: IPv6 2a00:1148:db00:0:b0b0::1
IP Location: Russia
IP Reverse DNS (Host): mail.ru
Top Level Host Usage: 2,562 sites use XXX.mail.ru as IP Reverse DNS
Hosting Company / IP Owner: @mail.ru group Mail.ru Llc
Owner IP Range: 94.100.176.0 - 94.100.183.255 (2,048 ip) Other Sites on IP »
Owner Address: Leningradskiy Prospect, 47, Build 2, 125167 Moscow Russia
Owner Country: Russia
Owner Phone: +7 495 7256357
Owner Website: corp.mail.ru
Owner CIDR: 94.100.176.0/21
Whois Record Created: 29 Nov 2010
Website Live Diagram
CLICK HERE >>>
Alexa
35
40
45
50
55
Jan '18
Apr '18
Jul '18
Oct '18
See also:



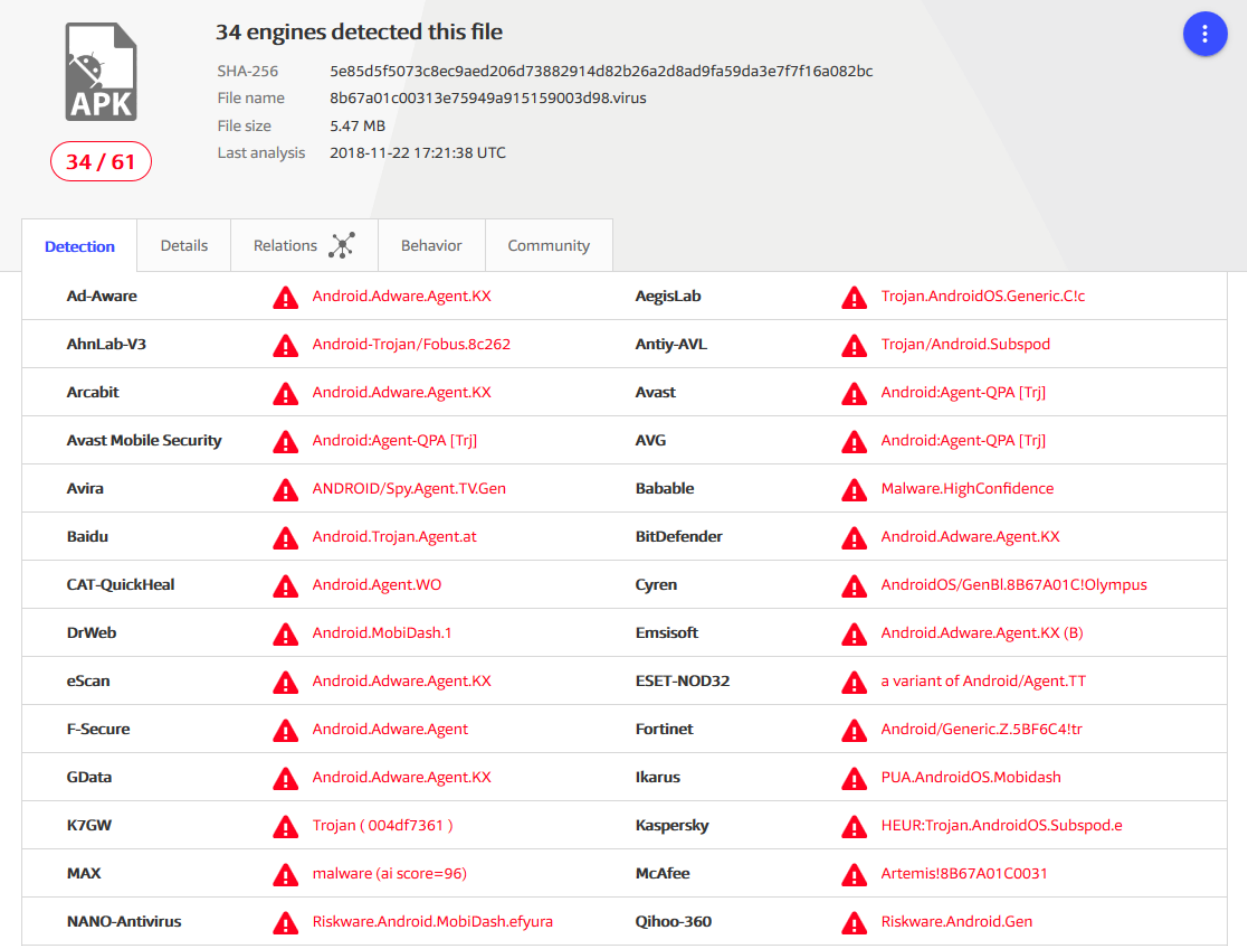
This command outputs a long line of code. From there you can start analyzing the domain. As a Forensic Analyst, you would go through each line of code and check for any signs of malware. If you see something odd or malformed, you may want to copy the code and debug it on a programming software. This way you can run the code safely. After analyzing several lines of code, I found a suspicious domain for top-fwz1[.]mail[.]ru

```
    ts.src = (d.location.protocol == "https:" ? "https:" : "http:") + "//top-fwz1.mail.ru/js/code.js";
    var f = function() {
        var s = d.getElementsByTagName("script")[0];
        s.parentNode.insertBefore(ts, s);
    };
    if (w.opera == "[object Opera]") {
        d.addEventListener("DOMContentLoaded", f, false);
    } else {
        f();
    }
})(document, window, "topmailru-code"); < /script>
[2018-11-23 15:34:23] < script language = "javascript"
type = "text/javascript" >
    //<![CDATA[
    new Image().src = "//counter.yadro.ru/hit;mail-splash/pc?r" +
    escape(document.referrer) + ((typeof(screen) == "undefined") ? "" :
        ";s" + screen.width + "*" + screen.height + "*" + (screen.colorDepth ?
            screen.colorDepth : screen.pixelDepth)) + ";u" + escape(document.URL) +
    ";" + Math.random();
//]]>
< /script>
remnux@remnux:~$
```

I searched the domain in VirusTotal and it seems to have 1 indication of malware.



Sometimes is not good to rely on the URL and I suggest analyzing what the URL has been communicating with. The first communicating file is an Android APK which indicates a sign of Adware.



This APK file indicates that is communicating with the URL I was analyzing from Mail.ru. Thug.py can be useful as a Forensic Analyst if an incident was targeting to a compromised website. Analyzing compromised websites are safe in a URL scanner. Is not recommended to enter the URL in a work environment unless if you test it out in a separate virtual machine.

Conclusion

Using REMnux as DF/IR is essential for any incident that may involve with malware. REMnux has a big list of commands that I didn't demonstrate, and they are useful when you need to perform malware analysis in a safe environment. REMnux can be useful in a Security Operation Center (SOC) for critical security incidents. This toolkit would fit into the Incident Response (IR) lifecycle under Containment, Eradication & Recovery phase. I like that REMnux is a package of useful tools all in one virtual machine. From there, you can configure another VM host-only adapter that talks to REMnux, but no communication outside to the Internet. There is a lot of documentation of how to use each of the tools which are very helpful when you are learning malware analysis for the first time. The beauty of this toolkit is open source, so anyone can start today learning about malware analysis and practice in your home lab.

References:

Sinn (October 30, 2017) How to Create a Malware Analysis Lab – VirtualBox. *YouTube*

Rochniak, D. (December 9, 2016) Remnux. *YouTube*

(November 2018) What is Gstatic? (Remove Gstatic Virus) Nov. 2018 Update. *HowToRemove*

MITRE (November 4, 2008) CVE-2008-2992 Detail. *NIST*

AnalyzePDF.py. *GitHub*

Remnux.org

Myip.ms

REMnux Docs

VirusTotal