International Journal of Engineering and Advanced Technology (IJEAT)
ISSN: 2249 – 8958 (Online), Volume-9 Issue-5, June 2020

Potentially Unwanted Program Analysis and Detection using YARA Rules

V.Shanmugavel , S.Aanantha Sankar, A.Pravin Kumar, M.Satheeshkumar, S.Malathi

Abstract: ***In this paper, some potentially unwanted program (PUP) samples are analyzed, detected and are blocked using YARA rules. Nowadays the user may notices the unwanted software such as PUP or a potentially unwanted application (PUA) . For security and parental control products subjective tagging criterion was used. To compromise privacy or weaken the computer's security such software was implemented. Third party software often bundle a wanted program to be downloaded with a wrapper application and may offer to install an unwanted application. In this paper, some samples of PUP under reverse engineering technique are analyzed by using YARA rules that promptly resist unwanted applications or programs.***

Keywords: PUP(Potentially unwanted software),YARA rule,PUA(Potentially unwanted application), Adware.

I. INTRODUCTION

Nowadays the user may notices the unwanted software such as PUP or a potentially unwanted application (PUA) .For security and parental control products subjective tagging criterion was used. To compromise privacy or weaken the computer's security such software was implemented. Companies often bundle a wanted program to be downloaded with a wrapper application and may offer installing an unwanted application, and in sometimes without providing a clear confined method. The software bundled as potentially unwanted programs defined by antivirus companies includes adware which is a software that displays intrusive advertising , or spyware which is used to track the user's Internet usage to sell information to advertisers , inserted webpages into own advertising that a user perceives, or uses premium SMS services to rack up charges for the use. Unethical consideration of this practice violates the security interests of users without their knowledge. In this paper, these spyware and adware PUP's are discussed in an efficient manner. The strings of particular PUP application in a closed environment are collected by reverse engineering techniques and are defined using YARA rules. While comparing the obtained results with existing detection methodologies, YARA rule gives good efficiency. Also, by giving the appropriate collective strings as input in YARA rule it overcomes the false positivity.

Revised Manuscript Received on June 30, 2020.
*Corresponding Author
V.Shanmugavel, Electronics and Communication Engineering, National Engineering College, Kovilpatti, Tamilnadu Email:shanmugavel1709@gmail.com
S.AananthaSankar, Electronics and Communication Engineering, National Engineering College Kovilpatti, Tamilnadu Email: aananthasankar33@gmail.com
A.PravinKumar, Electronics and Communication Engineering, National Engineering College, Kovilpatti, Tamilnadu Email:pravinking2015@gmail.com
M.Satheeshkumar, Electronics and Communication Engineering, National Engineering College, Kovilpatti, Tamilnadu Email:satheeshnecece@gmail.com
S.Malathi*, Electronics and Communication Engineering, National Engineering College, Kovilpatti, Tamilnadu Email:malathi.nec@gmail.com

© The Authors. Published by Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP). This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/)

II. POTENTIALLY UNWANTED PROGRAM (PUP)

A PUP, is a program that may be unwanted, despite the fact that a user consented to download it. The users can download PUPS that incudes spyware, adware, or additional web browser toolbars, along with a program that the user actually needs. It is a software that we may or may not want clogging up our system. Users might be found some problems in PUP as similar to malware in that they cause problems when downloaded and installed. But PUPs are downloaded with our consent. Huge amounts of system resources and are a common cause of clunky operating systems, but are not considered malicious or harmful were employed by PUP sometimes. However, they are often annoying, creating new toolbars in our web browser for shopping sites, changing our search provider from Google to Bing without reason, popping up ads constantly or giving regular weather updates from Swaziland. Some are even aggressive by intentionally slowing down our computer to later sell system-tuning or miracle speedup tools.

III. PUP DIFFERS FROM MALWARE

The main difference is the method in which the software is distributed. Malware is malicious software that is installed without one's permission. PUPs are programs that piggy back onto an End User License Agreement (EULA) and trick users into installing them. The distribution of PUPs tagging along with free software downloads were downloaded is a worthwhile market. And it is all completely legal, because fine print are accepted and given permission for the company to run the PUP program on the user's computer. In this paper, spyware and adware PUP's were analyzed by the collection of the strings of particular PUP application in a closed environment by reverse engineering techniques and are defined, detected and defended using YARA signatures.

IV. SPYWARE

Spyware is unwanted package that infiltrates data processor, stealing web usage knowledge and sensitive info.

Spyware is classed as a kind of malware malicious package designed to realize access to or injury laptop, usually while not the user's data. Spyware gathers user's personal info and relays it to advertisers, data firms, or external users. Spyware is employed for several functions. Typically it aims to trace and sell user's web usage knowledge, capture user's mastercard or checking account info, or steal user's personality. Spyware monitors user's web activity, pursuit user's login and arcanum info, and spying on user's sensitive info. Some sorts of spyware will install extra package and alter the settings on user's device, thus it's necessary to use secure passwords and update devices at regular amount. Spyware contributed to those numbers. the foremost common threats on the web is spyware. It will simply infect user's device and it are often arduous to spot. Spyware could be a threat to businesses and individual users, since it will steal sensitive info and damage the network.

V. SPYWARE IN EULA FORM

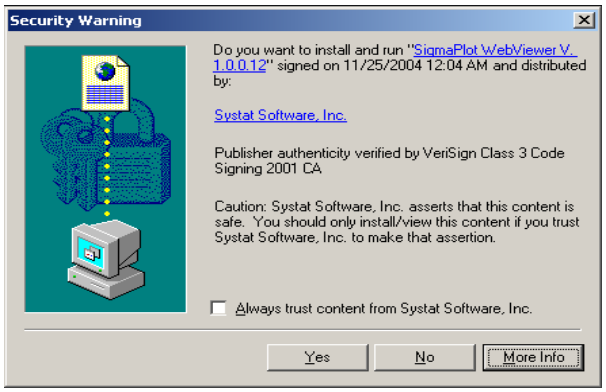

FIG 1: BLOCK DIAGRAM

In the above figure 1 there's a EULA to permit that individual package to form changes within the system or not. Here the wrongdoer bundled the malicious things with this and build itself appear as if a legitimate package. By this manner conjointly a spyware will enter into our system and observation our moves within the system and collect some precious info like master card details, necessary files, personal details etc. These all are kept away from the user data.

Spyware will have an effect on PCs, Macs, and IOS or golem devices. Though Windows operative systems is also additional vulnerable to attacks, attackers are getting higher at infiltrating Apple's operative systems further. a number of the foremost common ways that your laptop will become infected with spyware embrace these:

- Accepting a prompt or pop-up while not reading it 1st
- Downloading package from Associate in Nursing unreliable supply
- Opening email attachments from unknown senders
- Pirating media like movies, music, or games.

VI. REVERSE ENGINEERING PROCESS

Reverse engineering of PUP involves disassembling a package program. Through this method, binary directions were regenerate to code method. In order that we will explore what the program will and what systems it impacts. Solemnly by knowing its details we are able to produce solutions that may mitigate the program's supposed malicious effects. The reverse engineering method use a spread of tools to seek out however a program is propagating through a system and what it's built to try to to. And in doing thus, the reverser would then apprehend that vulnerabilities the program was assuming to exploit. The most issue of reverse engineering is to extract hints revealing once a program was created, what embedded resources they will be exploitation, secret writing keys, and different file, header, and data details.

VII. YARA RULE

The most best language and its syntax virtually resembles the C language is YARA rule. every rule consists of keyword rule and a rule symbol. Identifiers should follow constant lexical conventions of the C programing language, they will contain any alphanumeric character and also the underscore character, however the primary character can not be a digit. Some keywords square measure reserved and can't be used as AN symbol. Rules square measure usually composed of 2 sections known as strings definition and condition. The strings definition section is omitted if the rule does not accept any string, however the condition section is often needed. Strings is outlined in text or hex type. Text strings square measure fenced in quote rather like within the C language. within the on top of rule, there square measure 3 sections meta, strings and condition. In meta section, it consists of the detail regarding author United Nations agency created that rule, date that the rule has been created and conjointly the changed date. consequent section comprises strings in this specific file, this is often the most section of the rule. It solely decides what to discover within the specific intrude files. the ultimate half is that the condition section that consists of what's the mandatory action to be taken whereas detection, whether or not to permit it or block it.

```
rule rogues
{
    meta:
        created  = "09/11/2017 00:00:00"
        modified = "09/11/2017 17:12:07"
        author   = "Metallica"
        Vendor   = "PC Smart Cleanup"
    strings:
        $textstring1 = "smart" ascii wide nocase
        $textstring2 = "cleanup" ascii wide nocase
        $textstring3 = "longrun" ascii wide nocase
    condition:
        $textstring1 and @textstring2 and $textstring3
}
```

FIG 2: SAMPLE RULE

In the above figure2, there are three sections named as meta, strings and condition. In meta,it contains the details about author name and when it was created and modified etc..after that there is a string section in that section, the strings of that particular exe that we want to detect were defined under this section, there are some keywords also available along with that only the string section can be defined and modified. In the last section ,we have perform the action how a rule going to detect those files, it is a main section to perform the action against those exe called condition section.

VIII. PROPOSED ALGORITHM

The below block diagram denotes how the potentially unwanted applications were collected and each of the samples would go under the static analysis process. In that process the strings were collected.

Retrieval Number: E9855069520/2020©BEIESP
DOI: 10.35940/ijeat.E9855.069520
Journal Website: www.ijeat.org

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication
© Copyright: All rights reserved

These strings are compared with the legitimate file strings to avoid the false positive. This means both the PUP files and the normal software has some common strings and characteristics. To distinguish these two by static analysis method and to overcome the false positives. That means then only it detects the appropriate PUP programs.

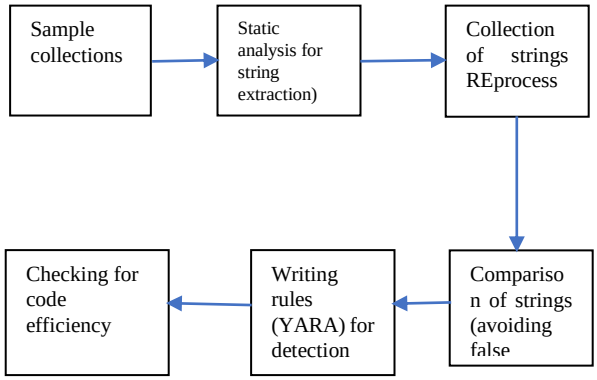

FIG 3: BLOCK DIAGRAM

From the information, collected from the samples, this rule has been created using yara,

```
import yaramodule
rule sample3
{
    strings:
        $sample                                    =
"5x6B`r@5x6Blq@9-x6B`r@_^][l6BDr@<r@Dr@x6BDr@T
p@Pp@
                <p@Dp@Hp@SVW|6B,@@s8j#,r@5Dr@,r@

5Dr@PPh@.B\q@_^[=,?BDr@v0j,r@=x6Bp4jp0jTq@tVPxr
@u

Pj@.Bt530q@|6Bt+PjD?B_^[5Dr@Pq@@SVRVh5x6B_^

[5Dr@pq@PSV5,r@5Dr@Pj@hTN@0r@j!j4p@(p@WWhDp
@PWhCSPhQ0r@`r@uv9Euh3"  wide ascii no case
    condition:
        sample
}

rule sample4
{
    strings:
        $sample_1=  "h+AP+A8+AP+A8+AP+A8+A8+A<,A
                H,AxRAp,AxRA                    -A8-Ah-A<
@(.A@.Au"@q$@D%@p.AP.A

p.AP.A/A`/A,&@p.AP.A@0Ah0A(1AP1AH2Ap2A3AH3Ap3A
                4AH4Ap4Ah5A86A`6A" wide ascii no case
    condition:
        sample_1
}
rule sample1
{
condition:
hash.md5("sample")=="8b2104e5a7e1f7fb65bab004e238bd09"
hash.SHA1(0,277.66)=="2a9495d0fc7880acbf0e2e1206b9
b85844ece5b"
hash.md5("sample1")=="ab8a97dbebe9616466037da6c3
1ae49fd"
hash.SHA1(0,526.00)=="51cd1170bbfc42d58a492ef140b
a8ffc767f5a9"
}
```

Efficiency

IX. ANALYSIS AND DETECTION

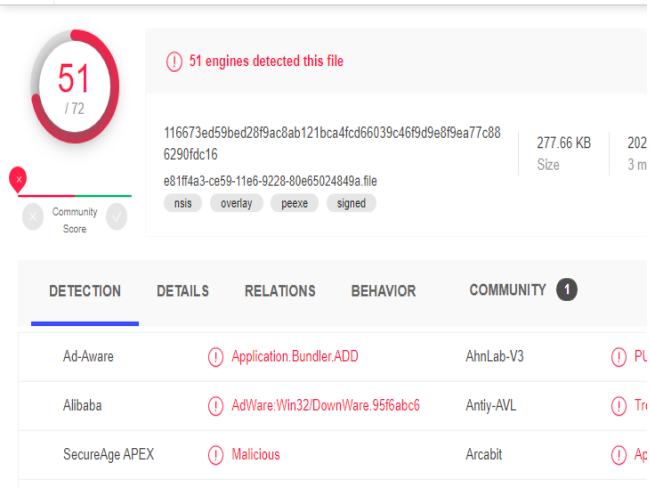

FIG 4: under virustotal environment

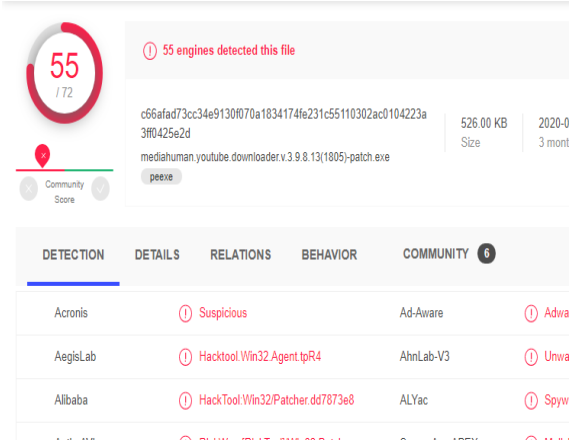

FIG 5: under virustotal environment

In that surrounding, there is lot of antivirus engines and its detections were stored as a database. Here in this, it clearly shows that out of 72 antivirus engines only 55 were detected that PUP sample which has showed in the Fig 4 and the remaining engines were detected it as a legitimate file. In this place only the efficiency came into the picture, at first for the given hash it checks for the signature or rule that matched with that file. If it matches, means it allows or otherwise it will detect it as malicious or unwanted program. Same as we take the Fig 8.2 out of 72 it was detected by 51 engines. So when compare these results, came to know that how much efficient a code must detect the unwanted application. The false positive rate also considered. It should be in low percent while compare to other factors. 100 samples of legitimate software and 100 samples of unwanted programs have taken to check the efficiency of the rule and also the false positive rate.

© Copyright: All rights reserved.

X. EVALUATION RESULTS

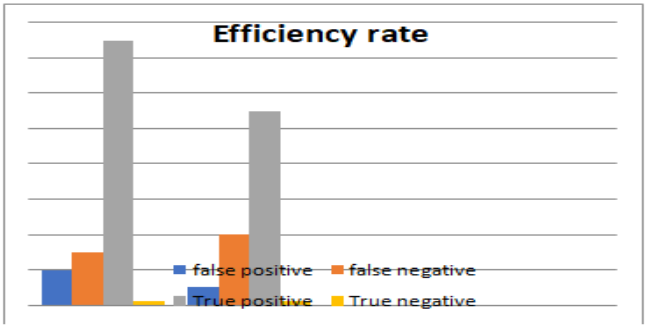

FIG 6: evaluation results

Accuracy (%) = (TP + TN / TM + TB) * 100
Where,
TP – True positive
TN- True negative
TM- Total no of malware
TB- Total no of benign malware

Types	Total no of samples	Detected samples	Undetected samples
Adware	40	33	7
spyware	30	25	5
Legitimate file	20	2	18

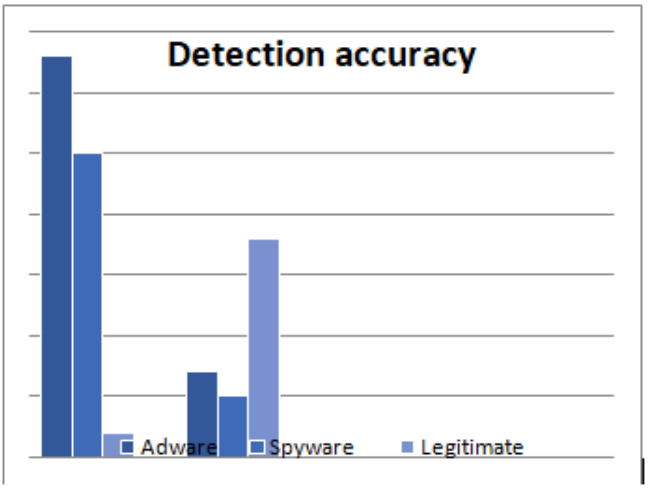

FIG 7: detection accuracy

XI. CONCLUSION

There are lot of search engines and Antivirus(AVs) were available to detect lot of malwares and PUPs, In windows operating system there is inbuilt detection called windows defender. It also checks for some malicious applications when we download it. But the thing is how much efficiency it check for the strings, some of the engines were fail to detect some potentially unwanted application and some others are detect it as false positives. In this paper PUP using RE techniques, with huge information about that file were analysed. Generally there are lot of information appended with a malicious application, in some cases it is difficult to judge whether the software is legitimate or malicious. For that purpose its need a close environment like virtual machine, this is a kind of precaution to safe our host machine from the malicious applications. This work concludes that adware samples with the help of reverse engineering techniques analysed and detected it by writing YARA rules with good efficiency.

REFERENCES

1. Pankaj Kumar and Ashok Madan, Review Paper on Reverse Engineering, Journal of Basic and Applied Engineering Research, Volume 2, Number 16; April-June, 2015 pp. 1377-1380.
2. M.G.Rekoff, "On reverse engineering," IEEE Transactions on Systems, Man, and Cybernetics, vol. SMC-15, no. 2, March-April 1985, pp. 244-252.
3. Raut, Laukik & Barai, Gayatri & Shete, Sunil " Design and Development of a component by reverse engineering".International Journal of Research in Engineering and Technology. 4,2015 539-546.
4. R.Pescaru,P.Kyratsis ,G.Oancea," A Case Study of Reverse Engineering Integrated in an Automated Design Process", IOP Conference Series: Materials Science and Engineering, Volume 161, 2016.
5. R.Dhipakumar, Dr. R.Ganesh, Reverse Engineering in Mechanical Component,IJERTCONV7IS11103,Volume 7,18-12-2019.
6. Ray, Anusmita & Nath, Asoke. International Journal of Advance Research in Computer Science and Management Studies Introduction to Malware and Malware Analysis: IJARCSMS. Volume 4. 22-30, (2016).
7. Samanvay Gupta," Types of Malware and its Analysis", International Journal of Scientific & Engineering Research Volume 4, Issue 1, January-2013
8. https://cybersecurity.att.com/blogs/labs-research/reverse-engineering-malware[online]
9. https://www.sans.org/course/reverse-engineering-malware-malware-analysis-tools-techniques[online]
10. https://www.researchgate.net/publication/329836170_Malware_Analysis_and_Detection_Using_Reverse_Engineering_Technique[Pdf]
11. https://www.lastline.com/blog/reverse-engineering-malware/[online]
12. https://us.norton.com/internetsecurity-how-to-catch-spyware-before-it-snags-you.html [online].
13. https://www.malwarebytes.com/adware/[online]
14. https://www.virussign.com/?gclid=EAIaIQobChMIgOuuwfzs5wIVBBePCh01kQm5EAMYASAAEgLJXvD_BwE[online]
15. https://link.springer.com/chapter/10.1007%2F978-3-642-04474-8_17[Pdf]
16. Spyware.Look2Me, http://securityresponse.symantec.com/avcenter/venc/data/spyware.look2me.html[online]
17. Adware.VirtuMonde,http://securityresponse.symantec.com/avcenter/venc/data/adware.virtumonde.html [online].
18. https://scholarsmine.mst.edu/cgi/viewcontent.cgi?article=6007&context=masters_theses[pdf]
19. https://hackercombat.com/static-malware-analysis-vs-dynamic-malware-analysis/[online].
20. http://www.differencebetween.net/technology/difference-between-static-malware-analysis-and-dynamic-malware-analysis/[online]
21. https://www.csoonline.com/article/2883832/virustotal-tackles-false-positive-malware-detections-plaguing-antivirus-and-software-vendors.html.

AUTHORS PROFILE

V.SHANMUGAVEL, Under Graduate Student Electronics and Communication Engineering, National Engineering College, Kovilpatti, Tamil Nadu- 628 503. His research interest includes cybersecurity.

© Copyright: All rights reserved

S.AANANTHASANKAR, Under Graduate Student Electronics and Communication Engineering, National Engineering College, Kovilpatti, Tamil Nadu- 628 503. His research interest includes cybersecurity.

A.PRAVINKUMAR, Under Graduate Student Electronics and Communication Engineering, National Engineering College, Kovilpatti, Tamil Nadu- 628 503. HIS research interest includes cybersecurity.

M.SATHEESHKUMAR, received M.E degree in Electronics and Communication Engineering. He is currently working in National Engineering College, Kovilpatti, Tamil Nadu- 628 503. as Assistant Professor. His research interest includes Networking and Cybersecurity.

S.MALATHI, received M.E degree in Electronics and Communication Engineering. She is currently working in National Engineering College, Kovilpatti, Tamil Nadu- 628 503. as Assistant Professor. Her research interest includes Optical Communication and Image Processing.

© Copyright: All rights reserved.