

מסמך איפיון למיני פרוייקט - נושאים במערכות הגנה

לרשת

מודול לשרת פרוקסי לחסימת קבצים עוינים

מגשים:

אברהם נתן – ת.ז 317306736

ארז כרמל – ת.ז 302491600

ספיר גאלי – ת.ז 307907436

מסמך אפיון – מיני פרוייקט בנושא מערכות הגנה ברשת

במסמך זה יינתן תיאור מפורט של הפרוייקט – מודול שמהווה תוספת לשרת אפאצ'י 2 במוד פרוקסי לצורך הגנה מפני קבצים עוינים.

המסמך יציג עקרונות מנחים, אופן פעולת המערכת, התקנה וקינפוג ודוגמאות לפעולת המערכת.

מבוא

בארגונים רבים כיום קיימת רשת ארוגנית שעל בסיסה מתבצעת תעבורת המידע ואחסון המידע בתוך הארגון וגם כלפי חוץ. למרות שמבחינת שמירה על בטיחות המידע, הפתרון האידיאלי הוא ניתוק לחלוטין של הרשת הארגונית מרשת האינטרנט, על מנת שהארגון יוכל להתנהל בצורה מיטבית עליו לאפשר חיבור לעולם החיצון. עצם חיבור זה יוצר נקודת תורפה ופותחת פתח לנוזקות אפשריות למצוא את דרכן לתוך הרשת הארגונית.

לצורך מתן מענה לסכנות כאלה יש צורך בבקרה על תעבורת המידע שמגיע אל או מתוך הרשת בכלל זה סינון קבצים שניתן להוריד מהרשת האינטרנט לרשת הארגונית, ובכך למנוע הורדה של קבצים עוינים, ניתורם, והתראות על ביצוע ניסיונות כאלה.

אופן פעולה:

אפאצ'י 2 הינו שרת ורסטילי וסקלבילי שמשתמש בכלים הנקראים מודולים לצורך הגדרת תצורות עבודה שונות. מודולים אלה הינם קבצים מהצורה `some_module.so` המגיעים עם החבילה של אפאצ'י, שניתן לטעון לתוך השרת לפני או בזמן עבודתו, ומופעלים ברגע שהשרת הופעל או מאותחל מחדש. בנוסף למודולים שמגיעים עם החבילה של אפאצ'י, ישנה אפשרות ליצור מודולים מותאמים אישית ולטעון אותם לשרת. זה האופן שבו מימשנו את הכלי.

ניטור:

הכלי יעבוד עם רשימת שמות הקבצים שמוגדרים כעוינים כפי שמופיעים ברשת, וישווה כל בקשת הורדה עם רשימה זו. במידה והקובץ שבבקשת ההורדה תואם ל"רשימה שחורה" זו, תיחסם ההורדה.

תגובה:

במקרה של תפיסת בקשה להורדה של קובץ עוין המודול יתריע זאת בפני המשתמש, יפנה אותו לאתר בטוח, ויבצע תיעוד של המקרה.

התקנה וקינפוג:

דרישות קדם:

1. מערכת לינוקס שעליה מותקן אפאצ'י 2.
2. הכלי apxs מותקן על המערכת.

התקנה:

פקודות ההתקנה מפורטות בקובץ Installation Notes.txt ובנוסף מצורף הקובץ installer.sh אשר ניתן להרצה דרך הcommand line ומבצע את כל ההתקנות הנדרשות באופן אוטומטי.

קינפוג:

בדפדפן יש להגדיר את תצורות הגישה של שרת ה Proxy – להגדרות proxy ידניות, יש להגדיר את הקו של שרת הפרוקסי וport 8080, עבור הפרוטוקולים http, ftp, ssl.

בשרת הפרוקסי יש להגדיר את השמות של הקבצים העוינים – בקובץ /var/www/html/BlackList.txt יש להזין את שמות הקבצים שנרצה לחסום הורדה שלהם.

מימוש התוכנית:

הקובץ המרכזי נכתב בשפת C ומכיל את הקוד של המודול ונוספים אליו 6 קבצי הגדרות:

- proxy.conf
- ports.conf
- forward_proxy.conf
- virus_block.conf
- mod_virus_block.c
- BlackList.txt
- virus_block_log.txt

נפרט כעת על כל אחד מהקבצים.

:proxy.conf

קובץ הקונפיגורציות שמאפשר את מודול הפרוקסי של אפאצ'י. ברירת המחדל בקובץ זה הוא לא להעביר אף בקשה דרך הפרוקסי.

```
proxy.conf
1 <IfModule mod_proxy.c>
2
3     # If you want to use apache2 as a forward proxy, uncomment the
4     # 'ProxyRequests On' line and the <Proxy *> block below.
5     # WARNING: Be careful to restrict access inside the <Proxy *> block.
6     # Open proxy servers are dangerous both to your network and to the
7     # Internet at large.
8     #
9     # If you only want to use apache2 as a reverse proxy/gateway in
10    # front of some web application server, you DON'T need
11    # 'ProxyRequests On'.
12
13    ProxyRequests On
14    <Proxy *>
15        AddDefaultCharset off
16        Require all denied
17        Require local
18    </Proxy>
19
20    # Enable/disable the handling of HTTP/1.1 "Via:" headers.
21    # ("Full" adds the server version; "Block" removes all outgoing Via: headers)
22    # Set to one of: Off | On | Full | Block
23    #ProxyVia Off
24
25 </IfModule>
26
27 # vim: syntax=apache ts=4 sw=4 sts=4 sr noet
28
```

:ports.conf

בקובץ זה מוגדר לשרת הפרוקסי להקשיב לפורט 8080.

```
ports.conf
1 # If you just change the port or add more ports here, you will likely also
2 # have to change the VirtualHost statement in
3 # /etc/apache2/sites-enabled/000-default.conf
4
5 Listen 80
6 Listen 8080
7
8 <IfModule ssl_module>
9     Listen 443
10 </IfModule>
11
12 <IfModule mod_gnutls.c>
13     Listen 443
14 </IfModule>
15
16 # vim: syntax=apache ts=4 sw=4 sts=4 sr noet
17
```

:forward_proxy.conf

קובץ הגדרות למודול הפרוקסי עבור תצורת עבודה של forward_proxy – פרוקסי שמסתיר את הרשת הארגונית מהרשת הגלובלית (local proxy). מקובץ זה מוגדר שכל url שמקורו ברשת הארגונית (ip שמתחיל ב 192.168) יעבור דרך שרת הפרוקסי. מוגדר בשורה 1 שמודול פרוקסי זה יקשיב לפורט 8080.

כמו כן מוגדרים תיעודי בקשות שעוברים דרך הפרוקסי (logging).

```
forward_proxy.conf
1 <VirtualHost *:8080>
2     # The ServerName directive sets the request scheme, hostname and port that
3     # the server uses to identify itself. This is used when creating
4     # redirection URLs. In the context of virtual hosts, the ServerName
5     # specifies what hostname must appear in the request's Host: header to
6     # match this virtual host. For the default virtual host (this file) this
7     # value is not decisive as it is used as a last resort host regardless.
8     # However, you must set it for any further virtual host explicitly.
9     #ServerName www.example.com
10
11     ProxyRequests On
12     ProxyVia Off
13     <Proxy " *>
14         Require ip 192.168
15     </Proxy>
16
17     # Available loglevels: trace8, ..., trace1, debug, info, notice, warn,
18     # error, crit, alert, emerg.
19     # It is also possible to configure the loglevel for particular
20     # modules, e.g.
21     #LogLevel info ssl:warn
22
23     ErrorLog ${APACHE_LOG_DIR}/error_forward_proxy.log
24     CustomLog ${APACHE_LOG_DIR}/access_forward_proxy.log combined
25     # For most configuration files from conf-available/, which are
26     # enabled or disabled at a global level, it is possible to
27     # include a line for only one particular virtual host. For example the
28     # following line enables the CGI configuration for this host only
29     # after it has been globally disabled with "a2disconf".
30     #Include conf-available/serve-cgi-bin.conf
31
32 </VirtualHost>
33 # vim: syntax=apache ts=4 sw=4 sts=4 sr noet
```

:virus_block.conf

קובץ הקונפיגורציות עבור הכלי שמסנן את הקבצים. בקובץ זה מוגדר לאפאצ'י לשלוח בקשות שעוברות דרכו ל handler של הכלי (virus_block_module) במידה וכלי זה מופעל.

```
virus_block.conf x
1 <IfModule virus_block_module>
2
3     SetHandler virus_block
4
5 </IfModule>
6
```

:mod_virus_block.c

קובץ קוד המקור של הכלי. מורכב מ 4 הפונקציות הבאות ומהרשימה הבאה:

```
12
13 /* Define prototypes of our functions in this module */
14 static void register_hooks(apr_pool_t *pool);
15 static int virus_block_handler(request_rec *r);
16 static char* fileNameFromURL(char* str);
17 static char* initBlackList();
18
19
20
21 /* Define the black list */
22 static char *blackList;
23
```

blackList שבשורה 22 תהיה רשימת הקבצים העוינים אשר אנחנו נקרא מקובץ בשם BlackList.txt ואליה נשווה שמות של קבצים.

הפונקציה register_hooks מפעילה את ה handler שמטפל בבקשות שעוברות דרך הפרוקסי ומפעילה את הפונקציה שמאתחלת את ה blackList.

```
66
67 /* The function that registers our handler */
68 static void register_hooks(apr_pool_t *pool)
69 {
70     blackList = initBlackList();
71     ap_hook_handler(virus_block_handler, NULL, NULL, APR_HOOK_REALLY_FIRST);
72 }
73
```

הפרמטר APR_HOOK_REALLY_FIRST נועד לוודא שמודול זה יהיה הראשון שיעבוד על בקשת ה GET של דפדפן המשתמש.

הפונקציה virus_block_handler מבצעת את הדברים הבאים:

1. פתיחה של קובץ לוג ששם יתועדו הבקשות שעוברות דרך שרת הפרוקסי.
2. בודקת שאכן קיים url לבקשה (בקשות מסוגים מסויימים לא מכילות url).
3. מחפשת את שם הקובץ בurl ברשימה השחורה.
4. במידה ונמצאה התאמה, כלומר הקובץ מוגדר כעוין, המערכת מתריעה בפני המשתמש ונותנת את האפשרות לעבור לאתר בטוח.
5. בנוסף מבצעת תיעוד של ניסיון ההורדה של הקובץ העוין.

```
26 /* Our handler function. receives a request record r, handles it. */
27 static int virus_block_handler(request_rec *r)
28 {
29     if (!r->handler || strcmp(r->handler, "virus_block")) return(DECLINED);
30     FILE *fp;
31     fp = fopen("/var/www/html/virus_block_log.txt", "a+");
32     if (fp == NULL) {
33         // do nothing
34     } else {
35         fprintf(fp, "A click:\n");
36         fprintf(fp, "\tREQUEST RECORD STRUCT (PARTIAL):\n");
37         fprintf(fp, "\t\tthe request: %s\n", r->the_request);
38         fprintf(fp, "\t\tprotocol: %s\n", r->protocol);
39         fprintf(fp, "\t\thostname: %s\n", r->hostname);
40         fprintf(fp, "\t\trequest time: %d\n", r->request_time);
41         fprintf(fp, "\t\tstatus: %d\n", r->status);
42         fprintf(fp, "\t\tmethod: %s\n", r->method);
43         fprintf(fp, "\t\tcontent type: %s\n", r->content_type);
44         fprintf(fp, "\t\tthandler: %s\n", r->handler);
45         fprintf(fp, "\t\tunparsed uri: %s\n", r->unparsed_uri);
46         fprintf(fp, "\t\turi: %s\n", r->uri);
47         fprintf(fp, "\t\tfilename: %s\n", r->filename);
48         fprintf(fp, "\t\tuseragent ip: %s\n", r->useragent_ip);
49     }
50
51     if ( (fileNameFromURL(r->uri) != NULL) && (strstr(blackList, fileNameFromURL(r->uri)) != NULL) ) {
52         ap_set_content_type(r, "text/html");
53         ap_rprintf(r, "<HTML><HEAD><TITLE>Virus Download Detected</TITLE></HEAD><BODY><H1>This file is a virus. please click on the following link to be
54         fprintf(fp, "\t\t-----VIRUS DETECTED!-----\n\n");
55         fclose(fp);
56         return (DONE);
57     } else {
58         fprintf(fp, "\n");
59         fclose(fp);
60         return (DECLINED);
61     }
62 }
```

הפונקציה fileNameFromURL מקבלת url ומוציאה מתוכו את שם הקובץ שאותו נרצה לבדוק. אם לא קיים קובץ אז מוחזר NULL ואין מה לבדוק.

```
89
90 /* This function receives a url and extracts the file name from it, if any.
91    If there isnt any file, returns NULL */
92 static char* fileNameFromURL(char* str){
93
94     char *temp = malloc(strlen(str));
95     strcpy(temp, str);
96     int slashCounter = 0;
97
98     for (int i=0; i<strlen(temp); i++){
99         if(temp[i] == '/'){
100             slashCounter++;
101         }
102     }
103
104     int i = 0;
105     char *strTokens = strtok(temp, "/");
106     char *tokensArray[slashCounter+1];
107
108     while(strTokens != NULL){
109         tokensArray[i++] = strTokens;
110         strTokens = strtok(NULL, "/");
111     }
112
113     return tokensArray[slashCounter-1];
114 }
115
```

הפונקציה `initBlackList` קוראת את הקובץ עם רשימת הקבצים העוינים ובונה מחרוזת שבה נחפש את שמות הקבצים.

במידה שהתווספו שמות של קבצים עוינים, צריך רק להוסיף אותם לקובץ ולקרוא אותו מחדש.

```
117
118 /* Initializes the virus list from a file */
119 static char* initBlackList(){
120
121     FILE *BlackList;
122     BlackList = fopen("/var/www/html/BlackList.txt", "r+");
123
124     fseek(BlackList, 0, SEEK_END);
125     long fsize = ftell(BlackList);
126     fseek(BlackList, 0, SEEK_SET); //same as rewind(f);
127
128     char *blackList = malloc(fsize + 1);
129     size_t numberOfItemsRead = fread(blackList, fsize, 1, BlackList);
130     fclose(BlackList);
131
132     return blackList;
133 }
```

כמו – כן ישנו המבנה שמגדיר כי כלי זה הינו מודול בשם `virus_block` ומשייך אליו את הפונקציה `register_hooks` אשר יודעת אילו `handlers` להפעיל עבור מודול זה.

```
74
75
76 /* Define our module as an entity and assign a function for registering hooks */
77 module AP_MODULE_DECLARE_DATA virus_block_module =
78 {
79     STANDARD20_MODULE_STUFF,
80     NULL, /* Per-directory configuration handler */
81     NULL, /* Merge handler for per-directory configurations */
82     NULL, /* Per-server configuration handler */
83     NULL, /* Merge handler for per-server configurations */
84     NULL, /* Any directives we may have for httpd */
85     register_hooks /* Our hook registering function */
86 };
87
88
```

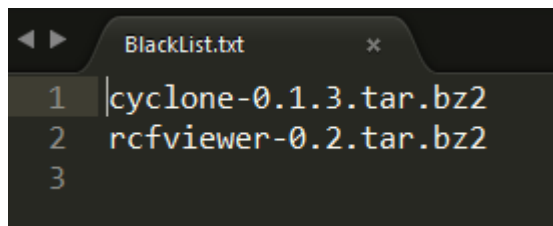

:BlackList.txt

קובץ זה הינו הקובץ אשר יכיל רשימת שמות של קבצים שהוגדרו עויינים ע"י מנהלי הרשת הארגונית (רשימה שחורה).

לתוך קובץ זה יוקלדו שמות קבצים שנמצאו עויינים ידנית, והכלי יסתמך על קובץ זה כאשר יסנן הורדות של קבצים מהאינטרנט.

מיקומו של הקובץ יהיה ב /var/www/html/

הקובץ ייראה כך:



:virus_block_log.txt

קובץ אשר יתעד את פעולת הכלי ובכלל זאת אירועים שבהם היה ניסיון להוריד קבצים עויינים. להלן דוגמה שבה הכלי תיעד ניסיון הורדה של קובץ זדוני, ולעומתו תיעוד הורדה של קובץ נקי:

(הרצה זו השתמשה ב BlackList.txt המוגדר לעיל, לכן הקובץ cyclone-0.1.3.tar.bz2 נחם)

```
242
243 A click:
244   REQUEST RECORD STRUCT (PARTIAL):
245     the_request: GET http://www.softwareclones.org/download/cyclone-0.1.3.tar.bz2 HTTP/1.1
246     protocol: HTTP/1.1
247     hostname: www.softwareclones.org
248     request_time: 1517829299910650
249     status: 200
250     method: GET
251     content_type: application/x-bzip2
252     handler: virus_block
253     unparsed_uri: http://www.softwareclones.org/download/cyclone-0.1.3.tar.bz2
254     uri: http://www.softwareclones.org/download/cyclone-0.1.3.tar.bz2
255     filename: proxy:http://www.softwareclones.org/download/cyclone-0.1.3.tar.bz2
256     useragent_ip: 192.168.2.2
257     =====VIRUS DETECTED!!=====
258
259 A click:
260   REQUEST RECORD STRUCT (PARTIAL):
261     the_request: GET http://www.softwareclones.org/download/rcf-0.3.tar.bz2 HTTP/1.1
262     protocol: HTTP/1.1
263     hostname: www.softwareclones.org
264     request_time: 1517829302326028
265     status: 200
266     method: GET
267     content_type: application/x-bzip2
268     handler: virus_block
269     unparsed_uri: http://www.softwareclones.org/download/rcf-0.3.tar.bz2
270     uri: http://www.softwareclones.org/download/rcf-0.3.tar.bz2
271     filename: proxy:http://www.softwareclones.org/download/rcf-0.3.tar.bz2
272     useragent_ip: 192.168.2.2
273
```

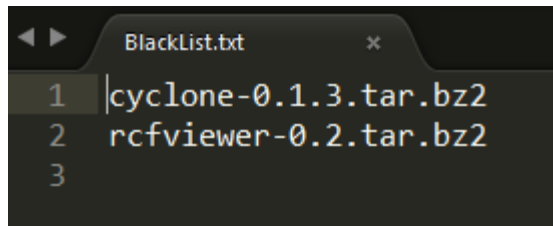
מגבלות הכלי:

הכלי מספק את הצרכים של רשת אירגונית בה המשתמשים אינם זדוניים ושבה מנהלי הרשת מסננים אתרים שנחשבים לא בטוחים. עם זאת, לכלי מספר מגבלות אשר הופכות אותו לפתרון לא אידאלי במקרה של ניסיון מכוון להוריד קבצים זדוניים. את המגבלות האלו נפרט כאן:

- קריטריון הכלי לסינון קבצים הינו שם הקובץ בלבד. דבר זה דורש עדכונים תכופים יחסית של קובץ ה `BlackList.txt` במקרים בהם שמו של קובץ זדוני שונה. בנוסף, במידה ומישהו רוצה במכוון להוריד קובץ עוין, אם הוא יוכל לגרום לכך ששם הקובץ ישונה, הוא יוכל להוריד את הקובץ. פתרון אפשרי לבעיה זו הינה הוספת קריטריונים נוספים לסינון קבצים, כגון שמירת רשימה שחורה של חתימות קבצים, אך באותה מידה ניתן לעקוף גם פתרון זה, למרות שדבר זה יהיה יותר מאתגר.
- חוסר יכולת לטפל בבקשות מוצפנות. כלי זה מתמודד היטב עם בקשות בפרוטוקול HTTP, מה שמאפשר לו להציץ לתוכן הבקשה ולשלוף משם את שם הקובץ. בבקשות מוצפנות, אין אפשרות לכלי לגשת לתוכן הבקשה, ולכן הוא לא יכול לשלוף שם של קובץ להשוואה. למשל, במידה ולקוח ינסה להוריד קובץ מפייסבוק, הכלי לא יוכל לסנן קובץ זה. פתרון לבעיה זו הינו מסובך היות והוא כרוך בפיצוח הבקשה המוצפנת ע"י זיוף certificates או ע"י הנפקה של certificates לשרת עצמו, כך שיוכל לראות תכני בקשות מוצפנות.

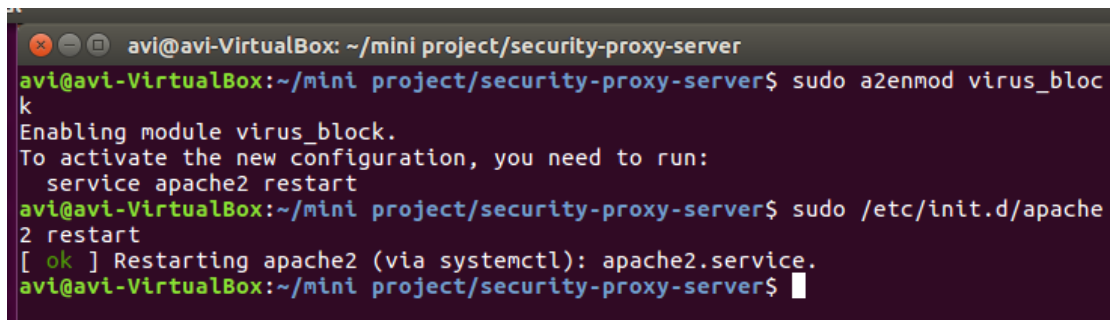
דוגמת הרצה:

נראה את פעולת הכלי עבור ה BlackList.txt שראינו לעיל. נזכיר:



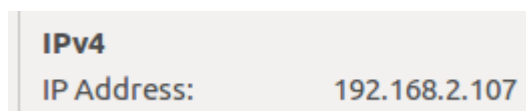
```
BlackList.txt
1 |cyclone-0.1.3.tar.bz2
2 |rcfviewer-0.2.tar.bz2
3
```

לאחר שהכלי מותקן, והרשימה השחורה מעודכנת, נפעיל את השרת:



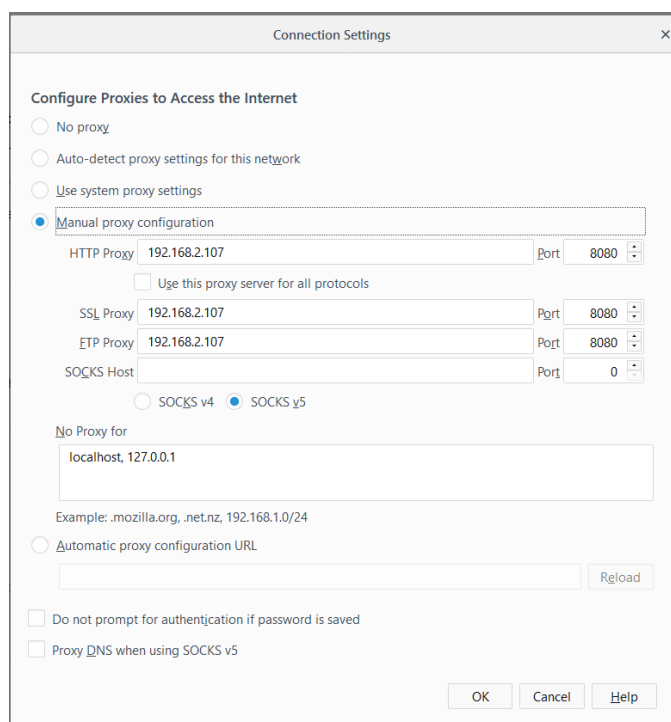
```
avi@avi-VirtualBox: ~/mini project/security-proxy-server
avi@avi-VirtualBox:~/mini project/security-proxy-server$ sudo a2enmod virus_block
Enabling module virus_block.
To activate the new configuration, you need to run:
service apache2 restart
avi@avi-VirtualBox:~/mini project/security-proxy-server$ sudo /etc/init.d/apache2 restart
[ ok ] Restarting apache2 (via systemctl): apache2.service.
avi@avi-VirtualBox:~/mini project/security-proxy-server$
```

לאחר מכן נבדוק מה ה IP של המכונה עליה רץ השרת:



```
IPv4
IP Address: 192.168.2.107
```

ובדפדפן הלקוח נגדיר את הגדרות הפרוקסי בצורה זו:



Connection Settings

Configure Proxies to Access the Internet

☐ No proxy

☐ Auto-detect proxy settings for this network

☐ Use system proxy settings

☒ Manual proxy configuration

HTTP Proxy: 192.168.2.107 Port: 8080

☐ Use this proxy server for all protocols

SSL Proxy: 192.168.2.107 Port: 8080

FTP Proxy: 192.168.2.107 Port: 8080

SOCKS Host: localhost Port: 0

☐ SOCKS v4 ☒ SOCKS v5

No Proxy for: localhost, 127.0.0.1

Example: .mozilla.org, .net.nz, 192.168.1.0/24

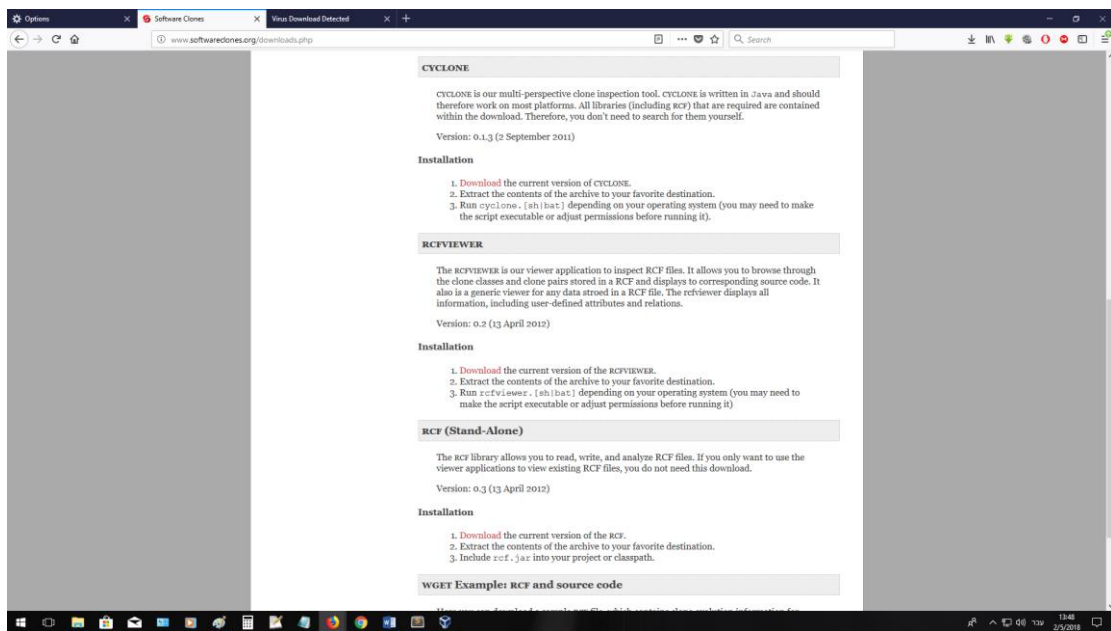
☐ Automatic proxy configuration URL

Do not prompt for authentication if password is saved

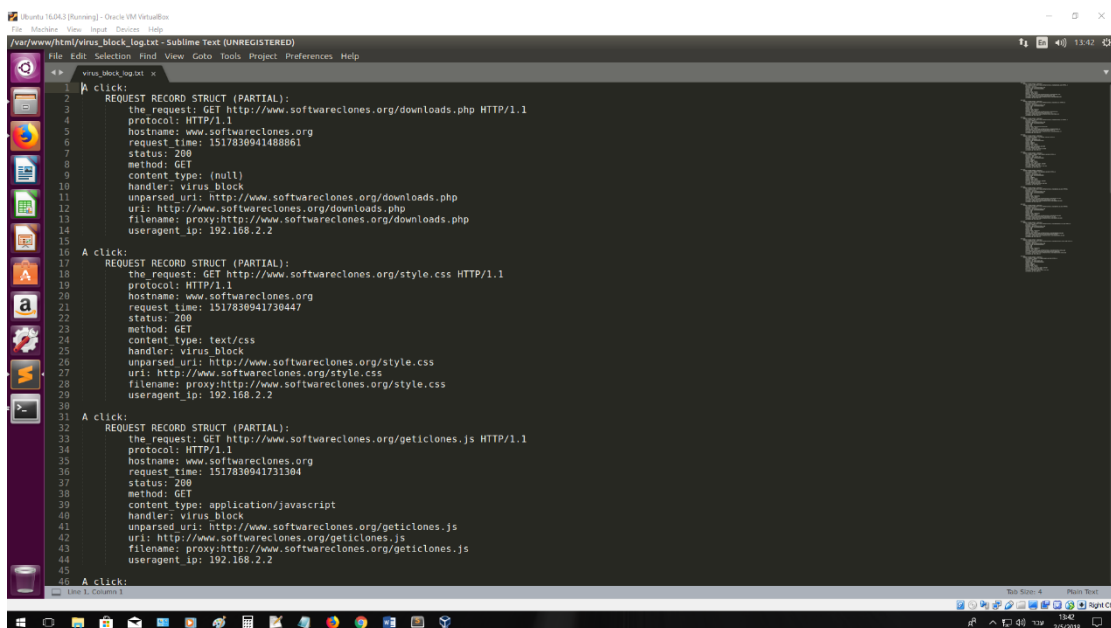
Proxy DNS when using SOCKS v5

OK Cancel Help

נגלוש כעת לאתר http מסויים:

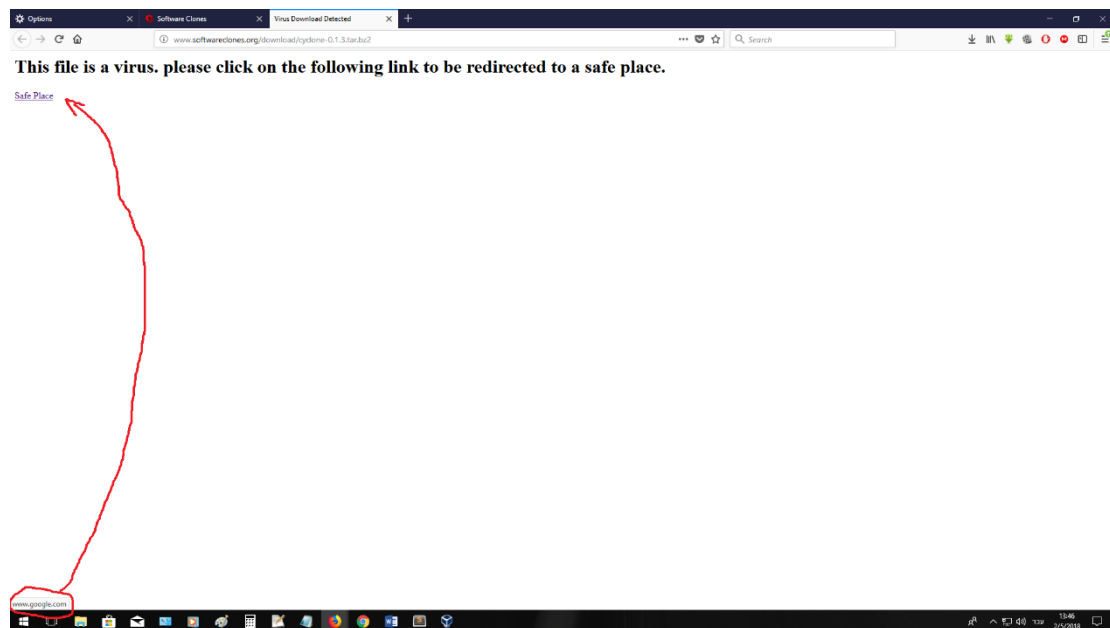


וכפי שנשים לב, יתחיל תיעוד של בקשות ה GET בקובץ virus_block_log.txt

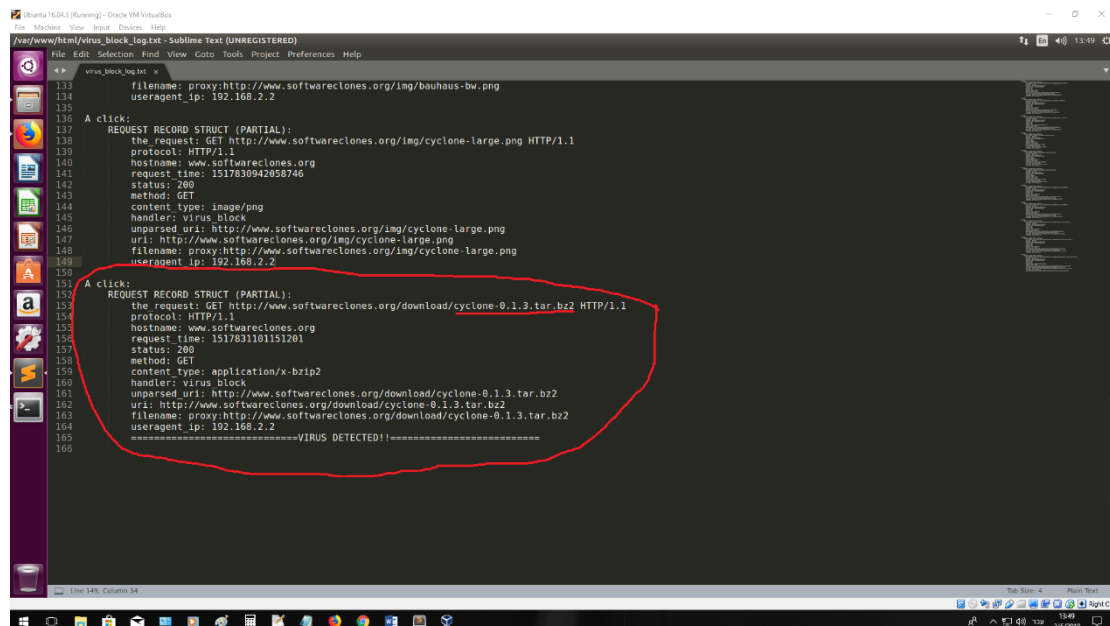


אם ננסה להוריד כעת את cyclclone, אשר מוגדר כקובץ עזר, לא נצליח.

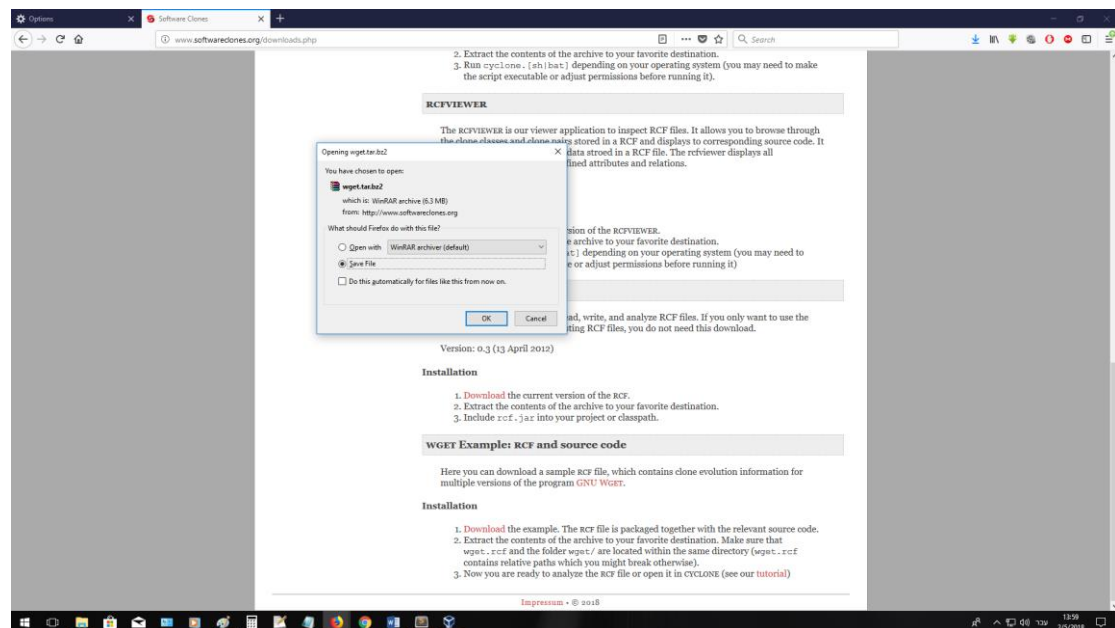
תחילה, השרת יציג הודעה על כך בדפדפן שלנו, עם לינק הפנייה לאתר הבית של גוגל ("אתר בטוח").



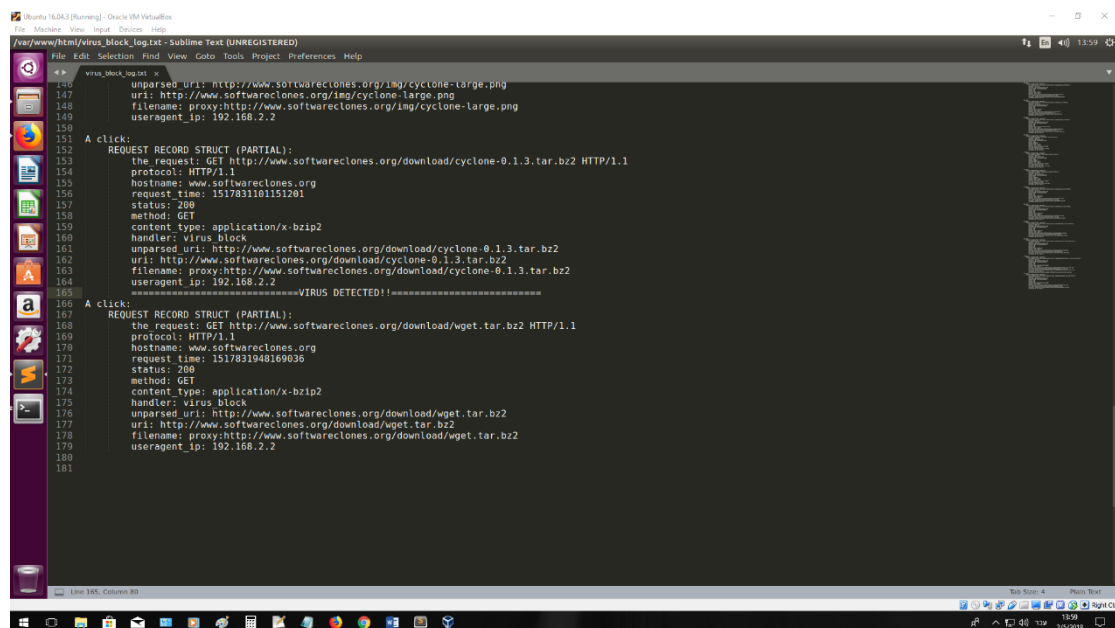
בנוסף, אנו נראה תיעוד של ניסיון ההורדה:



לעומת זאת, אם ננסה להוריד את WGET Example:



ופעולה זו תתועד כפעולה בטוחה:



סיכום:

הכלי שבנינו הינו כלי שמתאים לרשת ארגונית אשר מטרתו למנוע הורדה תמימה של קבצים עינים, הוא מצליח להתמודד עם מקרים פשוטים, מבצע ניטור ותגובה ברמה בסיסית, אשר מתאימה לארגון בו למשתמשים אין ידע עמוק בנושאי אבטחת מידע.

הכלי מתממשק בצורה טובה לשרת אפאצ'י 2 במוד פרוקסי, פשוט להתקנה, וקל לתחזוקה שבאה לידי ביטוי בעדכונים לבסיס הנתונים שלו.

ביבליוגרפיה:

- לצורך הבנה של אופן עבודת מודולים באפאצ'י 2 ואיך לכתוב מודול נעזרנו באתר של אפאצ'י 2 ובעיקר באתר זה:
<https://httpd.apache.org/docs/2.4/developer/modguide.html>
- לצורך קינפוג שרת פרוקסי של אפאצ'י כשרת forward-proxy נעזרנו באתר זה:
<http://geek-university.com/apache/configure-apache-as-a-forward-proxy>
- לצורך הדוגמאות שהצגנו השתמשנו באתר של softwareclones:
<http://www.softwareclones.org>
- צריכים לדעת. google, stackoverflow, Wikipedia עבור כל דבר נוסף שהיינו