

Logi Web serwera

Otrzymujemy dwa pliki plik pcapng oraz logi serwera http, plik z ruchem sieciowym polecam zamienić na pcap umożliwi nam to jego analizę również w narzędziu tcpdump jednak w tym przypadku nie będziemy go używać.

1. Jaki jest adres IP Atakującego

Pierwszą rzeczą jaką zrobimy będzie wyświetlimy pierwsze dwa logowania aby sprawdzić dokładny format logów co pomoże nam w dalszej fazie analizy

```
mtchal@linux: /media/mtchal/c640ace7-af1e-416c-a7d1-5f59b7e96f1d/konkurs$ ls
access.log  konkurs.pcap
mtchal@linux: /media/mtchal/c640ace7-af1e-416c-a7d1-5f59b7e96f1d/konkurs$ tail -n 2 access.log
szkolasecurity.pl:80 3.13.3.8 - - [16/Jan/2022:12:29:43 +0100] "GET /2022/01/07/nowe-szkolenie/ HTTP/1.1" 200 5521 "-" "Mozilla/5.0 (Windows NT 10.0; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/97.0.4692.71 YaBrowser/21.11.0 Yowser/2.5 Safari/537.36"
szkolasecurity.pl:80 2.0.2.2 - - [16/Jan/2022:12:29:45 +0100] "GET / HTTP/1.1" 200 4382 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/97.0.4692.71 Safari/537.36"
mtchal@linux: /media/mtchal/c640ace7-af1e-416c-a7d1-5f59b7e96f1d/konkurs$
```

Za pomocą narzędzi do filtrowania takich jak cat, cut, grep, uniq, sort będziemy starali się wyciągnąć coś podejrzanego:

```
mtchal@linux: /media/mtchal/c640ace7-af1e-416c-a7d1-5f59b7e96f1d/konkurs$ cat access.log | cut -d " " -f 2,13 | sort | uniq -c | sort
10326 1.3.3.7 "absolutnie_nie_wpscan"
4807 3.13.3.8 "Mozilla/5.0"
4891 2.0.2.2 "Mozilla/5.0"
7 1.3.3.7 "absolutnie_nie_hakerzy"
7 :11 "Apache/2.4.41"
87 1.3.3.7 "Mozilla/5.0"
9 30.30.10.10 "WordPress/5.8.3;"
mtchal@linux: /media/mtchal/c640ace7-af1e-416c-a7d1-5f59b7e96f1d/konkurs$
```

Jak widać na screenie udało nam się znaleźć adres z podejrzanym polem User-agent a dokładnie z trzema różnymi, jest to prawdopodobny atakujący czyli odpowiedzią jest:

1.3.3.7

2. Z jakiego "User-agent" korzystał atakujący (podaj wszystkie 3):

Odpowiedz znaleźliśmy już podczas wykonywania pierwszego zadania:

absolutnie_nie_wpscan

Mozilla/5.0

absolutnie_nie_hakerzy

Webshell jest to wrzucony na serwer webowy backdoor w postaci pliku php w którym znajduje się argument cmd do którego atakujący może się odwoływać np. poprzez modyfikację nagłówka http lub po prostu doklejając do adresu ?cmd="polecenie" spróbujemy więc znaleźć taki argument w zapytaniach z adresu atakującego:

Nie wszystkie polecenia są dla nas czytelne ponieważ zostały zakodowany aby można było je wykonywać z poziomu przeglądarki przypisując je do argumentu cmd dlatego użyjemy narzędzia cyberchef aby je odkodować:

Odpowiedź:

```
printf "open 10.0.0.20\nuser admin_bkp szkolasecurity2022!\nncd backup\nls\nbye\n" | ftp -n
```

PCAP

4. Wpisz poprawne credentiala do wp (format: "login:pass")

Aby odpowiedzieć na to pytanie musimy znaleźć zapytania które skutkowało odpowiedzią od serwera www kodem 302 czyli przekierowaniem na inną podstronę robimy to w następujący sposób:

ip.addr == 1.3.3.7 and (http.request.method == POST or http.response.code == 302)						
No.	Time	Source	Destination	Protocol	Length	Info
54755	336.595343318	1.3.3.7	30.30.10.10	HTTP	568	POST /wp-login.php HTTP/1.1 (application/x-www-form-urlencoded)
54756	336.595719420	1.3.3.7	30.30.10.10	HTTP	572	POST /wp-login.php HTTP/1.1 (application/x-www-form-urlencoded)
54758	336.596390694	1.3.3.7	30.30.10.10	HTTP	567	POST /wp-login.php HTTP/1.1 (application/x-www-form-urlencoded)
54761	336.597267354	1.3.3.7	30.30.10.10	HTTP	571	POST /wp-login.php HTTP/1.1 (application/x-www-form-urlencoded)
54766	336.599491300	1.3.3.7	30.30.10.10	HTTP	570	POST /wp-login.php HTTP/1.1 (application/x-www-form-urlencoded)
54781	336.784272223	1.3.3.7	30.30.10.10	HTTP	574	POST /wp-login.php HTTP/1.1 (application/x-www-form-urlencoded)
54782	336.784636694	1.3.3.7	30.30.10.10	HTTP	567	POST /wp-login.php HTTP/1.1 (application/x-www-form-urlencoded)
54785	336.785449096	1.3.3.7	30.30.10.10	HTTP	571	POST /wp-login.php HTTP/1.1 (application/x-www-form-urlencoded)
54786	336.785812818	1.3.3.7	30.30.10.10	HTTP	568	POST /wp-login.php HTTP/1.1 (application/x-www-form-urlencoded)
54792	336.788412161	1.3.3.7	30.30.10.10	HTTP	572	POST /wp-login.php HTTP/1.1 (application/x-www-form-urlencoded)
54807	336.886701905	1.3.3.7	30.30.10.10	HTTP	568	POST /wp-login.php HTTP/1.1 (application/x-www-form-urlencoded)
54808	336.886997391	1.3.3.7	30.30.10.10	HTTP	572	POST /wp-login.php HTTP/1.1 (application/x-www-form-urlencoded)
54809	336.887386524	1.3.3.7	30.30.10.10	HTTP	567	POST /wp-login.php HTTP/1.1 (application/x-www-form-urlencoded)
54812	336.887856188	1.3.3.7	30.30.10.10	HTTP	571	POST /wp-login.php HTTP/1.1 (application/x-www-form-urlencoded)
55330	373.394953677	30.30.10.10	1.3.3.7	HTTP	1195	HTTP/1.1 302 Found
55649	374.588149380	1.3.3.7	30.30.10.10	HTTP	1841	POST /wp-admin/admin-ajax.php HTTP/1.1 (application/x-www-form-urlencoded)
56018	396.585484940	1.3.3.7	30.30.10.10	HTTP	1883	POST /wp-admin/update.php?action=upload-plugin HTTP/1.1 (application/x-www-form-urlencoded)

(Response in frame: 55330)

Next request in frame: 55649

File Data: 126 bytes

HTML Form URL Encoded: application/x-www-form-urlencoded

- Form item: "log" = "admin_bkp"
 - Key: log
 - Value: admin_bkp
- Form item: "pwd" = "szkolasecurity2022!"
 - Key: pwd
 - Value: szkolasecurity2022!
- Form item: "wp-submit" = "Log In"
 - Key: wp-submit
 - Value: Log In
- Form item: "redirect_to" = "http://szkolasecurity.pl/wp-admin/"
 - Key: redirect_to
 - Value: http://szkolasecurity.pl/wp-admin/
- Form item: "testcookie" = "1"
 - Key: testcookie
 - Value: 1

0000 00 50 56 38 08 36 00 0c 29 2b e0 bd 08 00 45 00 PVB.6...)+....E:

0010 02 d2 e3 e3 02 40 00 3f 06 29 f2 01 03 03 07 1e 1e@?).....

0020 9a 9a 98 5c 00 50 59 18 7f 27 78 18 22 7a 00 18PY: 'x'Z..

Odpowiedz:

admin_bkp:szkolasecurity2022!

5. Jaka jest odpowiedź z toola do rekonesansu wewnętrznego (jaki porty są otwarte - podaj 2)

Wiemy już że atakujący odwoływał się do parametru cmd w adresie aplikacji www dlatego tym razem również postaramy się wyfiltrować ten parametr w ruchu sieciowym.

ip.addr == 1.3.3.7 and (http.request.uri contains "cmd" or http.response_for.uri contains "cmd")						
No.	Time	Source	Destination	Protocol	Length	Info
56175	411.393784524	1.3.3.7	30.30.10.10	HTTP	731	GET /wp-content/uploads/2022/01/absolutnie_nie_webshell.php?c...
56180	411.400133083	30.30.10.10	1.3.3.7	HTTP	278	HTTP/1.1 200 OK (text/html)
56471	449.702243850	1.3.3.7	30.30.10.10	HTTP	221	GET /wp-content/uploads/2022/01/absolutnie_nie_webshell.php?c...
56473	449.706076604	30.30.10.10	1.3.3.7	HTTP	260	HTTP/1.1 200 OK (text/html)
56631	467.978463701	1.3.3.7	30.30.10.10	HTTP	222	GET /wp-content/uploads/2022/01/absolutnie_nie_webshell.php?c...
56633	467.986632541	30.30.10.10	1.3.3.7	HTTP	1364	HTTP/1.1 200 OK (text/html)
56751	481.678904559	1.3.3.7	30.30.10.10	HTTP	226	GET /wp-content/uploads/2022/01/absolutnie_nie_webshell.php?c...
56753	481.686251640	30.30.10.10	1.3.3.7	HTTP	343	HTTP/1.1 200 OK (text/html)
57021	518.485375651	1.3.3.7	30.30.10.10	HTTP	273	GET /wp-content/uploads/2022/01/absolutnie_nie_webshell.php?c...
57053	520.574800963	30.30.10.10	1.3.3.7	HTTP	365	HTTP/1.1 200 OK (text/html)
57211	541.197010605	1.3.3.7	30.30.10.10	HTTP	313	GET /wp-content/uploads/2022/01/absolutnie_nie_webshell.php?c...
57213	541.246123219	30.30.10.10	1.3.3.7	HTTP	277	HTTP/1.1 200 OK (text/html)
57401	567.558298917	1.3.3.7	30.30.10.10	HTTP	326	GET /wp-content/uploads/2022/01/absolutnie_nie_webshell.php?c...
57493	567.618850868	30.30.10.10	1.3.3.7	HTTP	310	HTTP/1.1 200 OK (text/html)
57541	581.848792226	1.3.3.7	30.30.10.10	HTTP	346	GET /wp-content/uploads/2022/01/absolutnie_nie_webshell.php?c...
57552	581.908948381	30.30.10.10	1.3.3.7	HTTP	71	HTTP/1.1 200 OK (text/html)

Frame 57021: 273 bytes on wire (2184 bits), 273 bytes captured (2184 bits) on interface ens33, id 0

Ethernet II, Src: VMWare_2b:e0:bd (00:0c:29:2b:e0:bd), Dst: VMWare_38:08:36 (00:50:56:38:08:36)

Internet Protocol Version 4, Src: 1.3.3.7, Dst: 30.30.10.10

Transmission Control Protocol, Src Port: 39842, Dst Port: 80, Seq: 1, Ack: 1, Len: 207

Hypertext Transfer Protocol

GET /wp-content/uploads/2022/01/absolutnie_nie_webshell.php?cmd=nc+ -w1+ -v10.0.0.20+1-100+2%3E%261+%7C+grep+-v+refused HTTP/1.1\r\n

Host: szkolasecurity.pl\r\n

User-Agent: absolutnie_nie_hakierzyr\r\n

Accept: */*\r\n

\r\n

[Full request URI: http://szkolasecurity.pl/wp-content/uploads/2022/01/absolutnie_nie_webshell.php?cmd=nc+ -w1+ -v10.0.0.20+1-100+2%3E%261+%7C+grep+-v+refused]

[HTTP request 1/1]

[Response in frame: 57053]

W tym przypadku atakujący korzysta z narzędzia netcat co widać w odwołaniu do argumentu cmd w zapytaniu, zapewne serwer do którego miał dostęp nie było innych narzędzi dzięki którym mógłby to sprawdzić a pobieranie czegokolwiek wywołało by zbyt duży ruch.

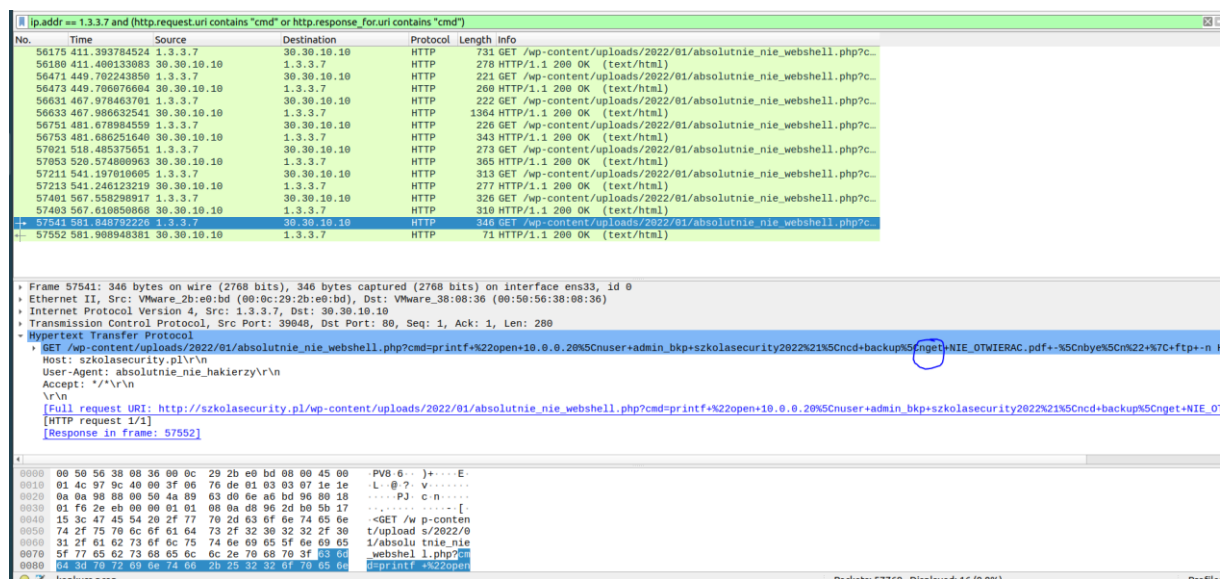
Odpowiedź:

Pod zapytaniem z parametrem cmd znajduje się odpowiedź serwera która wskazuje że porty 21 oraz 80 odpowiadają oznacza to że są otwarte (działają na nich usługi)

21,80

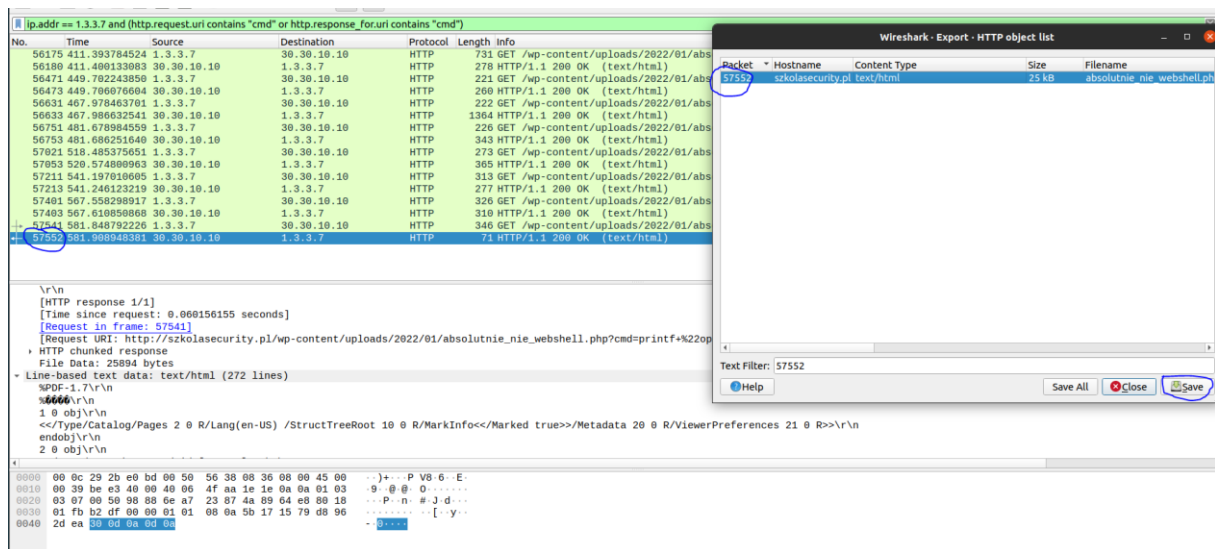
6. Jaki plik został ściągnięty przez atakującego?

Korzystając z tego samego filtru jak w pytaniu 5 idąc niżej możemy wywnioskować że atakującego zainteresował serwer FTP działający na porcie 21. Wyświetlił pliki które dostępne są na serwerze następnie wykonał polecenie get w lokalizacji w której znajdował się plik NIE_OTWIERAC.pdf:



7. Jaka była zawartość ściąganego pliku?

Przechodzimy w wiresharku do: File > Export Objects następnie filtrujemy po numerze pakietu widocznym po lewej stronie w pierwszej kolumnie:



Zapisujemy plik z rozszerzeniem .pdf możemy go uruchomić jako normalny plik pdf:

