

# KRYCY25Z – Laboratorium 1 – Systemy EDR

Piotr Szewczyk 311 105

Paweł Murdzek 310 850

Treść zadania:

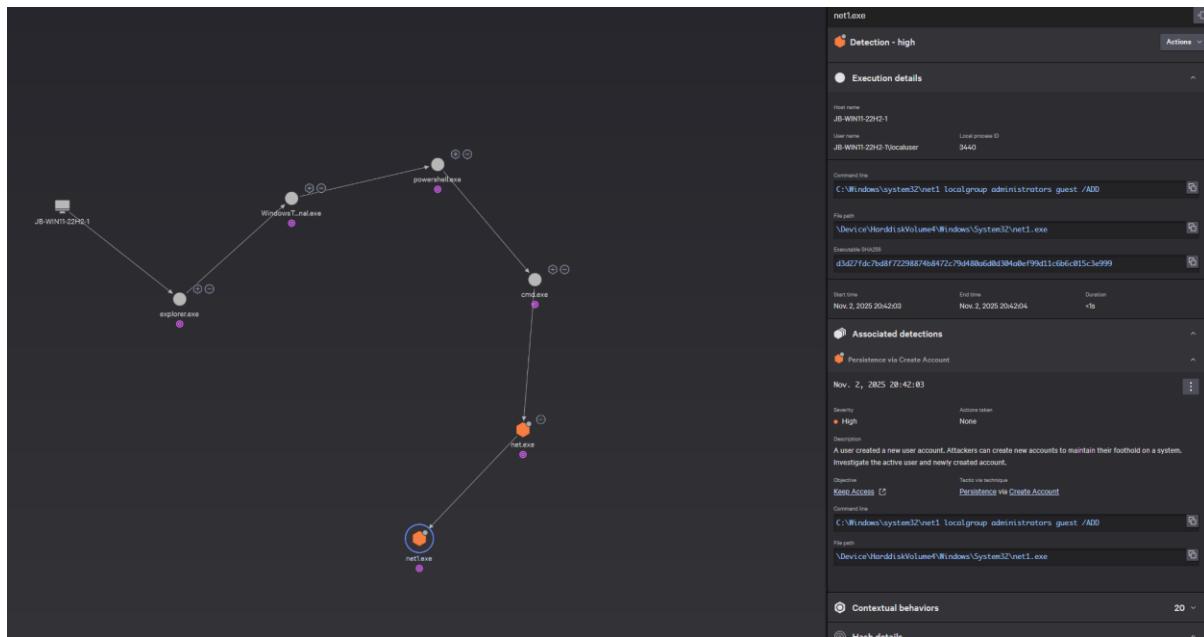
## Zadanie główne

Przygotuj raport z analizy incydentu na podstawie alertów i zawartości konsoli CrowdStrike Falcon (<https://falcon.eu-1.crowdstrike.com>). Wykorzystaj zawartość konsoli do dokumentacji ataku.

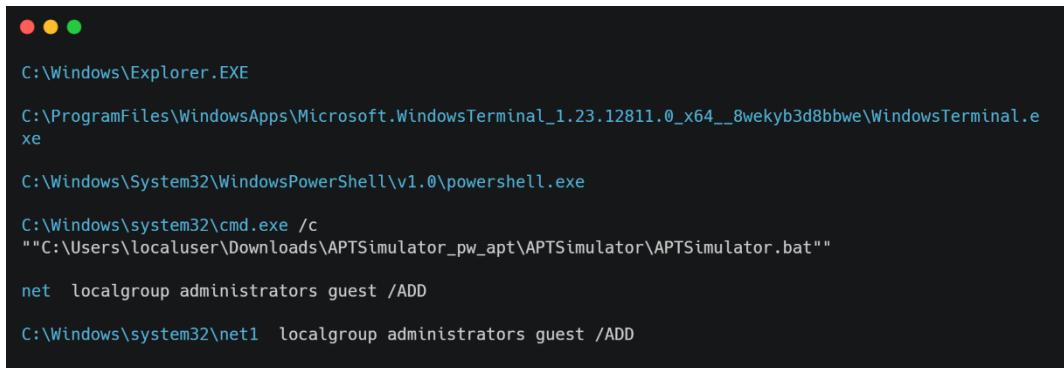
Znajdź i udokumentuj przynajmniej 5 różnych technik (wskazując na ich taktyki). Uwzględnij nazwy procesów, parametry, polecenia CMD i ich wyjaśnienia, zrzuty ekranu, znaczniki czasu. Szerzej wyjaśnij zastosowane techniki ataku poprzez mapowanie na matrycę MITRE ATT&CK. Skorzystaj z ogólnodostępnych źródeł wiedzy.

### 1. Technika Persistence via Create Account

Raport dotyczący 1 techniki:



Rysunek 1 Graf techniki Persistence via Create Account



```
C:\Windows\Explorer.EXE
C:\ProgramFiles\WindowsApps\Microsoft.WindowsTerminal_1.23.12811.0_x64_8wekyb3d8bbwe\WindowsTerminal.e
xe
C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe
C:\Windows\system32\cmd.exe /c
"C:\Users\localuser\Downloads\APTSimulator_pw_apt\APTSimulator\APTSimulator.bat"
net localgroup administrators guest /ADD
C:\Windows\system32\net1 localgroup administrators guest /ADD
```

Rysunek 2 cmdline techniki nr. 1

Detect time: 2025-11-02 20:42:03

Host: JB-WIN11-22H2-1

User: localuser

Proces: C:\Windows\system32\cmd.exe (uruchamia APTSimulator.bat) → net.exe / net1.exe

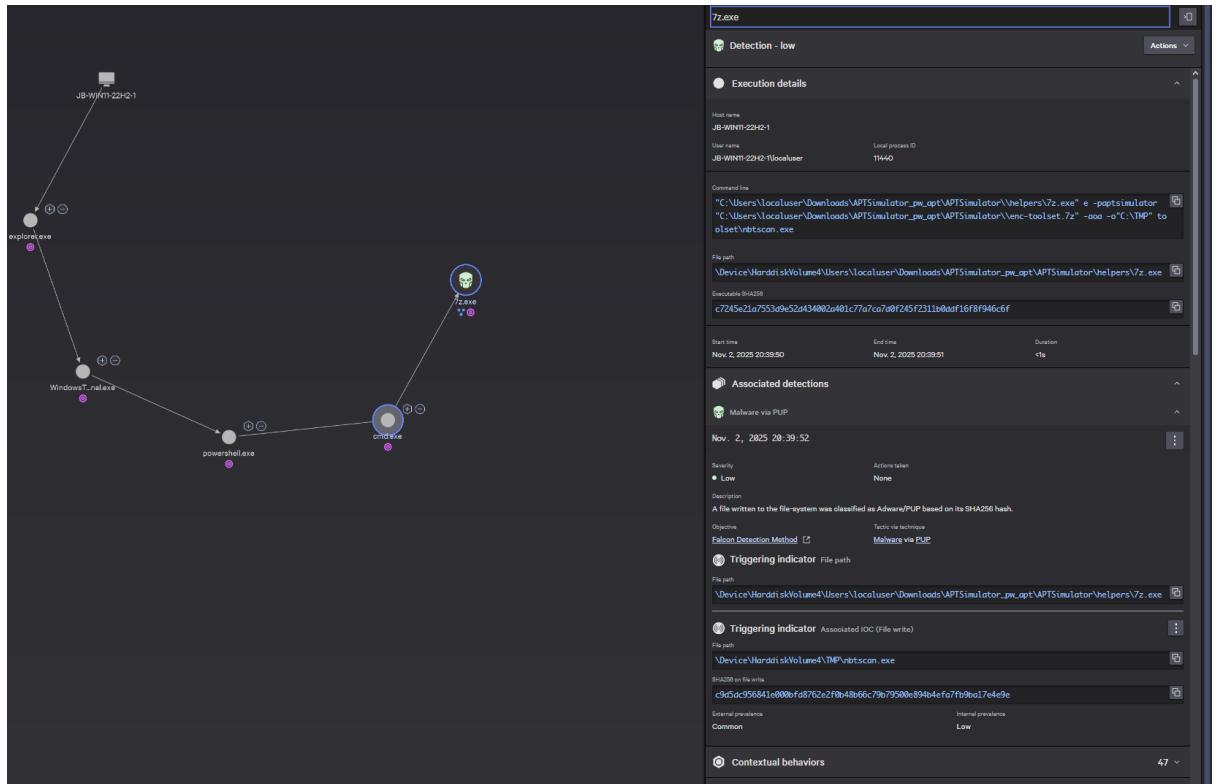
Polecenie: net localgroup administrators guest /ADD

Hash: d3d27fdc7bd8f72298874b8472c79d480a6d0d304a0ef99d11c6b6c015c3e999

Opis: Z logów i panelu detekcji wynika, że uruchomiono skrypt (APTSimulator.bat), który wykonał polecenie dodania konta 'guest' do lokalnej grupy Administrators na hoście JB-WIN11-22H2-1. Operacja zapewnia trwały dostęp z uprawnieniami administracyjnymi i mapuje się na MITRE ATT&CK: T1136 - Create Account ([MITRET1136](#)).

## 2. Technika Malware via PUP

Raport dotyczący 2 techniki:



Rysunek 3 Graf techniki Malware via PUP

```
C:\Windows\Explorer.EXE
"C:\ProgramFiles\WindowsApps\Microsoft.WindowsTerminal_1.23.12811.0_x64_8wekyb3d8bbwe\WindowsTerminal.exe"

C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe

C:\Windows\system32\cmd.exe /c
"C:\Users\localuser\Downloads\APTSimulator_pw_apt\APTSimulator\APTSimulator.bat"

"C:\Users\localuser\Downloads\APTSimulator_pw_apt\APTSimulator\helpers\7z.exe" e -paptimulator
"C:\Users\localuser\Downloads\APTSimulator_pw_apt\APTSimulator\enc-toolset.7z" -aoa -o"C:\TMP"
toolset\nbtscan.exe
```

Rysunek 4 cmdline techniki nr. 2

Detect time: 2025-11-02 20:39:52

Host: JB-WIN11-22H2-1

User: localuser

Proces: cmd.exe → 7z.exe

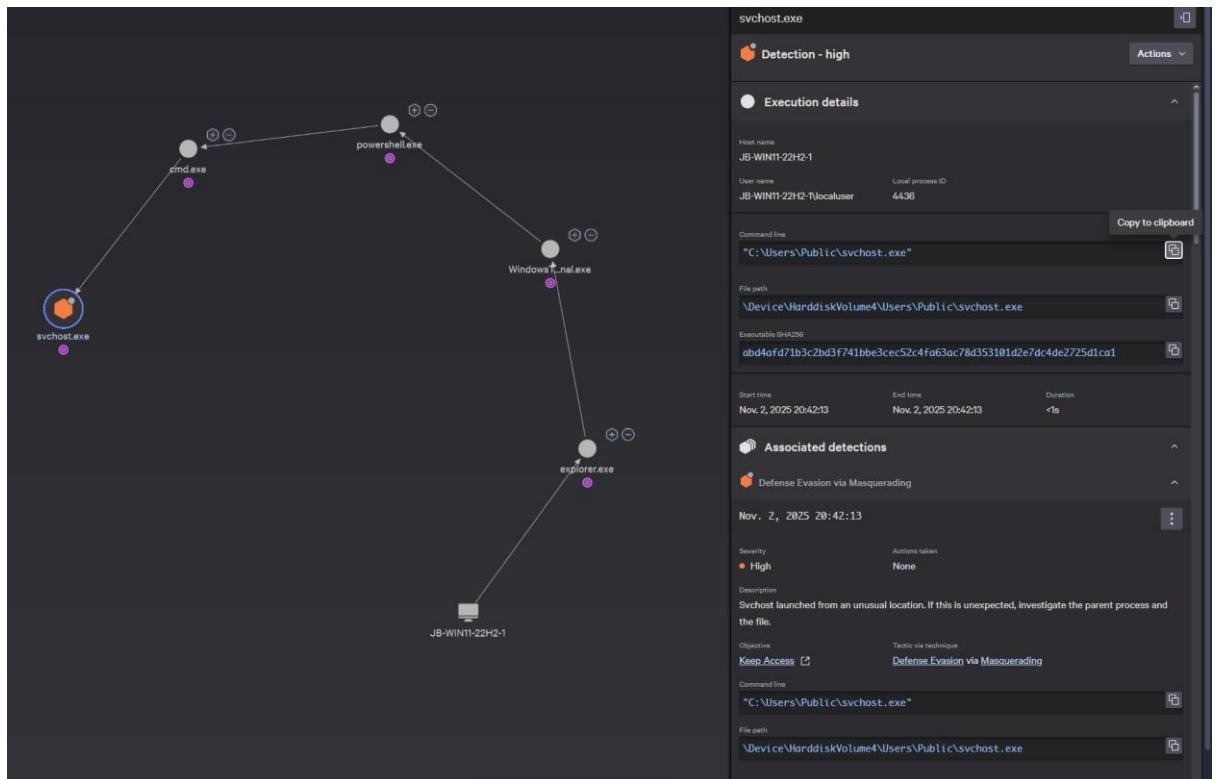
Hash: c7245e21a7553d9e52d434002a401c77a7ca7d0f245f2311b0ddf16f8f946c6f

The screenshot shows a dark-themed user interface for a security tool. At the top left, it says "Malware via PUP". Below that is a section titled "Description" with the subtext "A file written to the file-system was classified as Adware/PUP based on its SHA256 hash." Underneath this, there are two sections labeled "Triggering indicator" each, followed by "File path". The first "File path" is "\Device\HarddiskVolume4\Users\localuser\Downloads\APTSimulator\_pw\_apt\APTSimulator\helpers\7z.exe". The second "File path" is "\Device\HarddiskVolume4\TMP\nbtscan.exe". Below these paths are two more sections: "Associated IOC (File write)" and "SHA256 on file write". The "Associated IOC (File write)" section has a "File path" of "\Device\HarddiskVolume4\TMP\nbtscan.exe". The "SHA256 on file write" section contains the hash "c9d5dc956841e000bfd8762e2f0b48b66c79b79500e894b4efa7fb9ba17e4e9e". At the bottom, there are two prevalence sections: "External prevalence" (Common) and "Internal prevalence" (Low).

Rysunek 5 zapis pliku oznaczony jako Adware/PUP (hash)

Opis: Skrypt APTSimulator uruchomił proces 7z.exe w celu rozpakowania zaszyfrowanego archiwum „enc-toolset.7z”. W efekcie w katalogu tymczasowym C:\TMP zapisano plik nbtscan.exe. Detekcja Falcona sklasyfikowała ten zapis jako Adware/PUP na podstawie hash. Zachowanie odpowiada etapowi Ingress Tool Transfer ([MITRE T1105](#)), w którym atakujący wprowadza do systemu dodatkowe narzędzia. Z raportu wynika, że za PHANTOM ALPHA odpowiedzialnym może być m.in. GALLIUM. Nie zauważaliśmy, żeby później nbtscan.exe został uruchomiony.

### 3. Defense Evasion via Masquerading



Rysunek 6 Graf techniki Defense Evasion via Masquerading

```
C:\Windows\Explorer.EXE
"C:\ProgramFiles\WindowsApps\Microsoft.WindowsTerminal_1.23.12811.0_x64_8wekyb3d8bbwe\WindowsTerminal.exe"
C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe
C:\Windows\system32\cmd.exe /c
""C:\Users\localuser\Downloads\APTSimulator_pw_apt\APTSimulator\APTSimulator.bat"""

"C:\Users\Public\svchost.exe"
```

Rysunek 7 cmdline techniki nr. 3

Detect time: 2025-11-02 20:42:13

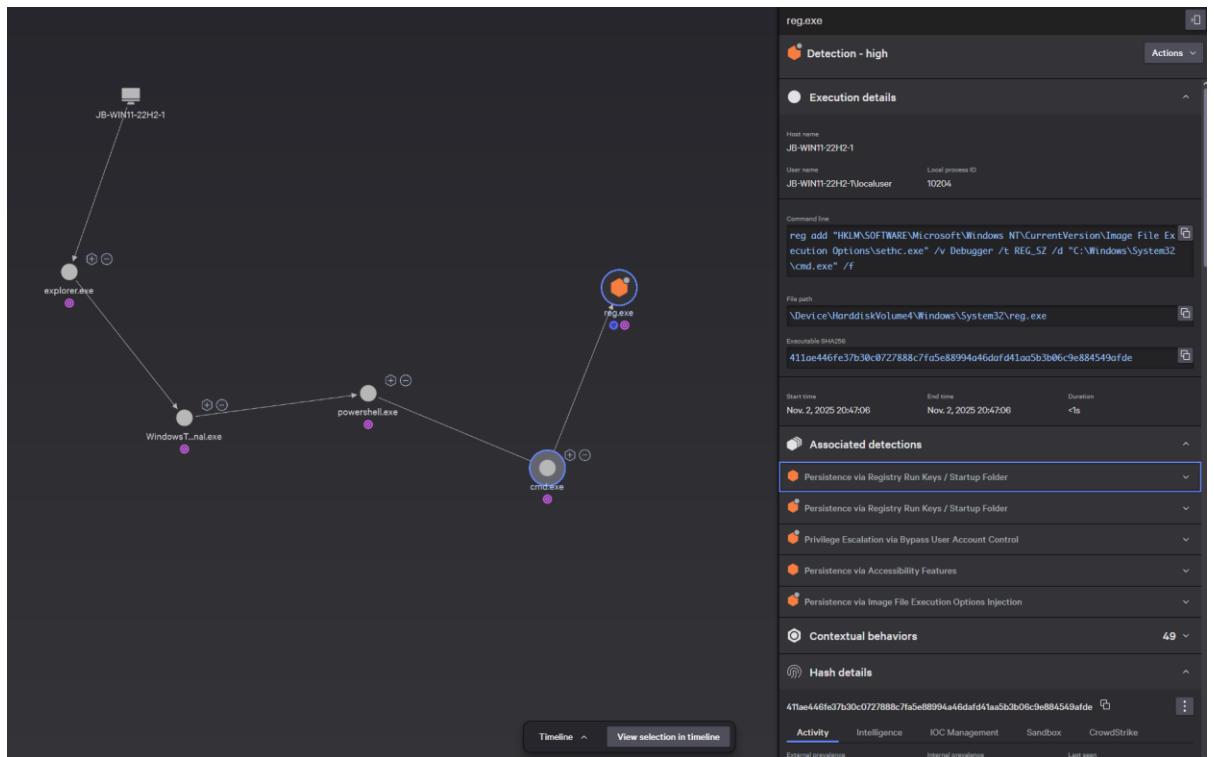
Host: JB-WIN11-22H2-1

User: localuser

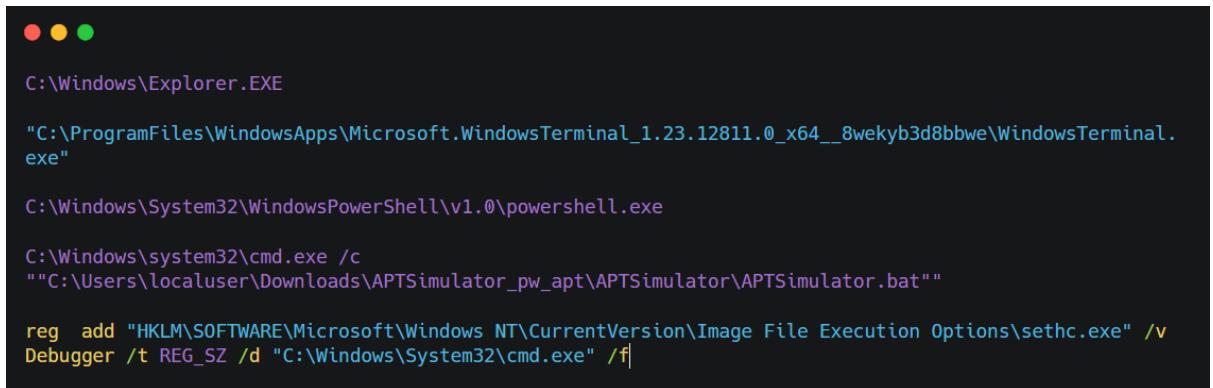
Hash: abd4afd71b3c2bd3f741bbe3cec52c4fa63ac78d353101d2e7dc4de2725d1ca1

Opis: Wykryto uruchomienie pliku nazwanego svchost.exe z nietypowej lokalizacji użytkownika (C:\Users\Public\svchost.exe). Nazwa odpowiada legalnemu procesowi systemowemu, jednak lokalizacja poza C:\Windows\System32 oraz brak wiarygodnego podpisu cyfrowego wskazują na próbę maskowania złośliwego programu. Zachowanie odzwierciedla technikę [T1036 -Masquerading](#).

#### 4. Technika Privilege Escalation via Bypass User Account Control



Rysunek 8 Graf techniki Privilege Escalation via Bypass User Account Control



```
C:\Windows\Explorer.EXE

"C:\ProgramFiles\WindowsApps\Microsoft.WindowsTerminal_1.23.12811.0_x64__8wekyb3d8bbwe\WindowsTerminal.exe"

C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe

C:\Windows\system32\cmd.exe /c
""C:\Users\localuser\Downloads\APTSimulator_pw_apt\APTSimulator\APTSimulator.bat""

reg add "HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Image File Execution Options\sethc.exe" /v Debugger /t REG_SZ /d "C:\Windows\System32\cmd.exe" /f
```

Rysunek 9 cmdline techniki nr. 4

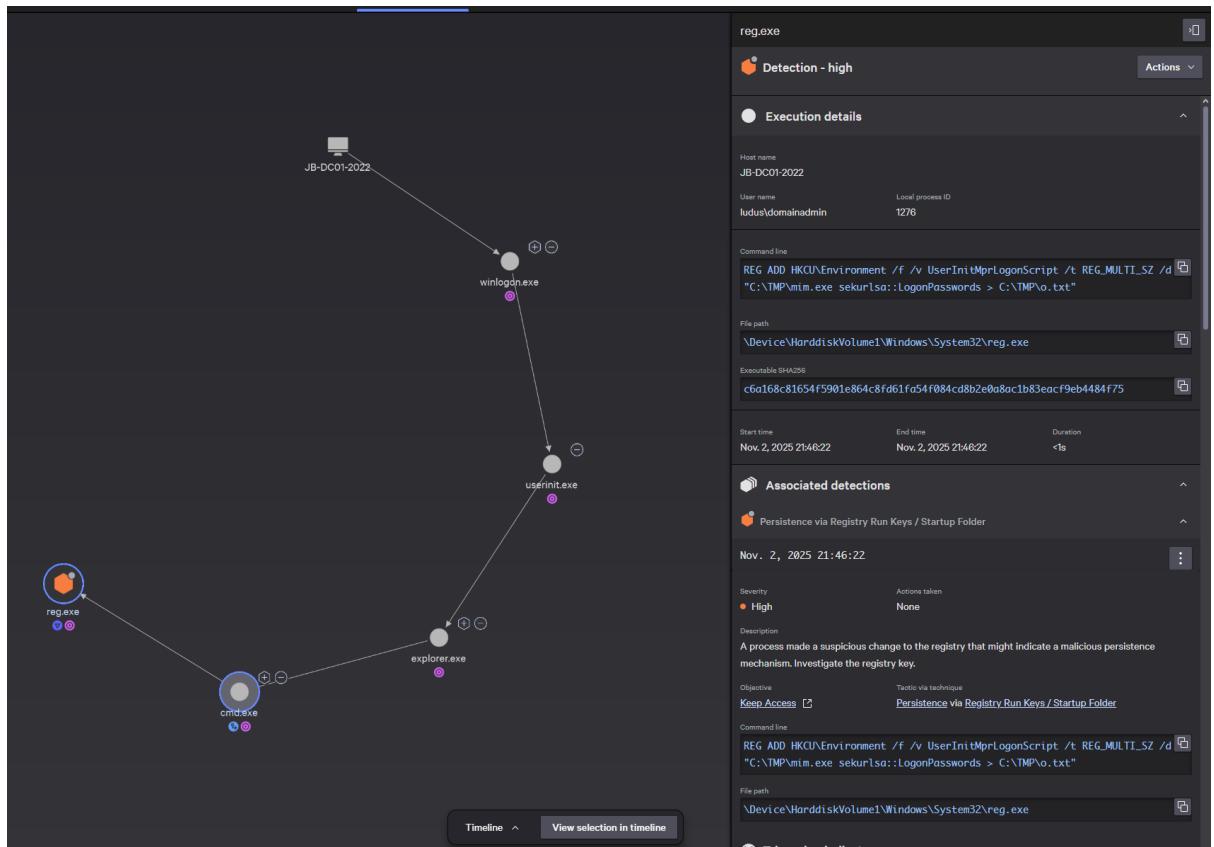
Hash: 411ae446fe37b30c0727888c7fa5e88994a46dafd41aa5b3b06c9e884549afde

W logach wykryto modyfikację klucza Image File Execution Options (IFEO) oraz wpis Debugger wskazujący na uruchamianie cmd.exe zamiast svchost.exe. To klasyczny sposób obchodzenia UAC i eskalacji przywilejów. Przy uruchomieniu procesu sethc.exe system uruchomi zamiast niego program wskazany w Debugger (cmd.exe) z kontekstem tego procesu. Jeśli target jest procesem systemowym uruchamianym z wyższymi uprawnieniami, atakujący uzyskuje eskalację i wykonanie kodu z wyższymi prawami Mapowanie MITRE: ([T1548.002](#))

Komenda: `reg add "HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Image File Execution Options\sethc.exe" /v Debugger /t REG_SZ /d "C:\Windows\System32\cmd.exe" /f` wiąże ze sobą inne detekcje:

- Persistence via Registry Run Keys / Startup Folder
- Persistence via Accessibility Features
- Persistence via Image File Execution Options Injection

## 5. Technika Credential Access via OS Credential Dumping



Rysunek 10 Graf techniki Credential Access via OS Credential Dumping

```
● ● ●
winlogon.exe
C:\Windows\system32\userinit.exe

C:\Windows\Explorer.EXE

"C:\Windows\system32\cmd.exe"

REG ADD HKCU\Environment /f /v UserInitMprLogonScript /t REG_MULTI_SZ /d "C:\TMP\mim.exe sekurlsa::LogonPasswords > C:\TMP\o.txt"
```

Rysunek 11 cmdline techniki nr. 5

Detect time: 2025-11-02T21:46:22Z

Host: JB-DC01-2022

User: ludus\domainadmin

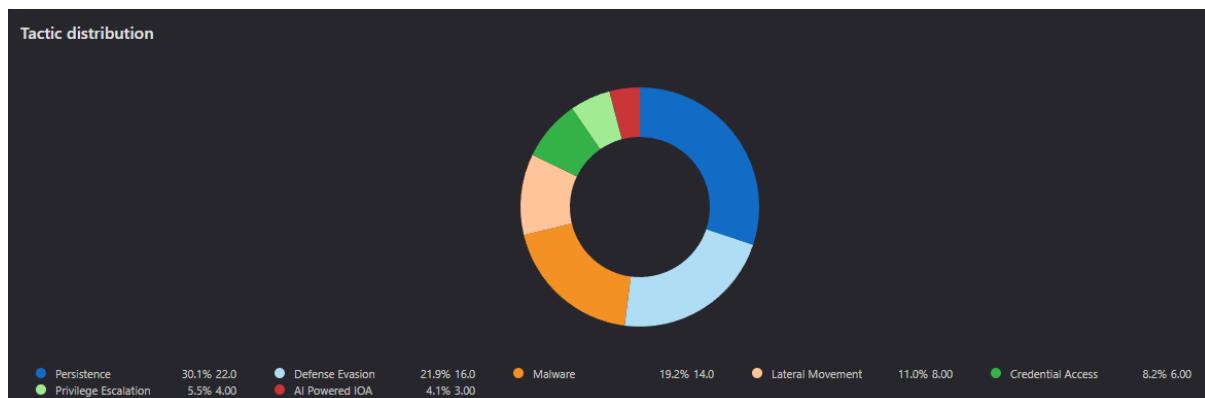
Hash: c6a168c81654f5901e864c8fd61fa54f084cd8b2e0a8ac1b83eacf9eb4484f75

komenda: REG ADD HKCU\Environment /f /v UserInitMprLogonScript /t REG\_MULTI\_SZ  
/d "C:\TMP\mim.exe sekurlsa::LogonPasswords > C:\TMP\o.txt"

Ścieżka rejestru: HKCU\Environment\UserInitMprLogonScript

Opis: Zmieniono klucz LogonScript tak, by przy logowaniu uruchamiać *mim.exe* z poleceniem *sekurlsa::LogonPasswords*. Narzędzie wykonało dump poświadczeń i zapisało wynik do C:\TMP\o.txt. To bezpośrednia implementacja techniki OS Credential Dumping ([MITRE T1003](#)).

Wszystkie taktyki użyte w ataku:



Rysunek 12 Dystrybucja wszystkich taktyk użytych w ataku

Wyżej mapowaliśmy jedynie techniki z ostatnich elementów procesów. W większości z nich na poprzednich etapach wyróżnia się jeszcze m.in.:

- T1059.003: Windows Command Shell
- T1059.001: PowerShell
- T1027.002: Pakowanie oprogramowania
- T1140: Dekodowanie plików lub informacji

## Pytania ogólne do sprawozdania

1. Jakie mechanizmy LotL (Living off the Land) wykorzystano podczas ataku?
2. Co należy zmienić w konfiguracji środowiska, aby nie dopuścić do eskalacji ataku?
3. Niektóre detekcje EDR nie są i nie muszą być automatycznie zablokowane ze względu na polityki (konfigurację narzędzia) lub typ detekcji.
  - Co można zmienić w pierwszej kolejności w ustawieniach CrowdStrike Falcon w ramach konfiguracji polityk prewencyjnych?
  - Jakie są inne możliwości wzmacniania wykrywania i reakcji poza politykami prewencyjnymi? Zaproponuj jakimi mechanizmami i funkcjonalnościami CrowdStrike Falcon można wzmacniać zdolności obronne.
4. Do czego wykorzystywane jest narzędzie Mimikatz? Dlaczego, mimo, że jest znane może nie zawsze być wykryte przez EDR/AV? W jakich technikach ataku na Active Directory jest wykorzystywane?
5. Jakie techniki „Persistence” zostały wykorzystane podczas ataku (zidentyfikuj minimum dwie)? Porównaj te mechanizmy.
6. Wyjaśnij sposób działania "C2 Beaconsing" przez Named Pipes. Jaki C2 Framework został wykorzystany w analizowanym scenariuszu?

Odpowiedź 1.

W trakcie incydentu zaobserwowano szereg technik Living off the Land (LOTL), polegających na nadużyciu wbudowanych narzędzi i mechanizmów systemu Windows w celu przeprowadzenia ataku. Wykorzystano narzędzie *reg.exe* do modyfikacji kluczy rejestru, w tym *Image File Execution Options (IFEO)* oraz kluczy autostartu takich jak *UserInitMprLogonScript*, co pozwoliło na utrzymanie trwałości w systemie. Do wykonywania poleceń i uruchamiania skryptów *.bat* używano wbudowanej powłoki *cmd.exe*, a także środowisk *powershell.exe* i *Windows Terminal*, które służyły do chainowania narzędzi oraz wykonywania dalszych złośliwych działań bez konieczności pobierania zewnętrznych payloadów. Narzędzia *net* oraz *net1* zostały wykorzystane do tworzenia nowych kont użytkowników oraz dodawania ich do grupy administratorów lokalnych, co umożliwiło eskalację uprawnień. Zaobserwowano również manipulację mechanizmami ułatwień dostępu, takimi jak *sethc.exe*, poprzez modyfikację kluczy rejestr. Ponadto, atakujący wykorzystywali legalne procesy systemowe, w tym *explorer.exe*, *userinit.exe* oraz *winlogon.exe*, jako procesy nadrzędne dla uruchamiania złośliwych aktywności, co pozwoliło na skuteczne ukrycie się w normalnym ruchu systemowym i obejście tradycyjnych zabezpieczeń opartych na białych listach aplikacji. Wszystkie te działania charakteryzują się wysokim stopniem wykorzystania natywnych funkcjonalności systemu, co utrudnia ich wykrycie przez konwencjonalne rozwiązania bezpieczeństwa. Jeżeli chodzi o 7.z to wydaje nam się, że to zależy, czy jest to LotL, czy nie w zależności od tego, czy jest on zainstalowany na atakowanym komputerze, ponieważ jest to normalne, legalne oprogramowanie.

Odpowiedź 3.

- Prawdopodobnie chodzi o zmiany z trybów z ‘detect’ na ‘prevent’, czyli z ‘wykrywania’ na ‘blokowanie’, we wszystkich ustawieniach. Wiele firm prawdopodobnie boi się oddać całkowitą kontrolę CrowdStrike’owi.

Odpowiedź druga mieści się tutaj. Dodatkowo odizolować hosta od sieci.

- Prompt (Google Gemini):

- „Jakie są możliwości wzmacniania wykrywania i reakcji poza politykami prewencyjnymi? Jakie mechanizmy i funkcjonalności oferuje CrowdStrike Falcon? Wymień i opisz je.”

#### **Falcon Identity (Ochrona Tożsamości)**

- **Problem:** Polityki prewencyjne chronią endpoint, ale nie chronią Active Directory (AD). Atakujący kradną hasła, by poruszać się po sieci.
- **Rozwiążanie:** Falcon Identity monitoruje w czasie rzeczywistym **kontrolery domeny**. Wykrywa i blokuje próby kradzieży haseł (jak DCSync), anomalie logowania, próby eskalacji uprawnień i ruch boczny (lateral movement). To absolutnie krytyczny moduł.

#### **Falcon Spotlight (Zarządzanie podatnościami)**

- **Problem:** Prewencja blokuje ataki, ale co z "dziurami" (podatnościami), które te ataki wykorzystują?
- **Rozwiążanie:** Spotlight to skaner podatności, który działa **w czasie rzeczywistym** i bez agenta (używa tego samego agenta Falcon). Pokazuje, które maszyny mają niezałączaną krytyczną lukę (np. w Chrome, w Windows), co pozwala zespołowi IT priorytetyzować latacie, zanim dojdzie do ataku.

#### **Falcon Discover (Higiena IT i Widoczność)**

- **Problem:** Nie możesz chronić tego, czego nie widzisz.
- **Rozwiążanie:** Discover daje pełną widoczność:
  - **Kto:** Jacy użytkownicy logują się na jakich maszynach? (np. do wykrywania kont serwisowych logujących się interaktywnie).
  - **Co:** Jaki software jest zainstalowany w firmie? (np. do znalezienia nielegalnego oprogramowania lub "shadow IT").

- **Gdzie:** Które maszyny nie mają włączonego szyfrowania dysku lub firewalla?

### **Własne reguły (Custom IOA / Custom Alerts)**

- **Problem:** Domyślne reguły są świetne, ale co, jeśli nasza firma ma specyficzną, wrażliwą aplikację, której nikt inny nie ma?
- **Rozwiążanie:** Możesz pisać własne reguły wykrywania. Na przykład:

"ZAALARMUJ MNIE (lub ZABLOKUJ), jeśli proces cmd.exe zostanie uruchomiony jako potomek procesu naszej aplikacji księgowej (ksiegi.exe)."

- Pozwala to "utwardzić" obronę wokół najbardziej krytycznych zasobów firmy.

### **Falcon Cloud Security (CSPM / CWP)**

- **Problem:** Wiele firm przenosi swoje serwery do chmury (AWS, Azure, GCP). Tradycyjna ochrona endpointów tam nie wystarcza.
- **Rozwiążanie:** Te moduły rozszerzają ochronę na kontenery, Kubernetes i samą konfigurację platformy chmurowej (np. wykrywanie, czy S3 bucket nie jest publicznie otwarty)."

Zgadzamy się z tą odpowiedzią, faktycznie są takie rozwiązania i świadczą o nich ich datasheet'y. Na serwisie YouTube jest dużo tutoriali i pokazowych video, które w/w możliwości ukazują.

## Odpowiedź nr. 4

Mimikatz służy do kradzieży poświadczeń z pamięci systemów Windows. Jego główną funkcją jest możliwość "prześwietlenia" pamięci procesu lsass.exe (Local Security Authority Subsystem Service). Proces ten przechowuje dane uwierzytelniające użytkowników, którzy są lub byli niedawno zalogowani na danej maszynie. Mimikatz użyty został przypadku techniki nr. 5 z pierwszej części raportu; funkcja sekurlsa::LogonPasswords jest najbardziej znaną metodą, która daje może dać atakującemu hasła w .txt.

Dlaczego jest trudny do wykrycia? Przez to, że jest mechanizmem LotL i dzięki temu jest ciężki do wykrycia. „*Dodatkowo narzędzie często używa standardowych wywołań API systemu Windows i operuje w kontekście uprawnionego procesu systemowego, co sprawia, że jego działania mogą wyglądać jak normalna aktywność systemowa. Mimikatz wykorzystuje również zaawansowane techniki przeciwdziałania analizie i detekcji. Narzędzie może dynamicznie modyfikować swój kod w pamięci, używać technik obfuscacji i szyfrowania, a także implementować różne metody unikania*

*wykrycia przez systemy monitorujące. W niektórych przypadkach Mimikatz może nawet wykrywać obecność środowisk wirtualnych lub narzędzi analitycznych i odpowiednio dostosowywać swoje zachowanie.*

*Szczególnie trudne do wykrycia są przypadki, gdy Mimikatz jest używany jako część większego arsenału narzędzi atakujących. W takich sytuacjach narzędzie może być dynamicznie ładowane do pamięci z innych procesów, wykorzystywać techniki DLL injection lub process hollowing, a także implementować złożone mechanizmy komunikacji z innymi komponentami złośliwego oprogramowania.”<sup>i</sup>*

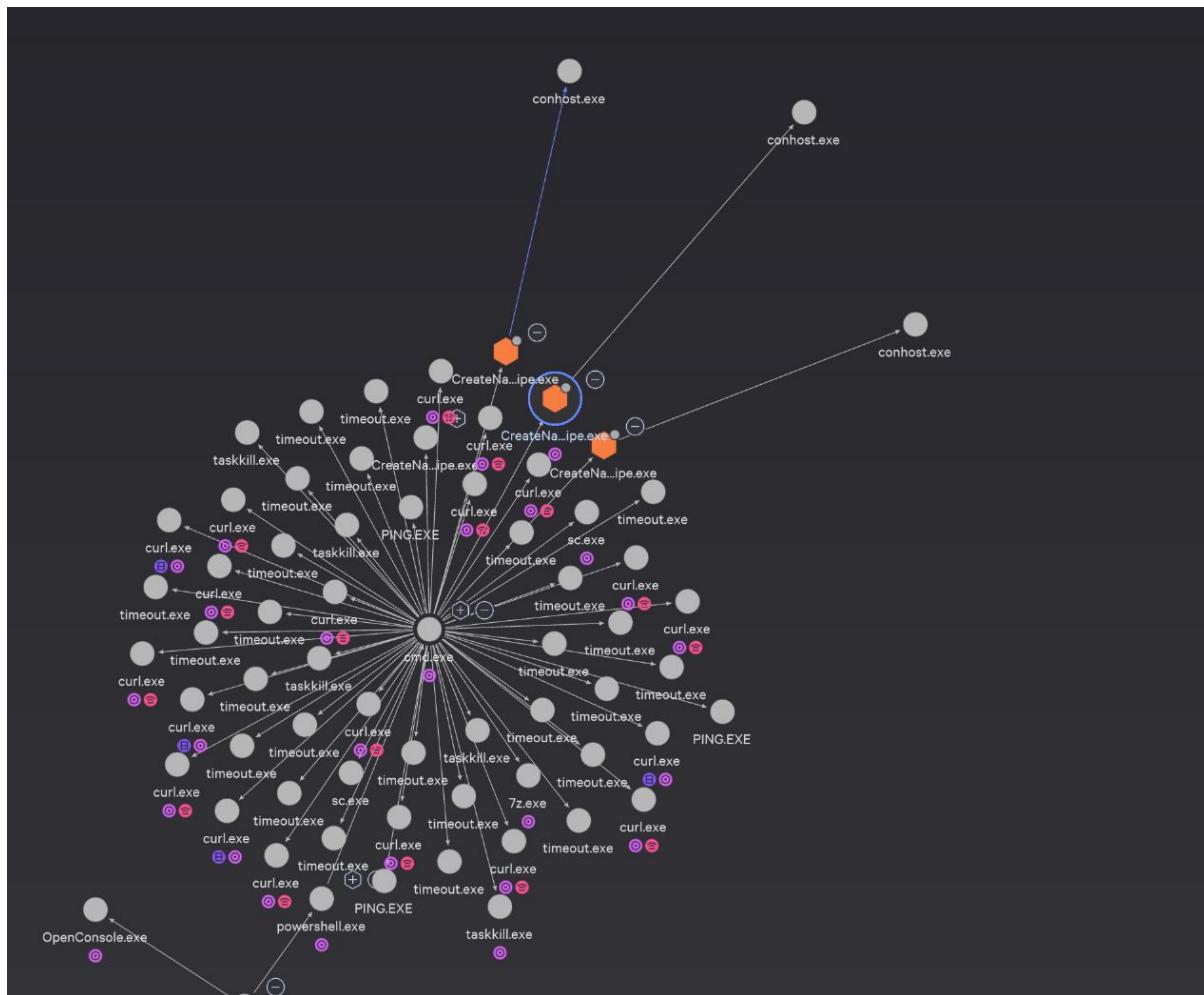
#### Odpowiedź nr. 5

Jedną taką technikę już omawialiśmy (Persistence via Create Account) i do niej porównamy kolejną - Persistence via Image File Execution Options Injection z wielu ataków typu Persistence w tym incydencie.

Powtarzając o pierwszej: Celem było utworzenie nowego, ukrytego konta użytkownika i dodanie go do grupy administratorów lokalnych. Technika ta, realizowana za pomocą *net.exe* lub *powershell.exe*, zapewnia atakującemu prosty, zdalny dostęp do systemu, o ile konto nie zostanie wykryte i usunięte. Jest to jednak metoda stosunkowo "głośna", gdyż nowe konto administratora jest stosunkowo łatwe do wykrycia.

Drugą, znacznie bardziej zaawansowaną i skrytą techniką była iniekcja za pomocą klucza rejestru Image File Execution Options (IFEO), służąca do porwania pliku *sethc.exe* (Klawisze Trwale). Używając narzędzia *reg.exe* – podmienił oryginalny program na *cmd.exe*. W efekcie, pięciokrotne naciśnięcie klawisza Shift na ekranie logowania systemu Windows uruchamia nie narzędzie ułatwień dostępu, a wiersz poleceń z pełnymi uprawnieniami SYSTEM. Pozwala to na uzyskanie dostępu do powłoki systemowej z najwyższymi uprawnieniami, całkowicie omijając mechanizm uwierzytelniania, bez konieczności znajdowania jakiegokolwiek hasła.

Odpowiedź nr 6.



Rysunek 13 Drzewo procesu CreateNamedPipes

W incydencie beaconing służył jako pasywne nasłuchiwanie na polecenia wewnętrz sieci. Ten typ beaconingu jest używany do ruchu bocznego (Lateral Movement)<sup>ii</sup>. Implant C2, będący w istocie Cobalt Strike SMB Beacon, działał w trybie pasywnego oczekiwania. Jego zadaniem było utworzenie wewnętrz systemu Windows specjalnego kanału komunikacyjnego (NamedPipe). Proces ten nie inicjował żadnej łączności z zewnętrznymi serwerami. Zamiast tego, działał jak nasłuchiwaniec, w sposób cykliczny nasłuchiwał na stworzonym przez siebie kanale, czekając na pojawienie się poleceń od operatora ataku.

Framework to najprawdopodobniej CobaltStrike, komenda "C:\TMP\CreateNamedPipe.exe" **MSSE-1337-server**, "C:\TMP\CreateNamedPipe.exe" msagent fedac123 oraz komenda "C:\TMP\CreateNamedPipe.exe" **334485** zawierają w sobie **indykatory**, który jest typowy dla tego frameworku. Tworzone kanały przez ten framework nazywają się na msagent-, msse-, postex-3<sup>iii</sup>, (1337 to wiadomo:p).

---

<sup>i</sup> Źródło: <https://nflo.pl/baza-wiedzy/co-to-jest-mimikatz-dzialanie>

<sup>ii</sup> Źródło: <https://falcon.eu-1.crowdstrike.com/documentation/detections/tactic/lateral-movement-ta0033>

<sup>iii</sup> Źródło: <https://wazuh.com/blog/detecting-cobalt-strike-beacons-using-wazuh>