KRYCY
Część projektowa 2

Borkowski Mateusz
Gryka Paweł
Popiołek Paweł
Wawrzyńczak Michał

16 marca 2022

Spis treści

1 Zadanie

Na podstawie danych przekazanych z Fazy 1 należy zrealizować następujące zadania:

- Przeprowadzenie zadań analitycznych na przekazanym materiale
- Odgadnięcie techniki (technik) wraz ze wskazaniem w matrycach MITRE
- Przedstawienie ciągu przyczynowo-skutkowego prowadzącego do uprawdopodobnienia hipotezy co do techniki (technik) widocznych w materiale źródłowym
- Oszacowanie poziomu ufności co do zaklasyfikowania technik (technik)
- Zamodelowanie hipotetycznego Kill Chain z wykorzystaniem danej techniki lub wskazanie na jeden znany cyberatak wraz z Kill Chain (analogicznie jak robiliście w Fazie 1 przy PoC ataku)

2 Otrzymane próbki

Do analizy otrzymaliśmy 3 pliki:

- audit.log
- auth.log
- pcap.pcap

Najbardziej przydatne okazały się **auth.log** oraz **pcap.pcap**.

3 Rekonstrukcja ataku

host atakujących - host o IP=192.168.56.107
host ofiary - host o IP=192.168.56.104

Sygnatura czasowa	**Zdarzenie**
2021-11-17 00:48:50	uruchomienie pliku *./Desktop/scriptV2.sh*
2021-11-17 00:49:01	uruchomienie *keylogger.py* i przekierowanie jego wyniku przez TCP na port 80 do hosta atakujących, ustanowienie sesji TCP między hostami ofiary i atakującego
2021-11-17 00:49:01	nawiązanie połączenia FTP przez TCP na port 21 do hosta atakujących
2021-11-17 00:49:01	uzyskanie przez hosta atakujących kontroli nad shell hosta ofiary przez FTP
2021-11-17 00:49:10 2021-11-17 00:49:29	wydanie polecenia (na klawiaturze ofiary) *sudo ls -la* oraz przesyłanie przez TCP (w ramach wcześniej ustalonej sesji) do hosta atakujących dwóch ciągów znaków: **sudo ls -la .** oraz **zst1493**
2021-11-17 00:49:41	wydanie polecenia *ls -la* przez host atakujących przez wcześniej ustaloną sesję FTP i otrzymanie w odpowiedzi wyniku tego polecenia - wypisania zawartości folderu
2021-11-17 00:50:45	wydanie polecenia *openssl enc -aes-256-cbc -in lol -out lol_encrypted* przez host atakujących przez wcześniej ustaloną sesję FTP oraz wpisanie hasła *hahaxd* - zaszyfrowanie pliku *lol* za pomocą AES256 z hasłem *hahaxd* do pliku wynikowego *lol_encrypted*
2021-11-17 00:51:06	wydanie polecenia *ls* przez host atakujących przez wcześniej ustaloną sesję FTP oraz otrzymanie odpowiedzi. Prawdopodobnie jest to sprawdzenie czy udało się stworzyć zaszyfrowany plik
2021-11-17 00:51:11	wydanie polecenia *rm lol* przez host atakujących przez wcześniej ustaloną sesję FTP. Usunięcie wcześniej zaszyfrowanego pliku
2021-11-17 00:51:11	wydanie polecenia *sudo ls /* przez host atakujących przez wcześniej ustaloną sesję FTP i otrzymanie odpowiedzi *sudo: no tty present and no askpass program specified*

3.1 Interpretacja po otrzymaniu wiedzy z scriptV2.sh

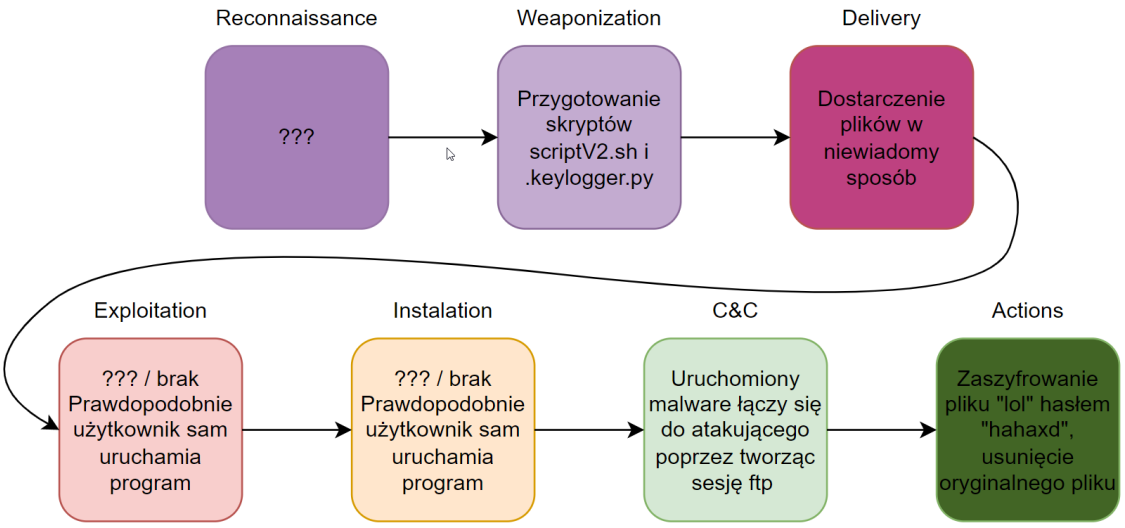
Po wstępnej interpretacji otrzymaliśmy dodatkowo plik *scriptV2.sh*. Dzięki niemu poznaliśmy treść *keylogger.py* oraz dowiedzieliśmy się, że włączenie keyloggera oraz inicjalizacja połączeń TCP zostały zapisane w *crontab* i to on wykonuje je co minutę. Dzięki takiemu zabiegowi keylogger oraz kontrola przez FTP mają szansę przetrwać restart systemu.

3.2 Komunikacja C2 otrzymana z ruchu sieciowego

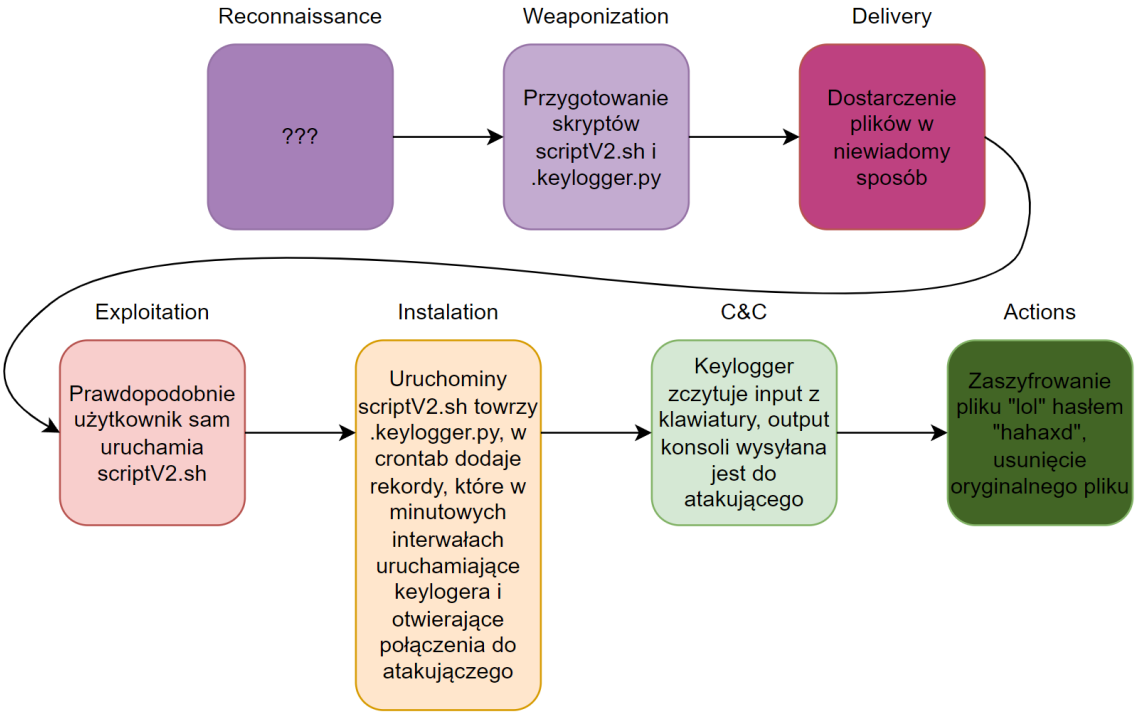
```
bash: cannot set terminal process group (2566): Inappropriate ioctl for device
bash: no job control in this shell
student@netlab-A:~$ ls -la
ls -la
total 888
drwxr-xr-x 24 student    student      4096 lis 17 00:21 .
drwxr-xr-x  4 root       root         4096 lut 28  2020 ..
-rw-------  1 student    student     13709 lis 17 00:36 .bash_history
-rw-r--r--  1 student    student       220 lut 28  2020 .bash_logout
-rw-r--r--  1 student    student      3771 lis 12 20:53 .bashrc
drwx------ 16 student    student      4096 lis 15 21:41 .cache
drwx------ 14 student    student      4096 wrz  4 23:47 .config
drwx------  3 root       root         4096 cze 21 12:46 .dbus
drwxr-xr-x  2 student    student      4096 lis 17 00:48 Desktop
drwxrwxr-x  4 student    student      4096 wrz  5 19:04 Development
drwxr-xr-x  2 student    student      4096 lut 28  2020 Documents
drwxr-xr-x  2 student    student      4096 lis 14 16:06 Downloads
-rw-r--r--  1 student    student      8980 lut 28  2020 examples.desktop
drwx------  3 student    student      4096 lis 12 12:11 .gnupg
drwxrwxr-x  2 student    student      4096 wrz  5 21:52 HostDrive
-rw-------  1 student    student     16054 lis 17 00:20 .ICEauthority
-rw-r--r--  1 root       root        46392 lis 16 22:05 install-snoopy.log
-rwxr-xr-x  1 student    student     20455 lis 16 22:05 install-snoopy.sh
-rw-rw-r--  1 student    student       444 lis 12 18:33 keylogger.py
-rw-rw-r--  1 student    student       497 lis 12 19:23 keyloggerV2.py
drwxrwxr-x  2 student    student      4096 wrz  5 21:59 Literatura
drwx------  6 student    student      4096 wrz  4 23:49 .local
-rw-rw-r--  1 student    student         8 lis 12 19:40 lol
-rw-------  1 root       root          78 cze 21 22:19 .mininet_history
drwx------  5 student    student      4096 gru 28  2020 .mozilla
drwxr-xr-x  2 student    student      4096 lut 28  2020 Music
drwxr-xr-x  2 student    student      4096 wrz  7 19:49 Pictures
drwx------  3 student    student      4096 wrz  4 23:22 .pki
-rw-r--r--  1 student    student       807 lut 28  2020 .profile
drwxr-xr-x  2 student    student      4096 lut 28  2020 Public
-rw-------  1 student    student        78 lis 12 21:23 .python_history
-rw-rw-r--  1 student    student        66 lis 12 20:30 .selected_editor
-rwxrwxr-x  1 student    student       450 lis 14 16:50 slcd_create.sh
-rwxrwxr-x  1 student    student       104 lis 14 16:45 slcd_delete.sh
drwx------  3 student    student      4096 wrz  4 23:22 snap
drwxrwxr-x 11 instructor instructor   4096 lis 16 22:05 snoopy-2.4.15
-rw-r--r--  1 root       root       617952 pa.. 17 05:26 snoopy-2.4.15.tar.gz
drwx------  2 student    student      4096 wrz  4 23:34 .ssh
-rw-r--r--  1 student    student         0 lut 28  2020 .sudo_as_admin_successful
drwxr-xr-x  2 student    student      4096 lut 28  2020 Templates
-rw-r-----  1 student    student         5 lis 17 00:20 .vboxclient-clipboard.pid
-rw-r-----  1 student    student         5 lis 17 00:20 .vboxclient-display-svga-x11.pid
-rw-r-----  1 student    student         5 lis 17 00:20 .vboxclient-draganddrop.pid
-rw-r-----  1 student    student         5 lis 17 00:20 .vboxclient-seamless.pid
drwxr-xr-x  2 student    student      4096 lut 28  2020 Videos
-rw-------  1 root       root         9793 lis 16 22:03 .viminfo
drwxrwxr-x  3 student    student      4096 wrz  4 23:22 .vscode
-rw-rw-r--  1 student    student       254 lis 16 22:05 .wget-hsts
student@netlab-A:~$ openssl enc -aes-256-cbc -in lol -out lol_encrypted
openssl enc -aes-256-cbc -in lol -out lol_encrypted
```

```
enter aes-256-cbc encryption password:hahaxd

Verifying - enter aes-256-cbc encryption password:hahaxd

*** WARNING : deprecated key derivation used.
Using -iter or -pbkdf2 would be better.
student@netlab-A:~$ ls
ls
Desktop
Development
Documents
Downloads
examples.desktop
HostDrive
install-snoopy.log
install-snoopy.sh
keylogger.py
keyloggerV2.py
Literatura
lol
lol_encrypted
Music
Pictures
Public
slcd_create.sh
slcd_delete.sh
snap
snoopy-2.4.15
snoopy-2.4.15.tar.gz
Templates
Videos
student@netlab-A:~$ rm lol
rm lol
student@netlab-A:~$ sudo ls /
sudo ls /
sudo: no tty present and no askpass program specified
student@netlab-A:~$
```

4 Model hipotetycznego Kill Chain



Rysunek 1: Kill chain przed otrzymaniem pliku scriptV2.sh



Rysunek 2: Kill chain po otrzymaniem pliku scriptV2.sh

5 Techniki ataku (ze wskazaniem w matrycach MITRE)

- *Resource Development - Develop Capabilities* - atakujący mógł przygotować oraz odpowiednio sparametryzować użyte w ataku skrypty
- *Execution - User Execution: Malicious File* - przypuszczamy że ofiara sama uruchomiła u siebie złośliwy skrypt *scriptV2.sh*
- *Collection - Input Capture: Keylogging* - działający na hoście ofiary keylogger odczytuje hasło do konta *student* (następnie przesyła je do atakującego)
- *Command and Control - Application Layer Protocol: File Transfer Protocols* - komunikacja C2 odbywa się przy pomocy FTP na porcie 21 (*FTP command*)

tcp.port eq 21

No.	Time	Source	Destination	Protoc	Length	Info
63	39.970690	192.168.56.104	192.168.56.107	FTP	72	Request: ls -la
65	39.973599	192.168.56.104	192.168.56.107	FTP	67	Request:
67	39.974472	192.168.56.104	192.168.56.107	FTP	3354	Request: total 888
69	39.975574	192.168.56.104	192.168.56.107	FTP	86	Request: student@netlab-A:~$
81	103.975975	192.168.56.107	192.168.56.104	FTP	118	Response: openssl enc -aes-256-cbc -in lol -out lol_encrypted
83	103.976322	192.168.56.104	192.168.56.107	FTP	117	Request: openssl enc -aes-256-cbc -in lol -out lol_encrypted
85	103.977216	192.168.56.104	192.168.56.107	FTP	67	Request:
87	104.004411	192.168.56.104	192.168.56.107	FTP	104	Request: enter aes-256-cbc encryption password:
94	112.244005	192.168.56.107	192.168.56.104	FTP	73	Response: hahaxd
96	112.244203	192.168.56.104	192.168.56.107	FTP	67	Request:
98	112.244606	192.168.56.104	192.168.56.107	FTP	116	Request: Verifying - enter aes-256-cbc encryption password:
100	117.102831	192.168.56.107	192.168.56.104	FTP	73	Response: hahaxd

Rysunek 3: Zrzut części ruchu sieciowego na porcie 21

- *Impact - Data Encrypted for Impact* - na hoście ofiary zostaje zaszyfrowany plik *lol*

```
Response: openssl enc -aes-256-cbc -in lol -out lol_encrypted
Request: openssl enc -aes-256-cbc -in lol -out lol_encrypted
Request:
Request: enter aes-256-cbc encryption password:
Response: hahaxd
Request:
Request: Verifying - enter aes-256-cbc encryption password:
Response: hahaxd
Request:
Request: *** WARNING : deprecated key derivation used.
Request: student@netlab-A:~$
Response: ls
Request: ls
Request:
Response: rm lol
Request: rm lol
```

Rysunek 4: przebieg szyfrowania pliku *lol* na hoście ofairy

6 Oszacowanie poziomu ufności

Uważamy, że mamy wysoki poziom ufności w zaklasyfikowaniu technik użytych przez atakujących. Tak naprawdę nie wiemy tylko jak plik *sctiptV2.sh* dostał się na host ofiary oraz przez kogo został on wykonany. Niestety obie te niewiadome są bardzo istotnymi elementami analizy. Zakładamy jednak, że grupa, od której otrzymaliśmy próbki "ręcznie" przerzuciła wspomniany plik i założyła, że są już w trzecim etapie killchain i nie zastanawiała się czy skrypt ten został uruchomiony przez użytkownika czy atakującego.

7 Projekt rozwiązania cyberobrony na podstawie analizy IR

Opracowaliśmy sygnaturę detekcji techniki oraz przetestowaliśmy ją na tym materiale źródłowym. Wykorzystaliśmy aplikację napisaną przez nas w trakcie realizacji laboratorium 1, dopisaliśmy dedykowaną regułę YARA do wykrywania analizowanej techniki. Przetestowaliśmy regułę poddając detekcji próbkę zawierającą uruchomiony podczas ataku skrypt.

```
rule keylogger
{
    strings:
        $s1 = "install pyxhook"
        $s2 = "keylogger"
        $s3 = "crontab <<"
        $s4 = "bash -i"
        $s5 = ">& /dev/tcp/"
        $s6 = /\..{1,50}(.py)/

    condition:
        ($s4 and $s5) or ($s3 and $s6) or 3 of ($s*)
}
```

Rysunek 5: Wykrycie złośliwego pliku za pomocą powyższej reguły YARA

8 Podsumowanie i wnioski

8.1 Skrótowy przebieg ataku

1. Napisanie lub przesłanie na hosta ofiary przez atakujących pliku *sctriptV2.sh*.
2. Uruchomienie skryptu przez użytkownika lub atakującego - jeżeli atakujący mieli już na tym etapie dostep do hosta ofiary. Uruchomienie skryptu powoduje dopisanie do pliku *crontab* wywołań keyloggera i połączenia FTP - zagwarantowanie przetrwania C2 i keyloggera po restarcie systemu.
3. Ustanowienie dwóch połączeń TCP z atakującym. Pierwszy do przesyłania outputu z keyloggera, drugi do shella dla atakującego przez FTP.
4. Zaszyfrowanie pliku *lol* a następnie usunięcie go.

8.2 Wnioski

1. Otrzymanie dodatkowej informacji w postaci pliku *scriptV2.sh* wbrew pozorom nie dało nam wiele nowych, przydatnych informacji. Poznaliśmy jedynie treść keyloggera i fakt dopisania do *crontab* wywołań keyloggera i połączeń. Nie były to dla nas bardzo ważne informacje ponieważ już wcześniej wiedzieliśmy co robi keylogger (w zrzucie ruchu sieciowego można było zauważyć przesyłane przez niego dane) i widzieliśmy, że próby połączenia wykonują się równo co minutę.
2. Brak jakiegokolwiek innego ruchu sieciowego bardzo ułatwił analizę pcap.
3. Czytanie logów wcale nie musi być nudne a wręcz można poczuć się jak Sherlock Holmes.
4. Stanowczo przesadziliśmy z rozbudowaniem etapu pierwszego gry projektowej i współczujemy grupie, która otrzymała do przeanalizowania naszą pracę.
5. Mogą wystąpić niepewności w rekonstruowaniu kroków atakujących ze względu na brak wiedzy na temat wszystkiego co się stało przed i w trakcie ataku.