

6. Übung

Im Folgenden werden Übungen beschrieben, welche das Betreiben, die Auswertung und die Visualisierung von Honeypots beinhaltet.

Der Ablauf sieht wie folgt aus:

1. Cowrie:

- a) Installation
- b) Konfiguration
- c) Test
- d) Auswertung

2. Erkennung von Honeypots:

- a) Charakteristika von Honeypots
- b) nmap
- c) masscan
- d) Censys & Shodan

6.1 Aufgabe 1

Diese Aufgabe befasst sich mit der Installation und Konfiguration des Honeypots Cowrie. Außerdem wird auf die Auswertung der Daten eingegangen.

Für diese Aufgabe benötigen Sie die bekannte und bereits häufig verwandte Kali-Linux VM. Allerdings sollten Sie diese Übungen auch auf allen anderen Betriebssystemen mit Docker-Installation und einem SSH-Client durchführen können.

6.1.1 A) Installation

Cowrie ist ein Honeypot für SSH- und Telnet-Verbindungen mit mittlerer bis hoher Interaktionsmöglichkeit, der darauf ausgelegt ist, Brute-Force-Angriffe sowie die Shell-Interaktionen von Angreifern zu protokollieren. Im Modus der mittleren Interaktion (Shell) emuliert es ein UNIX-System in Python. Im Modus der hohen Interaktion (Proxy) fungiert es als SSH- und Telnet-Proxy, um das Verhalten von Angreifern in Bezug auf ein anderes System zu beobachten.

Zur Installation von Cowrie müssen Sie zunächst `docker-compose` installieren. Hierzu verwenden Sie die Anleitung unter <https://docs.docker.com/compose/install/>. Beachten Sie hierbei, dass Sie nicht das `docker-compose-plugin`, welches mit `docker compose` ausgeführt wird installieren sollen, da dieses unter Kali schwerer zu installieren ist, und in bei diesem experimentellen Setup keine Vorteile bietet. Daher installieren Sie bitte

das Paket `docker-compose`, welches dann auch wie unten zu sehen ist, mit dem Befehl `docker-compose` verwendet wird.

Laden Sie sich danach das `zip`-Archiv mit vorbereiteten Konfigurationsdateien und dem `docker-compose.yml`-File herunter. Entpacken Sie das Archiv und wechseln Sie mit der Konsole in das Verzeichnis.

Überprüfen Sie nun, die Berechtigungen der Ordner `/config` und `/logs`. Diese sollten auf `777` gesetzt sein. Falls dies nicht der Fall ist, können Sie die Berechtigungen mit dem folgenden Befehl setzen:

```
1 sudo chmod 777 -R ./config ./logs
```

Sie können zur vereinfachung der Nutzung noch Ihren Nutzer der Kali-VM (vermutlich `kali`) zur `docker`-Gruppe hinzufügen, so dass Sie nicht für jeden `docker-compose` und `docker` Befehl `sudo` verwenden müssen. Hierzu verwenden Sie die Befehle:

```
1 sudo usermod -aG docker kali
2 newgrp docker
```

Dann können Sie zur Ausführung von Cowrie den folgenden Befehl verwenden:

```
1 docker-compose up
```

Dadurch wird Cowrie gestartet und bietet in der Folge seinen Honeypot-Dienst auf dem Port 21234 der Kali-VM an. Bevor sie Jedoch die Konfiguration von Cowrie anpassen, sollten Sie zunächst die Funktionsweise von Cowrie testen. Hierzu können Sie sich mit dem Benutzer `test` und dem Passwort `test` auf dem Port 21234 der Kali-VM anmelden. **Achtung:** Lassen Sie hierzu die Konsole mit dem laufenden Cowrie-Prozess geöffnet, und öffnen Sie eine weitere Konsole, um sich mit dem Honeypot zu verbinden.

Um sich dann mit dem Honeypot zu verbinden, verwenden Sie den folgenden Befehl:

```
1 ssh test@root@localhost -p 21234
```

Nach der Anmeldung sollten Sie sich in einer Shell befinden, welche der Shell eines Linux-Systems ähnelt. An dieser Stelle haben Sie ausreichend die Funktionalität getestet um zunächst die Konfiguration von Cowrie anzupassen.

Beenden Sie daher die SSH-Verbindung zum Honeypot (mit `STRG + D`), und die laufende Cowrie-Instanz mit der Tastenkombination `STRG + C`.

6.1.2 B) Konfiguration

Das `docker-compose.yml`-File mapped den Ordner `/config/` aus dem Archiv in das Verzeichnis `/cowrie/cowrie-git/etc/` des Containers. An dieser Stelle erwartet Cowrie die Konfigurationsdateien. Daher können Sie die Konfiguration von Cowrie direkt in diesem Ordner vornehmen. Allerdings ist ein Neustart des Containers notwendig, um die Änderungen zu übernehmen.

In dem Ordner `/config/` finden Sie die zwei relevantesten Dateien für die Konfiguration von Cowrie:

- `cowrie.cfg` - Die Hauptkonfigurationsdatei von Cowrie
- `userdb.txt` - Die Benutzerdatenbank von Cowrie

cowrie.cfg

Öffnen Sie die Datei `cowrie.cfg` mit einem Texteditor Ihrer Wahl. In dieser Datei finden Sie die Konfiguration von Cowrie. Hier können Sie beispielsweise sowohl Änderungen am *frontend* (also der SSH-Umgebung) als auch am *backend* (also Logspeicherung, Modus Operandi oder den Speicherorten heruntergeladener Dateien) vornehmen.

Sie werden feststellen, dass die Konfigurationsdatei sehr umfangreich ist. Allerdings ist sie mit Kommentaren versehen, sodass Sie sich gut zurechtfinden sollten. Im Folgenden wird Ihnen aufgetragen einige Konfigurationen anzupassen.

- ändern Sie die Zeit, bis eine SSH-Verbindung bei Inaktivität getrennt wird auf 240 Sekunden.
- passen Sie den Hostnamen des Honeypots an, also den Namen, welcher einem Angreifer bei eingehender SSH-Verbindung des systems angezeigt wird.
- ändern Sie Parameter der Konfiguration so ab, dass die Ausgabe des `uname -a` Befehls den gleichen Output wie auf Ihrer Kali-VM hat.

userdb.txt

Nun haben Sie bisher nur die Konfiguration des Honeypots im Bezug auf Back- und Frontend angepasst. Was Sie allerdings noch machen sollten ist die Datei `userdb.txt` anzupassen. In dieser Datei werden die Benutzerkonten-Passwort Kombinationen gespeichert, welche für die Anmeldung am Honeypot verwendet werden können. Hierbei gibt es die folgende Syntax:

- Die Syntax sind aus durch Doppelpunkt (:) getrennten Felder, welche zeilenweise verarbeitet werden.

- **Die Verarbeitung stoppt bei der ersten Übereinstimmung.**
- Die Felder sind wie folgt definiert:
 1. **Feld #1** enthält den Benutzernamen.
 2. **Feld #2** wird aktuell nicht verwendet.
 3. **Feld #3** enthält das Passwort.
- Wie üblich Bedeutet ein **!** Negierung und ***** eine Wildcard.
- Ein Schrägstrich (**/**) kann verwendet werden, um einen regulären Ausdruck zu schreiben.

Beispiele für Einträge sind:

- **root:x:!root** – Der Benutzer **root** kann sich nicht mit dem Passwort **root** anmelden.
- **special:x:*** – Der Benutzer **special** kann sich mit jedem Passwort anmelden.
- ***:x:somepassword** – Jeder Benutzer kann sich mit dem Passwort **somepassword** anmelden.

Verändern Sie nun die Datei **userdb.txt** so, dass die folgenden Kriterien erfüllt sind:

- Der Benutzer **root** darf sich nicht mit dem Passwort **iamgroot** anmelden, davon abgesehen ist keine Anmeldung mit dem Nutzer **root** erlaubt.
- Der Benutzer **cooladmin** darf sich mit allen Passwörtern (abgesehen von anderen Einschränkungen) anmelden.
- Abgesehen von allen anderen Einschränkungen sollen alle Benutzer sich mit dem Passwort **admin123** anmelden können.
- **Niemand** darf sich mit dem Passwort **honeypot** anmelden.

6.1.3 C) Test

Nachdem Sie die Konfiguration von Cowrie angepasst haben, können Sie den Honeypot erneut starten. Hierzu verwenden Sie den Befehl aus Aufgabe 1.A). Nachdem der Honeypot gestartet ist, können Sie sich erneut mit dem Honeypot verbinden. Testen Sie hierbei ob die Anmeldung mit den Benutzern wie von Ihnen konfiguriert funktioniert. Außerdem können Sie testen, ob Ihre Konfiguration abgesehen davon erfolgreich war.

Dann können Sie weiter den Honeypot verwenden und experimentieren wie detailgetreu die Implementierung ist.

6.1.4 D) Auswertung

Nachdem Sie den Honeypot getestet haben, können Sie die Daten auswerten. Hierzu können Sie die Datei `cowrie.json` im Ordner `/logs/` verwenden. Diese Datei enthält alle Daten, welche von Cowrie aufgezeichnet wurden. Sie können diese Datei mit einem Texteditor öffnen und die Daten einsehen. Allerdings ist die Datei sehr umfangreich und unübersichtlich. Daher verwenden Sie das Tool `cowrie-logviewer`, welches Sie bereits mit dem ausführen des `docker-compose.yml`-Files installiert und gestartet haben. Diese Tool bietet Ihnen nun ein graphisches Interface in Ihrem Browser an. Die Adresse des Interfaces ist `http://localhost:5000/`.

6.2 Aufgabe 2

Diese Aufgabe befasst sich mit der Erkennung von Honeypots. Hierzu werden Sie verschiedene Methoden verwenden, um Honeypots zu erkennen. Sie werden hierbei sowohl die Erkennung von Honeypots in einem Netzwerk als auch die Erkennung von Honeypots im Internet durchführen.

Für diese Aufgabe benötigen Sie erneut Ihre Kali-Linux VM.

6.2.1 A) Charakteristika von Honeypots

Je nachdem wie Honeypots implementiert sind, können sie verschiedene Charakteristika aufweisen. So antworten manche Honeypots beispielsweise häufig mit einer charakteristischen Fehlermeldung, einem charakteristischen Banner oder einer charakteristischen Antwort auf einen Befehl. Diese Charakteristika können Sie verwenden, um Honeypots zu erkennen. Diese Technik ist allerdings von einem stetigen Katz-und-Maus-Spiel geprägt, da Honeypots häufig angepasst werden, um diese Charakteristika zu verbergen.

Häufig werden mehrere Honeypots auf einer IP betrieben. Hierbei kann es dann vorkommen, dass die Anzahl und die Kombination der verschiedenen Services auf einer IP nicht realistisch ist. Auch kann eine spezielle Kombination von Services auf einer IP charakteristisch für einen Honeypot sein.

Häufig sind auch einfache IP-Adressen mit vielen offenen Ports und Services ein Indiz für einen Honeypot.

Ihre Aufgabe ist es nun einfache zu findende Charakteristika für standard-Installationen des bekannten Honeypot-Frameworks T-pot¹ zu finden. Hierzu ist bereits das `README.md` des Projektes eine sehr gute Quelle.

¹<https://github.com/telekom-security/tpotce>

Stellen Sie nun zwei Kombination von Ports (Ports&Services) zusammen, welche Sie für gute Erkennungszeichen für T-pot halten. Hierbei soll eine Kombination mit vielen Ports aber ohne Services funktionieren und eine Kombination mit lediglich zwei Ports und Services.

6.2.2 B) nmap

Nmap ist ein bekanntes Tool zur Erkundung von Netzwerken und zur Erkennung von offenen Ports und Services. In der Vergangenheit hatte Nmap auch zeitweise die Fähigkeit einzelne angebotene Services als Honeypots zu erkennen. Dies geschah allerdings mit der Erkennung von Technischen Besonderheiten in den Antworten/dem Verhalten des Services, welche dann von den Entwicklern wieder angepasst wurden. Daher kann man hier davon ausgehen, dass die meisten Honeypot-Services nicht mehr von Nmap erkannt werden.

Allerdings kann es möglich sein mit Nmap Aufklärung über die angebotenen Services zu erhalten, welche dann in der Kombination auf einen Honeypot hinweisen können.

Hierzu nutzen Sie den folgenden Befehl um zunächst auf Ihren Cowrie Honeypot zu scannen (dieser sollte hierzu laufen).

```
1 nmap -sV -p21234 localhost
```

Hier werden Sie feststellen, dass Nmap den Honeypot nicht als solchen erkennt.

Mit dem folgenden Befehl würden Sie nun den gesamten möglichen Portbereich einer IP-Adresse scannen. Er dauert einige Zeit und ist daher nur bei der Untersuchung einzelner IP-Adressen sinnvoll.

```
1 nmap -sV -p1-65535 localhost
```

6.2.3 C) masscan

Masscan ist ein Tool, welches ähnlich wie Nmap funktioniert. Allerdings ist es deutlich schneller als Nmap, da es asynchron arbeitet, also nicht zunächst eine Antwort eines Ports/Ips abwartet, bevor es den nächsten Port/IP scannt, sondern zwei Threads startet. Wobei einer der beiden Threads lediglich Pakete sendet und der andere nur Pakete empfängt.

Leider ist Masscan trotz der Ähnlichkeiten zu Nmap und der sich teils deckenden Syntax nicht Feature-identisch. Daher kann Masscan nicht die Service-Discovery von Nmap abbilden.

Daher kann man entweder alleine an Charakteristischen Kombinationen von offenen Ports mit masscan auf Honeypots schließen, oder für die offenen Ports dann auf die Service-Discovery von Nmap zurückgreifen.

Um masscan zu verwenden, können Sie den folgenden Befehl verwenden:

```
1 sudo masscan -p1-65535 127.0.0.1
```

Beachten Sie an dieser Stelle, dass Masscan mit erhöhten Rechten ausgeführt werden muss.

6.2.4 D) Censys & Shodan

Censys² und Shodan³ sind Suchmaschinen für das Internet. Diese Suchmaschinen scannen das Internet nach offenen Ports und Services und speichern diese Informationen in einer Datenbank. Vereinfacht gesagt finden Sie bei Shodan und Censys also eine Datenbank mit Ergebnissen der zuvor besprochenen Scans masscan und nmap.

Diese Datenbanken können Sie nun verwenden, um Honeypots zu finden. Hierzu können Sie die Suchmaschinen verwenden, um nach charakteristischen Kombinationen von offenen Ports und Services zu suchen. Außerdem können Sie nach charakteristischen Bannern suchen, welche auf einen Honeypot hinweisen.

Darüber hinaus stellt Shodan sowohl als auch Censys ihre Interpretation der Daten insofern zur Verfügung, dass Sie nach Honeypots suchen können. Selbstverständlich unterliegt auch diese Honeypot-Erkennung dem Katz und Maus Spiel aus Erkennung und Anpassung.

²<https://search.censys.io/>

³<https://shodan.io>