Honeypot Setup Report

Internet Storm Center Internship

Initial Setup

The honeypot was deployed on a Raspberry Pi 5 using a 128GB microSD card. The SD card came pre-installed with the Raspberry Pi OS, eliminating the need for an initial image installation.

Once connected to a monitor and booted up, I followed the provided setup guide and executed the install script. After rebooting, I enhanced logging capabilities by modifying configuration files:

- In `/etc/dshield.ini`, I added:
  `localcopy=/tmp/local.log`
- In `/srv/cowrie/cowrie.cfg`, I updated:
  `ttylog = true` (previously `false`)

With these changes complete, I ran the `status.sh` script to verify that all services were functioning properly. At this point, I discovered that the web server was not being exposed externally.

Troubleshooting Exposure Issues

Initial debugging revealed that my Xfinity router was blocking external access despite the honeypot appearing functional locally. To address this, I:

1. Accessed the Xfinity router's management app.
2. Configured port forwarding for the Raspberry Pi on the following ports:
   `22, 23, 43, 80, 443, 445, 8080`

After rebooting both the router and the Raspberry Pi, the web server issue was resolved.

However, I later discovered that Cowrie SSH logs were still not being sent to the ISC. I reviewed the Cowrie GitHub repository and determined that outbound firewall rules may have been blocking log transmission despite port forwarding. To isolate the honeypot and safely expose it, I created a **DMZ (Demilitarized Zone)**.

DMZ & Firewall Configuration

To safely assign the honeypot to a DMZ:

- I set a static IP address for the Raspberry Pi to avoid accidental IP reassignment.
- I restricted SSH access to only my PC by adding iptables rules:

  ```bash
  sudo iptables -A INPUT -p tcp --dport 12222 -s <my.pc.ip> -j ACCEPT
  sudo iptables -A INPUT -p tcp --dport 12222 -j DROP
  ```

- I also configured a static IP on my PC to ensure secure and persistent access.

Despite these changes, logs remained inconsistent. After consulting ChatGPT, I inspected the router's firewall logs and found over 600 entries blocking outbound traffic from the honeypot—specifically on ports 80 and 443.

Given the limited customization on the Xfinity router (only preset security levels), I ordered a **GL.iNet router** with advanced controls. Temporarily lowering the firewall settings validated the issue, as logs began transmitting. The new router allowed me to apply firewall rules solely to the honeypot without risking the rest of the network.

Honeypot Environment

The environment consists of:

- **Cowrie:**
  An SSH and Telnet honeypot that supports medium interaction, capable of logging brute-force attempts and capturing sessions.

  - Downloads by attackers are stored in:
    `/srv/cowrielib/cowrie/downloads`
    (Files are named using SHA-256 hashes, suitable for analysis on VirusTotal.)

  - Successful sessions are logged in:
    `/srv/cowrie/var/lib/cowrie/tty`
    (Replayable via Python scripts for attack analysis.)

- **DShield Sensor:**
  Simulates network services (e.g., SSH, HTTP) and sends data to the ISC, helping identify malicious patterns.

Future Enhancements

To enhance visibility and threat detection, I plan to incorporate the following tools:

- **tcpdump:**
  For capturing raw packet data to analyze source/destination patterns.

- **Wireshark:**
  For visual analysis of packet flows and payload inspection.

- **Snort:**
  For real-time intrusion detection, rule-based filtering, and signature matching to identify anomalous activity such as non-crypto miner malware or advanced tactics.

Conclusion

The honeypot deployment posed a variety of challenges—from restrictive router settings to outbound firewall filtering. Overcoming these obstacles required careful networking and security configuration, along with tailored iptables and firewall rule implementation. The decision to upgrade the router proved essential in enabling persistent log collection without exposing other devices.

As the environment matures with enhanced logging and detection tools, it will serve as a robust platform for observing attacker behavior and crafting detailed analytical reports on emerging threats.