**AWS Honeypot – Week 1 Observation Report**

**📅 Date**

2025-11-30: 00:00AM 🡪 2025-12-7: 23:30:00.000AM

**Background: What are honeypots and why are they important?**

A honeypot is a controlled decoy system, network, or data resource that is designed to attract, trap, and analyze cyber-attacks. They are used by organizations to provide valuable threat intelligence without exposing their actual assets. By analyzing the various attacks, security professionals can identify new threats, possible vulnerabilities, and attack patterns, which will inform future security policies and counteractive measures to protect sensitive data.

T-Pot (Threat Intelligence Pot) is an open-source honeypot platform that supports over 20 honeypot frameworks such as Cowrie, Dionaea, and Honeytrap. It simplifies data analysis by providing multiple visualization options like live attack maps, and all other necessary tools and documentation needed to build a honeypot system. It also offers versatility in its implementation, able to be installed from a multitude of clouds and hosts including Microsoft Azure, GCP, and AWS, making it an ideal and accessible learning simulation.

**Document Summary and Table of Contents**

This document is a report of the first week of activity of the T-Pot Honeypot following its setup on November 30th at approximately 10:00 AM. By analyzing patterns and significant activity, this report aims to provide insight into peak attack periods, common attack sources, and provide insight into future cyber-attacks.

This report will differentiate between the types of Honeypots represented in this simulation, analyze findings, and provide predictions on anticipated trends over the next week

**Key Events:**

Cloudflare Outage: A ~25-minute network issue impacted many customers, stemming from Cloudflare's attempt to patch an industry-wide React Server Components vulnerability, not a direct cyberattack. On December 5, 2025, from 08:47 UTC to 09:12, a portion of Cloudflare’s network began to experience significant failures where approximately 28% of all HTTP traffic served by CloudFlare impact. This was caused by Cloudflare’s direct attempt to protect customers against a critical vulnerability, CVE-2025-55182 where turning off the WAF testing tool led to a bug leading to a Lua exception.

Freedom Mobile Breach: Personal customer data was exposed in a significant data breach. Hackers stole key PII including names, addresses, phone numbers, DOBs, and account numbers.

Brazil Health System Attack: The Nova ransomware group targeted and compromised Brazil's National Primary Health System.

**Defining the Top 10 HoneyPots Based on Attacks (As of 12/7/2025)**

**Analysis & Predictions:** Based on my preliminary research and gathered data, I predict the low-interaction honeypots that simulate vulnerable network services on common ports would attract the most attacks. Low-interaction honeypots typically capture broad, automated scanning activity because they expose widely targeted services with minimal complexity. Because high-interaction honeypots simulate more sophisticated and realistic environments, they would likely attract fewer but more sophisticated and targeted attackers. Given that most malicious traffic originates from automated tools like nmap that scan common vulnerable ports, I expect Dionaea, HoneyTrap, and Cowrie to continue receiving the highest attack volumes throughout the simulation.

**1). Cowrie: 176k**

Another commonly used, cybersecurity tool, Cowrie is a medium to high interaction SSH and Telnet Honeypot which is designed to log brute force attacks and the shell interaction performed by the attackers. It emulates a full UNIX system environment, including a fake filesystem to allow attackers to interact as if they have gained access. It logs brute force attempts, the credentials used, the commands typed during the SSH or Telnet session, the files downloaded/downloaded with wget/curl or SFTP/SCP. Collected data can be integrated with event management software like Splunk of MS Sentinel to provide threat intelligence and alerts

**2). Dionaea: 145k**

A widely used, low-interaction cybersecurity tool designed primarily to trap and collect malware samples by emulating vulnerable services across a network and Windows environments. It’s primary purpose to capture the actual malware that binary attackers use in their exploitation attacks. It logs hash values such as MD5 from the files for further analysis and intelligence gathering. It emulates common protocols like SMB, FTP, HTTP, MySQL, SIP, MSSQL to attract attacks that target specific, well-known vulnerabilities. Collected attack logs can be ingested through analysis platforms like Splunk or MHN for visualization.

**3). HoneyTrap: 68k**

Another low-interaction honeypot written to catch attacks against TCP and UDP services. It starts when a connection attempt to a port is made, running as a daemon thread. It provides multiple operation modes that vary the level of defense and analysis the user wants. In normal mode, the server sends the data formatted as a basic template. For a more defensive approach, incoming connections are proxied back to the initiator. Finally, for more of a high-interaction approach, the proxy mode allows the forwarding of sessions to high-interaction honeypots.

**4). Heralding: 2k**

A credentials capturing honeypot. Developed by johnnykv, Heralding captures relevant log data that is split into 3 files: log session data, login authentication data, and log session data.

**5). TANNER/SNARE: 2k**

SNARE (Super Next Gen Advanced Reactive Honeypot) is a web app honeypot sensor that attracts malicious activity. Events (HTTP requests) in SNARE are sent to TANNER, a remote data analysis and classification service that decides how SNARE should respond to the client.

**6). Mailoney: 2k**

A low interaction SMTP honeypot developed in python that simulates a vulnerable mail server to detect and log unauthorized access attempts.

**7). Cisco ASA: 1k**

A low-interaction, open-source honeypot designed to detect a DoS and remote code execution vulnerability called CVE-2018-0101.

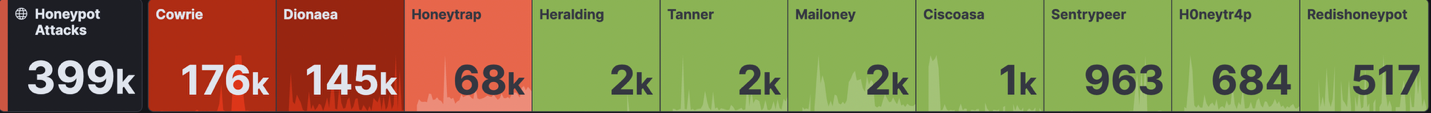
**9). H0neytr4p:**

An open-source, easily configurable honeypot designed to protect against web recon and exploitation.

**10). Redis Honey Pot:**

A security decoy system that mimics a vulnerable/misconfigured Redis (Remote Dictionary Server) instance to attract, detect, and store data on cyberattacks. It supports basic commands like PING GET, and KEYS. Because Redis is a popular in-memory data store that stores cryptocurrency, the Redis Honey Pot is especially useful for analyzing attack patterns and methods to protect against cryptocurrency mining malware and backdoors. A Redis Honeypot alerts security teams to unauthorized scanning and brute force attempts on its ports and logs all client interactions.

**Network Traffic Summary**

****

Observations: Most of the attacks (**176,019**) were directed against the **Cowrie**, constituting ~**44.1%** of the overall Honeypot attacks within the 7-day period. This is followed by **Dionaea** with **144,674** attacks (**36.3%**) and **HoneyTrap** with **67,891** attacks (**17.4%**). It makes sense that the Dionaea honeypot would receive the most attacks due to their low-interaction nature. Because Dionaea emulates a Windows environment with commonly used protocols, it is likely the most accessible for scanning attacks and CVE exploitations. The fact that the Cowrie honeypot attracted the most attacks is somewhat surprising since it requires medium-high levels of interaction, where attackers try to brute-force their way into systems remotely either using ssh or telnet. This is uncharacteristic since most attackers try to attack as many systems as possible.

|  |  |
| --- | --- |
| **Analysis** | **Figure** |
| During this 7-day period, attacks seemed to peak daily around 21:00, 12:00, 0:00, and 09:00. These patterns suggest:  1). Business Hours Attack Strategy (0900 & 12:00): Attackers target when systems are under heaviest legitimate use, which is harder to detect. SOC teams may also be overwhelmed with alerts during peak usage. Alternatively, at 12:00, workers may be on lunch break meaning there is reduced monitoring over systems  2). Shift Changes (00:00 & 21:00): Security team shift changes which means there is reduced monitoring over systems. A general increase of attacks after 18:00 indicates that attackers capitalize on after-business hours when IT staff have left  3). Geographic distribution: These times correspond to optimal attack windows across multiple time zones.  One outlier is the peak in attacks from 09:00 to 18:00. After doing research on recent cybersecurity events, specifically on December 5, I discovered that at approximately 09:00, many users experienced service interruptions due to CloudFlare server outages. The peak in attacks can likely be explained by attackers trying to capitalize on user and server vulnerabilities and the distraction of IT workers. | Figure 1: Honeypot Attacks Histogram  *A graph on a black background  AI-generated content may be incorrect.* |
| Dionaea receives the highest number of attacks consistently overall and is responsible for the large spikes visible in the traffic data. Unlike Honeytrap and the other honeypots, which experience a steady and consistent rate of attacks, Dionaea shows periods of intense, burst-like activity. These spikes are likely caused by large-scale automated scans or botnet sweeps targeting the common ports and protocols that Dionaea emulates. Because Dionaea simulates a broad range of widely exploited services, it naturally attracts high-volume, automated attacks that occur during peak scanning activity on the internet. However, the reason why Cowrie experienced the most attacks during this 7-day period was due to the Cloudflare Outage on December 5th. This is consistent with Figure 1, which shows most of the attacks were leveraged during and after the outage. The sudden spike of ssh attacks is contrasted by a sudden drop in Dionaea attacks, during which automated scanners interpreted Cloudflare issues as potential security weaknesses and pivoted IP-based scanning. Because the WAF protection was down, previously protected IPs became visible. Consequently, the vulnerable servers became higher value targets. This reveals that botnets have event-triggered scanning protocols and that infrastructure incidents are actively monitored by threat actors | *Figure 2: Attacks by Honeypot Histogram*  **A screenshot of a graph  AI-generated content may be incorrect.** |
| **The spikes in attack volume appear to correlate strongly with specific countries. While the United States consistently generates the highest baseline number of attacks, all represented countries show noticeable spikes during certain time periods. This pattern suggests that large, coordinated scanning events or botnet activities are distributed globally, with multiple regions contributing to the bursts in traffic. The consistent volume from the U.S. may reflect the high concentration of cloud infrastructure and compromised hosts located there, whereas other countries show more intermittent, spike-driven activity. The different spike times may be attributed to time-zone. For instance, the peak in attacks from Iran occurs daily around 00:00 UTC, indicating that bots run automatic scans on a daily schedule. Interestingly, 31,553 attacks (nearly 91% of the attacks) originated from US IP addresses. Keeping in mind these attacks primary targeted SSH/Telnet, this suggests:**  **1). Compromised US Infrastructure: There are botnets using US-based servers from cloud providers such as AWS and Azure**  **2). Automated Cloud Scanning Tools: Shodan/Censys-style automated scanners may have been triggered, US -based security companies were doing vulnerability assessments, or researchers/gray-hat scanners were scanning for vulnerabilities.** | Figure 3: Attacks by Country Histogram  A screenshot of a graph  AI-generated content may be incorrect. |
| Port 445 is associate with the Server Message Block Protocol (SMB), which is used to facilitate file and printer sharing on windows-based networks. It makes sense that it received the most attacks since this port is critical for internal network communications. Other commonly attacked ports include 80, which is associated with HTTP traffic, and 22, the default protocol assigned to SSH, these are all commonly used ports and essential to day-to-day operations. Consistent with my previous observations, there is a sudden peak in ssh attacks on December 5th. | Figure 4: Attacks by Destination Port Histogram  A screenshot of a graph  AI-generated content may be incorrect. |
| Most of the attacks seem to originate in the Middle East, western Europe, the United States, and Southeast Asia. | Figure 5:Dynamic Attack Map  A map of the world with dots  AI-generated content may be incorrect. |
| Top 10 Countries by Attack Percentages:   1. United States ~60% (Expected: 5-10%) 2. China ~ 7% (Expected 30-40%) 3. Moldova ~6% (Expected <1%) 4. Iran/South Korea ~ 5% 5. India/Vietnam ~4% 6. Tunisia/Indonesia/Bolivia ~3%   In these findings, the United States accounting for most attacks is unprecedented, but can largely be attributed to the CloudFlare outage. This may be an indicator of massive cloud infrastructure abuse, where many cloud instances were deployed for malicious scanning purposes. Surprisingly, Russia didn’t make the top 10 countries although it normally appears in the top 5 countries responsible for cyberattacks and accounts for approximately 15-25% of attacks against honeypot data, according to trustworthy sources. It is also unusual that China only accounts for 7% of honeypot attacks. A possible explanation is that Russia, China, and other countries are using US proxies and/or VPNs | Figure 6: Attacks by Country  A screenshot of a graph  AI-generated content may be incorrect. |
| Interestingly, most attacks that originate from each country targets a specific port. For instance, many of the attacks that originate from the United States and Moldova target port 22 (SSH), China and Iran target (SMB), while attacks from South Korea targets port 5920 (VNC) and 5900 (RFB). According to ResearchGate and other OSINT sources, SSH is the favorite port for many attacks, so it is unsurprising that many attacks target SSH. SMB is also a common and high-value target since attackers can exploit vulnerabilities to gain unauthorized access, execute code remotely, or escalate privileges. Commonly exploited vulnerabilities include CVE-2017-0144 (EternalBlue) and CVE-2020-0796. Attacks against VNC involve attempts to exploit remote access vulnerabilities to gain access to systems. These attack patterns may be in response to vulnerable servers that were exposed by the loss of the WAF during the CloudFlare outage. | Figure 7: Attacks by Country & Port  A screenshot of a graph  AI-generated content may be incorrect. |
| It also appears that most of the attacks originate from a known attackers and mass scanners, indicating that many attackers are repeat offenders. | A pie chart with text on it  AI-generated content may be incorrect. |
| Suricata is an open-source Network Security Monitoring engine. It provides IDS alerts, tracks protocol transactions, network flows, PCAP recordings, and extracted files. Strangely, large peaks occur from 18:00 to 03:00 from November 30 to December 1st. This may be coincidental or an indicator of malicious activity. Overall, most suricata alerts are Misc Activity alerts and misc attacks. Misc Activity generally refers to alerts of port scanning, unusual protocol behavior, and non-standard service probing. Attempted Administrator Privilege Gain alerts. Otherwise, attacks are quite stable and low. These patterns may be counting on employees being out of office for breaks and ends of shift to try to gain access to administrator privileges while remaining unnoticed. This may be why activity during regular business hours is lower. | Figure 8: Suricata Alert Category Histogram  A screenshot of a graph  AI-generated content may be incorrect. |
| Interestingly, the commonly usernames and passwords used in these attacks are very simple, often one word or predictable number sequences. This is common among credential stuffing and brute-force attacks. Attackers hope to capitalize on insecure servers and resources. This is why it is essential that access credentials are changed from default credentials, and that users use secure usernames and passwords and servers/services often require a longer combination of letters, numbers, and special characters to increase security. Input validation is especially important, where GET requests and Html language may indicate attempts to perform sequel injection. | Figure 9: Password and Username Tagclouds  A screenshot of a computer screen  AI-generated content may be incorrect. |

**More Suricata Findings:**

What is Suricata?

Suricata is an open-source network analysis and threat detection engine used as an IDS (intrusion detection system), IPS (Intrusion Prevention System), and a network security monitoring program. It monitors network traffic for threats and logs network vulnerabilities and malicious behavior.

**Top Exploited CVEs (Suricata)**

What are CVEs?

* Dubbed “The Common Vulnerabilities and Exposures Program”, CVE was created by the National Vulnerability Database (NIST) to deal with common problems
* Each CVE is a vulnerability found in software and hardware components that can result in a negative impact to confidentiality, integrity, or availability when exploited.
* Mitigation methods generally involve code changes, specification changes, or even the removal of affected protocols or functionality.
* The National Vulnerability Database (NVD) hosts a list of common CVEs and their descriptions and origins.
* CVEs have an associated CVSS (Common vulnerability scoring system that rates the threat level based on a variety of factors)

|  |  |
| --- | --- |
| 1.[CVE-2006-2369](https://nvd.nist.gov/vuln/detail/CVE-2006-2369)  Describes vulnerabilities in products that use RealVNC which allows remote attackers to bypass authentication via a request in which a client specifies an insecure security type, which is accepted even if it is not offered by the server  - CVSS Base Score: 7.5 (high)  **2.** [CVE-2021-3449](https://nvd.nist.gov/vuln/detail/CVE-2021-3449) & [CVE-2021-3449](https://nvd.nist.gov/vuln/detail/CVE-2021-3449)  Describes the event where an OpenSSL TLS server is liable to crash due to a maliciously crafted renegotiation ClientHello message from a client. Affects all OpenSSL 1.1.1 versions   * CVSS Base Score: 5.9 (med)   **3.** [CVE-2019-11500](https://nvd.nist.gov/vuln/detail/CVE-2019-11500) & [CVE-2019-11500](https://nvd.nist.gov/vuln/detail/CVE-2019-11500)  Describes the event where the mishandling of quoted string ‘\0’ can lead to out-of-bounds writes and remote code execution in Dovecot (IMAP Server)   * Base Score: 9.8 (critical)   4.[CVE-2002-0013](https://nvd.nist.gov/vuln/detail/CVE-2002-0013) & [CVE-2002-0012](https://nvd.nist.gov/vuln/detail/CVE-2002-0012)  Describes vulnerabilities related to handling requests & processing in Microsoft IIS, a web server application. Base Score: 10 (critical)   * [CVE-2002-0013](https://nvd.nist.gov/vuln/detail/CVE-2002-0013): Allowed an attacker to send specially crafted HTTP requests to IIS servers, potentially leading to buffer overflows/execution of arbitrary code (User input handling issue) * [CVE-2002-0012](https://nvd.nist.gov/vuln/detail/CVE-2002-0012): Allowed an attacker to execute arbitrary code by exploiting a buffer overflow (Indexing service issue)   **5.** [CVE-2024-4577](https://nvd.nist.gov/vuln/detail/CVE-2024-4577) & [CVE-2002-0953](https://nvd.nist.gov/vuln/detail/CVE-2002-0953)  - [CVE-2024-4577](https://nvd.nist.gov/vuln/detail/CVE-2024-4577): Describes an error in specific PHP versions where windows replaces characters in the command line interface to Win32 API functions, which allows users to see the source code of scripts and run arbitrary PHP code on the server  - [CVE-2002-0953](https://nvd.nist.gov/vuln/detail/CVE-2002-0953): Allows remote attackers to include functions to load data from a remote URL instead of just local files  **- Base Score: 9.8 (critical)** **5.** [CVE-2021-41773](https://nvd.nist.gov/vuln/detail/CVE-2021-41773): Flaw in the Apache HTTP Server 2.4.49 path normalization functionality. Attackers were able to use a path traversal attack to map URLs to files outside the directories and execute code remotely. Fix is incomplete in Apache HTTP Server 2.4.50**6.** [CVE-2021-42013](https://nvd.nist.gov/vuln/detail/CVE-2021-42013): Found the fix for the CVE-2021-41773 was insufficient in Apache Server 2.4.50.**7.** [CVE-2023-46604](https://nvd.nist.gov/vuln/detail/CVE-2023-46604): Vulnerability that allows remote code execution within Java OpenWire. Where remote attacker with network access to either a OpenWire broker or client to run shell commands**8.** [CVE-2009-2765](https://nvd.nist.gov/vuln/detail/CVE-2009-2765): Describes a command injection vulnerability in DD-WRT (linux-based router firmware) where the web server httpd.c was taking user-supplied input from the URL and passing it to the Linux shell without sanitizing, allowing OS command injection | Figure 10: Suricata CVEs  **A screenshot of a computer  AI-generated content may be incorrect.**    Analysis: Common Trends seen within the top 10 CVEs  Based on these findings, attackers are exploiting vulnerability in web servers to initiate remote code execution in older, out-of-date software models. Most of these Suricata alerts are caused by automated scanners probing old, well-known web server vulnerabilities. This is why it’s essential that devices do not run unpatched embedded firmware and to follow security best practices. The most exploited CVE, CVE-2006-2369 allows VNC authentication bypass for complete desktop access and administrator controls without credentials. |

**Suricata Alert Signature- Top 10**

|  |
| --- |
| **A screenshot of a computer program  AI-generated content may be incorrect.** |
| **GPL INFO VNC Server Response:** An alert that fires when Suricata detects a VNC server responding to connection attempts. This is triggered during the VNC protocol handshake.   1. Client connects to port 5900 (or 5901-5920). 2. VNC server responds with: “RFB 003.008\n” or similar version   **ET EXPLOIT [PTsecurity] DoublePulsar Backdoor installation communication:** The most prevalent attack signature is that of DoublePulsar Backdoor, or the infamous EthernalBlue exploit, which was used in the WannaCry ransomware attack. The EternalBlue exploits a vulnerability in Microsoft Windows systems to install a backdoor on the target system, allowing them to maintain persistent access and control over the compromised machine. It’s therefore no surprise that this is the most used exploit as so many Windows machines were susceptible to compromise.  **ET DROP Dshield Block Listed Source group 1**: A rule that blocks traffic originating from IP addresses on a known block list maintained by DShield.  **ET SCAN NMAP -sS window 1024**: A signature used to detect port scanning activity, specifically those conducted with the Nmap tool using a TCP SYN scan (-sS) with a window size of 1024 bits.  **ET EXPLOIT Possible CVE-2020-11899 Multicast out-of-bound read**: Used to detect attempts to exploit a vulnerability in Windows Hyper-V that could allow a remote attacker to execute arbitrary code.  **SURICATA STREAM Packet with broken ack**: Detects TCP packets with invalid acknowledgement (ACK) numbers, which could indicate attempts to disrupt network connections or conduct denial-of-service attacks.  **ET SCAN MS Terminal Server Traffic on Non-standard Port**: Detects attempts to scan for Microsoft Terminal Server traffic on non-standard ports.  **ET POLICY Reserved Internal IP Traffic**: A rule that detects network traffic containing reserved IP addresses, which could indicate attempts to probe or exploit internal networks.  **ET INFO User-Agent (python-requests) Inbound to Webserver**: Notes inbound web traffic with a specific user-agent string indicating the use of the Python requests library. |

**Attacker AS/N – Top 10**

**A screenshot of a computer

AI-generated content may be incorrect.**

**Attacker IP Addresses**

**A screenshot of a computer

AI-generated content may be incorrect.**