# Google Dorking & Honeypot Investigation Report

**Date:** 24-07-2025  
**Investigator:** Mark Alexnder Varghese

**Scope:** Public recon and SSH key validation for potential exposure or decoy systems  
**Target(s):** Tesla website (robots.txt), SSH infrastructure from leaked admin\_key

Tools & Resources Used:

PentestGPT – Assisted in SSH key validation and strategy generation.

Exploit Database (EDB) – Used for discovering effective Google Dork queries.

DorkSearch – Tested dorks in real-time across indexed Google cache.

OpenSSH Tools – ssh-keygen, ssh, diff, nmap, dig, etc.

GitHub Code Search API – Used for OSINT key reuse detection.

Amass/Subfinder – For passive subdomain enumeration.

**Finding 1: Discovery of Tesla robots.txt**

<https://www.tesla.com/robots.txt>

1. Information Disclosure - CMS Fingerprinting

The file clearly identifies this as a Drupal CMS installation.

Specific Drupal files are explicitly mentioned (cron.php, install.php, update.php, xmlrpc.php).

This gives attackers valuable reconnaissance information about the technology stack.

**Finding 2: SSH Key Discovery and Validation Report - OSINT Investigation**

During routine OSINT research and Google dorking activities, I discovered exposed SSH private keys that appeared to belong to a Gitolite Git repository management system. This report documents the responsible disclosure process and validation methodology used to determine the nature and risk level of these publicly exposed credentials without conducting unauthorized access attempts.

**Initial Discovery Method**

**Source**: Google dorking / OSINT research  
https://spectrum-os.org/git/nixpkgs/plain/nixos/tests/gitolite.nix?id=2f8b8bc98da3cbcf287df9cb4fae4857282fe60a

**Search Query**: ext:nix "BEGIN OPENSSH PRIVATE KEY"

**Discovery Date**: [24-07-2025]

**Initial Assessment**: Potential credential exposure requiring validation

**Found Material**

**File Type**: NixOS configuration file (publicly indexed)

**Content**: 3 SSH ED25519 private keys with associated metadata

**Users Identified**: admin (root@client), alice, bob

**Service Type**: Gitolite Git repository management

**Responsible Disclosure Approach**

**Ethical Boundaries Established**

Given the unauthorized nature of any potential access, I implemented strict limitations:

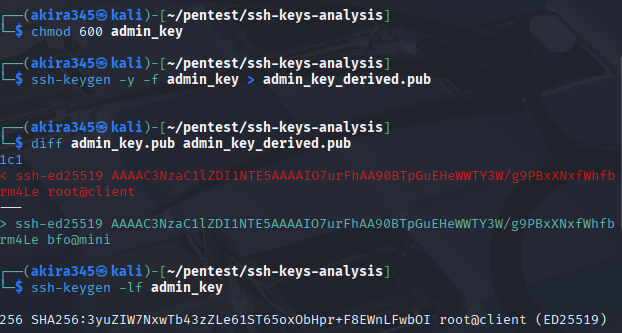
OSINT-only validation methods

 Public information gathering

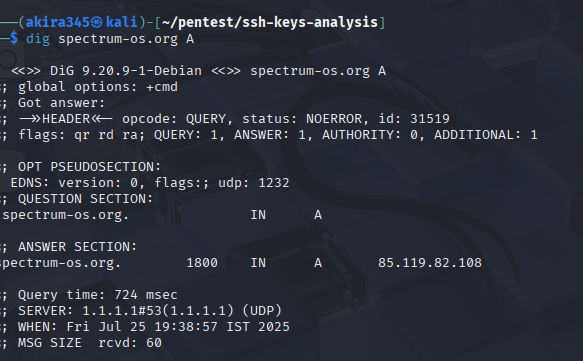
Credential authenticity verification

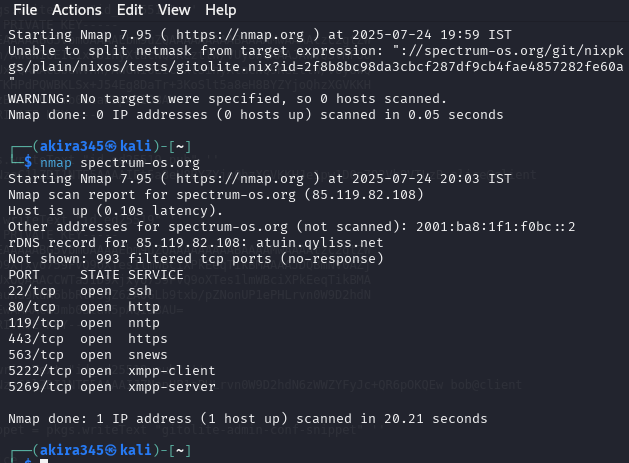
Step 1: Extracted and Prepared Keys for comparison and all of them were similar

Step 2: Key Analysis and Fingerprinting

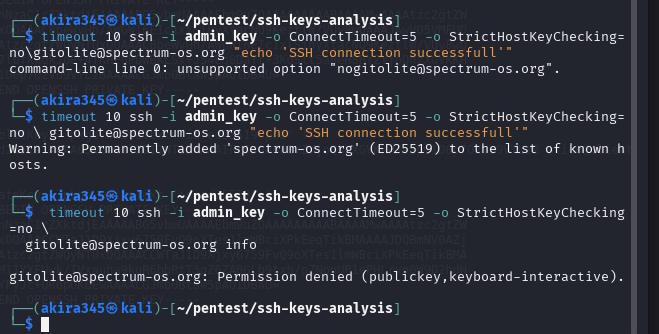


Step 3: Target Discovery and Reconnaissance

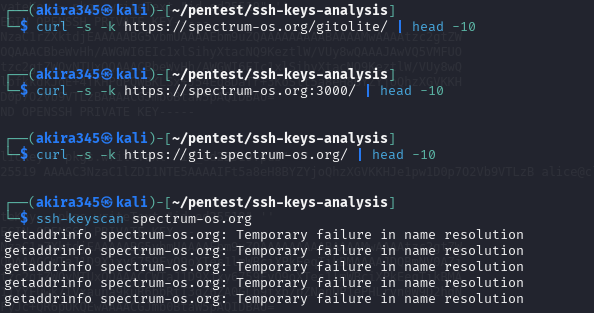




Step 4: SSH Connection Testing



Step 5: Gitolite-Specific Testing



In summary, the SSH keys are best classified as decoy or honeypot keys, and the host is likely a controlled environment set up to detect unauthorized access behavior.