OpenCanary

OpenCanary Honeypot - Quick Start

===================================

Installation (Ubuntu/Debian):

```
$ sudo apt update
$ sudo apt install python3-pip python3-dev libssl-dev libpcap-dev -y
$ sudo pip3 install opencanary
$ sudo mkdir -p /etc/opencanary /var/log/opencanary
$ sudo opencanaryd --copyconfig
```

Basic Commands:

```
$ sudo opencanaryd --start        # Test mode (foreground)
$ sudo opencanaryd --daemon       # Production mode
$ sudo opencanaryd --stop         # Stop daemon
$ tail -f /var/log/opencanary/opencanary.log
```

First Test:

From another machine: `nmap -sV <your_honeypot_IP>`

Check logs for the scan!

Troubleshooting:

- Permission denied? Use sudo
- Port conflict? Change port in config or stop conflicting service
- JSON errors? Validate at jsonlint.com
- Not seeing traffic? Check firewall rules

Advanced OpenCanary Configurations
====================================

Custom Banners:

Make your honeypot more convincing with realistic banners:

```
"http.banner": "Apache/2.4.41 (Ubuntu)",
"ssh.version": "SSH-2.0-OpenSSH_8.2p1 Ubuntu-4ubuntu0.5",
"ftp.banner": "vsFTPd 3.0.3",
```

Credential Pairs for Telnet:

```
{
    "username": "admin",
    "password": "admin123"
},
{
    "username": "support",
    "password": "D-Link123"
}
```

Multiple Instances:

Run different honeypots on different IPs using Docker:

```
$ docker run -d --name iot-honeypot \
  -v /path/to/iot-config.json:/etc/opencanary/opencanary.conf \
  opencanary
```

Logging to Syslog:

Add to your config:

```
"logger": {
    "class": "PyLogger",
    "kwargs": {
        "handlers": {
            "syslog": {
                "class": "logging.handlers.SysLogHandler",
                "address": ["192.168.1.100", 514]
            }
        }
    }
}
```

Integration with Wazuh:

1. Configure syslog output (above)
2. Add Wazuh rules for OpenCanary events
3. Create alerts for specific attack patterns
4. Use active response to auto-block attackers

Corporate File Server Lure

Attracts: Ransomware operators, data exfiltration, corporate espionage

```
{
    "device.node_id": "corp-fileserver-01",
    "ip.ignorelist": [],
    "git.enabled": false,
    "git.port": 9418,
    "ftp.enabled": true,
    "ftp.port": 21,
    "ftp.banner": "Microsoft FTP Service",
    "http.banner": "Apache",
    "http.enabled": true,
    "http.port": 80,
    "http.skin": "nasLogin",
    "httpproxy.enabled": false,
    "httpproxy.port": 8080,
    "httpproxy.skin": "squid",
    "logger": {
        "class": "PyLogger",
        "kwargs": {
            "formatters": {
                "plain": {
                    "format": "%(message)s"
                }
            },
            "handlers": {
                "file": {
                    "class": "logging.FileHandler",
                    "filename": "/var/log/opencanary/corp-fileserver.log"
                }
            }
        }
    },
    "portscan.enabled": true,
    "smb.auditfile": "/var/log/opencanary/smb_audit.log",
    "smb.enabled": true,
    "mysql.enabled": false,
    "mysql.port": 3306,
    "mysql.banner": "5.5.43-0ubuntu0.14.04.1",
    "ssh.enabled": true,
    "ssh.port": 22,
    "ssh.version": "SSH-2.0-OpenSSH_7.4",
    "redis.enabled": false,
    "rdp.enabled": true,
    "rdp.port": 3389,
    "sip.enabled": false,
    "snmp.enabled": false,
    "ntp.enabled": true,
    "ntp.port": 123,
    "tftp.enabled": false,
    "tcpbanner.maxnum": 10,
    "tcpbanner.enabled": false,
    "telnet.enabled": false,
    "mssql.enabled": true,
    "mssql.version": "2012",
    "mssql.port": 1433,
    "vnc.enabled": false
}
```

Why this works: SMB + FTP + fake NAS login + MSSQL mimics a corporate file/database server. Attracts ransomware gangs doing recon, lateral movement tools, and credential stuffing.

IoT/Embedded Device Cluster

Attracts: IoT botnets (Mirai variants), cryptominers, DDoS recruit attempts

```
{
    "device.node_id": "iot-camera-lobby",
    "ip.ignorelist": [],
    "git.enabled": false,
    "ftp.enabled": false,
    "http.banner": "IPC-HX3300",
    "http.enabled": true,
    "http.port": 80,
    "http.skin": "basicLogin",
    "httpproxy.enabled": false,
    "logger": {
        "class": "PyLogger",
        "kwargs": {
            "formatters": {
                "plain": {
                    "format": "%(message)s"
                }
            },
            "handlers": {
                "file": {
                    "class": "logging.FileHandler",
                    "filename": "/var/log/opencanary/iot-devices.log"
                }
            }
        }
    },
    "portscan.enabled": true,
    "smb.enabled": false,
    "mysql.enabled": false,
    "ssh.enabled": true,
    "ssh.port": 22,
    "ssh.version": "SSH-2.0-dropbear_0.52",
    "redis.enabled": false,
    "rdp.enabled": false,
    "sip.enabled": true,
    "sip.port": 5060,
    "snmp.enabled": true,
    "snmp.port": 161,
    "ntp.enabled": true,
    "ntp.port": 123,
    "tftp.enabled": true,
    "tftp.port": 69,
    "tcpbanner.maxnum": 10,
    "tcpbanner.enabled": true,
    "tcpbanner_1.enabled": true,
    "tcpbanner_1.port": 8080,
    "tcpbanner_1.datareceivedbanner": "",
    "tcpbanner_1.initbanner": "RTSP/1.0 200 OK",
    "tcpbanner_1.alertstring.enabled": false,
    "tcpbanner_1.keep_alive.enabled": false,
    "tcpbanner_1.keep_alive_secret": "",
    "tcpbanner_1.keep_alive_probes": 11,
    "tcpbanner_1.keep_alive_interval": 300,
    "tcpbanner_1.keep_alive_idle": 300,
    "telnet.enabled": true,
    "telnet.port": 23,
    "telnet.banner": "Welcome to IoT Device",
    "telnet.honeycreds": [
        {
            "username": "admin",
            "password": "admin"
        },
        {
            "username": "root",
            "password": "root"
        },
        {
            "username": "admin",
            "password": "12345"
        }
    ],
    "mssql.enabled": false,
    "vnc.enabled": false
}
```

Why this works: Telnet with weak creds, RTSP banner, SIP, SNMP, and Dropbear SSH screams "vulnerable IoT camera/DVR." Catches Mirai, Gafgyt, and similar IoT botnet scanners. The weak telnet creds are deliberate bait.

Development/Staging Server

Attracts: Web app attacks, supply chain compromise attempts, credential harvesters, crypto miners

```
{
    "device.node_id": "dev-staging-02",
    "ip.ignorelist": [],
    "git.enabled": true,
    "git.port": 9418,
    "ftp.enabled": false,
    "http.banner": "nginx/1.18.0",
    "http.enabled": true,
    "http.port": 8000,
    "http.skin": "basicLogin",
    "httpproxy.enabled": false,
    "logger": {
        "class": "PyLogger",
        "kwargs": {
            "formatters": {
                "plain": {
                    "format": "%(message)s"
                }
            },
            "handlers": {
                "file": {
                    "class": "logging.FileHandler",
                    "filename": "/var/log/opencanary/dev-staging.log"
                }
            }
        }
    },
    "portscan.enabled": true,
    "smb.enabled": false,
    "mysql.enabled": true,
    "mysql.port": 3306,
    "mysql.banner": "5.7.33-0ubuntu0.18.04.1",
    "ssh.enabled": true,
    "ssh.port": 22,
    "ssh.version": "SSH-2.0-OpenSSH_8.2p1 Ubuntu-4ubuntu0.5",
    "redis.enabled": true,
    "redis.port": 6379,
    "rdp.enabled": false,
    "sip.enabled": false,
    "snmp.enabled": false,
    "ntp.enabled": false,
    "tftp.enabled": false,
    "tcpbanner.maxnum": 10,
    "tcpbanner.enabled": true,
  "tcpbanner_1.enabled": true,
  "tcpbanner_1.port": 5000,
  "tcpbanner_1.datareceivedbanner": "",
  "tcpbanner_1.initbanner": "Flask Development Server",
  "tcpbanner_1.alertstring.enabled": false,
  "telnet.enabled": false,
  "mssql.enabled": false,
  "vnc.enabled": false
}
```

Why this works: Git daemon + Redis + MySQL + Flask dev server on 5000 + nginx signals exposed development infrastructure. Attracts attackers looking for unprotected Redis instances, exposed .git directories, and misconfigured dev environments. Common target for cryptominers and supply chain attacks.

Legacy Windows Server

Attracts: Vulnerability scanners, EternalBlue/SMBv1 exploits, legacy system attackers

```
{
    "device.node_id": "legacy-dc-backup",
    "ip.ignorelist": [],
    "git.enabled": false,
    "ftp.enabled": true,
    "ftp.port": 21,
    "ftp.banner": "Microsoft FTP Service",
    "http.banner": "Microsoft-IIS/7.5",
    "http.enabled": true,
    "http.port": 80,
    "http.skin": "basicLogin",
    "httpproxy.enabled": false,
    "logger": {
        "class": "PyLogger",
        "kwargs": {
            "formatters": {
                "plain": {
                    "format": "%(message)s"
                }
            },
            "handlers": {
                "file": {
                    "class": "logging.FileHandler",
                    "filename": "/var/log/opencanary/legacy-windows.log"
                }
            }
        }
    },
    "portscan.enabled": true,
    "smb.auditfile": "/var/log/opencanary/smb_audit.log",
    "smb.enabled": true,
    "mysql.enabled": false,
    "ssh.enabled": false,
    "redis.enabled": false,
    "rdp.enabled": true,
    "rdp.port": 3389,
    "sip.enabled": false,
    "snmp.enabled": true,
    "snmp.port": 161,
    "ntp.enabled": true,
    "ntp.port": 123,
    "tftp.enabled": false,
    "tcpbanner.maxnum": 10,
    "tcpbanner.enabled": true,
    "tcpbanner_1.enabled": true,
    "tcpbanner_1.port": 135,
    "tcpbanner_1.datareceivedbanner": "",
    "tcpbanner_1.initbanner": "",
    "tcpbanner_1.alertstring.enabled": false,
    "tcpbanner_2.enabled": true,
    "tcpbanner_2.port": 139,
    "tcpbanner_2.datareceivedbanner": "",
    "tcpbanner_2.initbanner": "",
    "tcpbanner_2.alertstring.enabled": false,
    "tcpbanner_3.enabled": true,
    "tcpbanner_3.port": 445,
    "tcpbanner_3.datareceivedbanner": "",
    "tcpbanner_3.initbanner": "",
    "tcpbanner_3.alertstring.enabled": false,
    "telnet.enabled": false,
    "mssql.enabled": true,
    "mssql.version": "2008R2",
    "mssql.port": 1433,
    "vnc.enabled": false
}
```

Why this works: IIS 7.5, SMB, RDP, MSSQL 2008R2, and ports 135/139/445 scream "old Windows Server 2008." Attracts EternalBlue scanners, BlueKeep exploits, and attackers specifically hunting unpatched legacy systems. The "backup" in the node name makes it extra juicy.