Configuration

Since we have many services, and each service has a few options, we have listed them all for you.

Services Configuration

We have gone ahead and listed all the services by their configuration key with a quick description of the service and when it alerts.

Currently, OpenCanary supports faking the following services natively:

- *ssh*: a Secure Shell server that alerts on login attempts
- *ftp* - a File Transfer Protocol server that alerts on login attempts
- *git* - a Git protocol that alerts on repo cloning
- *http* - an HTTP web server that alerts on login attempts
- *httpproxy* - an HTTP web proxy that alerts when there is an attempt to proxy to another page
- *mssql* - an MS SQL server that alerts on login attempts
- *mysql* - an MYSQL server that alerts on login attempts
- *telnet* - a Telnet server that alerts on login attempts
- *snmp* - an SNMP server that alerts on oid requests
- *sip* - a SIP server that alerts on sip requests
- *vnc* - a VNC server that alerts on login attempts
- *redis* - a Redis server that alerts on actions
- *tftp* - a TFTP server that alerts on requests
- *ntp* - an NTP server that alerts on NTP requests.
- *tcpbanner* - a TCPbanner service that alerts on connection and subsequent data received events.
- *ignorelist* - comma-separated ips or CIDRs that will ignore alerting on.

Please note that each service may have other configurations such as *port*. For example, the *tcpbanner* service has a bunch of extra settings that drastically change the way, the service would interact with an attacker.

The default generated config will include all options, with all services set to *false* (except for *ftp*).

You may also want to fiddle with some of our other services which require a bit more setup;

smb - a log watcher for Samba logging files which allows Opencanary to alert on files being opened in a Windows File Share.

For this configuration, you will need to set up your own Windows File Share. Please read the steps over here.

portscan - a log watcher that works with iptables to monitor when your Opencanary is being scanned. At this stage, the portscan module supports the detection of Nmap OS, Nmap FIN, Nmap OS, Nmap NULL, and normal port scans. *portscan.iptables_path* is available for you to specify the path to your *iptables* binary.

Logger Configuration

Opencanary allows us to define a bunch of logging/alerting sinks. Below are a list of options you can simply add to the *logger* section in your config file,

```
"logger": {
    "class": "PyLogger",
    "kwargs": {
        "formatters": {
            "plain": {
                "format": "%(message)s"
            },
            "syslog_rfc": {
                "format": "opencanaryd[%(process)-5s:%(thread)d]: %(name)s %(levelname)-5s
            }
        },
        "handlers": {
            "console": {
                "class": "logging.StreamHandler",
                "stream": "ext://sys.stdout"
            },
            "file": {
                "class": "logging.FileHandler",
                "filename": "/var/tmp/opencanary.log"
            },
            "syslog-unix": {
                "class": "logging.handlers.SysLogHandler",
                "formatter":"syslog_rfc",
                "address": [
                    "localhost",
                    514
                ],
                "socktype": "ext://socket.SOCK_DGRAM"
            },
            "json-tcp": {
```

```
                    "class": "opencanary.logger.SocketJSONHandler",
                    "host": "127.0.0.1",
                    "port": 1514
                },
                "SMTP": {
                    "class": "logging.handlers.SMTPHandler",
                    "mailhost": ["smtp.yourserver.com", 25],
                    "fromaddr": "noreply@yourdomain.com",
                    "toaddrs" : ["youraddress@gmail.com"],
                    "subject" : "OpenCanary Alert"
                },
                "slack":{
                    "class":"opencanary.logger.SlackHandler",
                    "webhook_url":"https://hooks.slack.com/services/..."
                },
                "teams": {
                    "class": "opencanary.logger.TeamsHandler",
                    "webhook_url":"https://my-organisation.webhook.office.com/webhookb2/..."
                },
                "Webhook": {
                    "class": "opencanary.logger.WebhookHandler",
                    "url": "http://domain.example.com/path",
                    "method": "POST",
                    "data": {"message": "%(message)s"},
                    "status_code": 200
                }
            }
        }
    }
```

Please note that the above are not the only logging options. You can use any Python logging class. The above are the most popular. You can also head over to Email Alerts for more **SMTP** options that require authentication.

You may want to look through some other python logging options over at PyLogger page.

We have provided you with two different formatters. One is the plain message with incident information; the other is the Syslog RFC format. We have already added it to the *syslog-unix* handler for your convenience.

Environment Variables

You can use environment variables in the configuration file to pass confidential information such as passwords or tokens from the host machine to the application.

For example on your host machine you would export your password:

```
export TELNET_PASSWORD=TopsyKretts
```

And in your config file you would reference it by name proceeded by a dollar sign (*$*):

```
"telnet.honeycreds": [
    {
        "username": "admin",
        "password": "$TELNET_PASSWORD"
    }
]
```

> Note: For Windows, you can also use *%TELNET_PASSWORD%*

If you are using the Docker version, you would need to pass the environment variable to the container as well as part of the run command:

```
docker run -e TELNET_PASSWORD ...
```

For Docker Compose, you would need to add it to the service definition:

```
service:
  opencanary:
    image: "..."
    environment:
      - TELNET_PASSWORD
    ...
```

Default Configuration

When you generate the default OpenCanary config file using,

```
$ opencanaryd --copyconfig
```

you will receive a json formatted config file at */etc/opencanary/opencanary.conf* such as the following,

```
{
    "device.node_id": "opencanary-1",
    "ip.ignorelist": [ ],
    "git.enabled": false,
    "git.port" : 9418,
    "ftp.enabled": true,
    "ftp.port": 21,
    "ftp.banner": "FTP server ready",
    "http.banner": "Apache/2.2.22 (Ubuntu)",
    "http.enabled": false,
    "http.port": 80,
    "http.skin": "nasLogin",
    "http.skin.list": [
        {
            "desc": "Plain HTML Login",
            "name": "basicLogin"
        },
        {
            "desc": "Synology NAS Login",
            "name": "nasLogin"
        }
    ],
    "httpproxy.enabled" : false,
    "httpproxy.port": 8080,
    "httpproxy.skin": "squid",
    "httproxy.skin.list": [
        {
            "desc": "Squid",
            "name": "squid"
        },
        {
            "desc": "Microsoft ISA Server Web Proxy",
            "name": "ms-isa"
        }
    ],
    "logger": {
        "class": "PyLogger",
        "kwargs": {
            "formatters": {
                "plain": {
                    "format": "%(message)s"
                }
            },
            "handlers": {
                "console": {
```

```json
                    "class": "logging.StreamHandler",
                    "stream": "ext://sys.stdout"
                },
                "file": {
                    "class": "logging.FileHandler",
                    "filename": "/var/tmp/opencanary.log"
                }
            }
        }
    },
    "portscan.enabled": false,
    "portscan.logfile":"/var/log/kern.log",
    "portscan.synrate": 5,
    "portscan.nmaposrate": 5,
    "portscan.lorate": 3,
    "smb.auditfile": "/var/log/samba-audit.log",
    "smb.enabled": false,
    "mysql.enabled": false,
    "mysql.port": 3306,
    "mysql.banner": "5.5.43-0ubuntu0.14.04.1",
    "ssh.enabled": false,
    "ssh.port": 22,
    "ssh.version": "SSH-2.0-OpenSSH_5.1p1 Debian-4",
    "redis.enabled": false,
    "redis.port": 6379,
    "rdp.enabled": false,
    "rdp.port": 3389,
    "sip.enabled": false,
    "sip.port": 5060,
    "snmp.enabled": false,
    "snmp.port": 161,
    "ntp.enabled": false,
    "ntp.port": "123",
    "tftp.enabled": false,
    "tftp.port": 69,
    "tcpbanner.maxnum":10,
    "tcpbanner.enabled": false,
    "tcpbanner_1.enabled": false,
    "tcpbanner_1.port": 8001,
    "tcpbanner_1.datareceivedbanner": "",
    "tcpbanner_1.initbanner": "",
    "tcpbanner_1.alertstring.enabled": false,
    "tcpbanner_1.alertstring": "",
    "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": false,
        "telnet.port": "23",
        "telnet.banner": "",
        "telnet.honeycreds": [
            {
                "username": "admin",
                "password": "$pbkdf2-sha512$19000$bG1NaY3xvjdGyBlj7N37Xw$dGrmBqqWa1okTCpN3QEmec
            },
            {
                "username": "admin",
                "password": "admin1"
            }
        ],
        "mssql.enabled": false,
        "mssql.version": "2012",
        "mssql.port":1433,
        "vnc.enabled": false,
        "vnc.port":5000
    }
```

Should you have any other questions regarding configuration or setup, please do not hesitate to contact us on GitHub.

OpenCanary

Navigation

HTTP Webhook Alerts

Quick search

	Go