OpenCanary Documentation

Release 0.1

Thinkst Applied Research

Dec 15, 2018

Contents

Welcome to the OpenCanary guide.

OpenCanary is a daemon that runs canary services, which trigger alerts when (ab)used. The alerts can be sent to a variety of sources, including syslog, emails and a companion daemon opencanary-correlator.

The Correlator coalesces multiple related events (eg. individual brute-force login attempts) into a single alert sent via email or SMS.

CHAPTER 1

Getting Started

The first section will get you quickly up and running with canary services sending alerts.

1.1 OpenCanary

1.1.1 Getting Started

To get started create a virtual environment to play in:

```
$ virtualenv env
$ . env/bin/activate
```

Inside the virtualenv, install OpenCanary following the instructions in the README.

OpenCanary ships with a default config, which we'll copy and edit to get started. The config is a single JSON dictionary.

```
$ opencanaryd --copyconfig
$ $EDITOR ~/.opencanary.conf
```

In the config file we'll change **device.node_id** which must be unique for each instance of opencanaryd, and we'll configure **logger** to log alerts to a file.

```
{
    "device.node_id": "Your-very-own-unique-name",
    [...]
     "logger": {
         "class" : "PyLogger",
         "kwargs" : {
             "handlers": {
                 "file": {
                     "class": "logging.FileHandler",
                     "filename": "/var/tmp/opencanary.log"
```

(continues on next page)

(continued from previous page)

```
                    }
                }
            }
        }
    [...]
}
```

With that in place, we can run the daemon, and test that it logs a failed FTP login attempt to the log file.

```
$ opencanaryd --start
[...]
$ ftp localhost
[...]
$ cat /var/tmp/opencanary.log
[...]
{"dst_host": "127.0.0.1", "dst_port": 21, "local_time": "2015-07-20 13:38:21.281259",
↪"logdata": {"PASSWORD": "default", "USERNAME": "admin"}, "logtype": 2000, "node_id
↪": "opencanary-0", "src_host": "127.0.0.1", "src_port": 49635}
```

1.1.2 Troubleshooting

The tool JQ can be used to check that the config file is well-formed JSON.

```
$ jq . ~/.opencanary.conf
```

Run opencanaryd in the foreground to see more error messages.

```
$ sudo env/bin/twistd -noy env/bin/opencanary.tac
```

1.2 Correlator

1.2.1 Getting Started

To get started create a virtual environment to play in:

```
$ virtualenv env
$ . env/bin/activate
```

Inside the virtualenv, install OpenCanary Correlator following the instructions in the README.

The correlator runs with a default config, which we'll copy and edit to get started.

```
$ opencanary-correlator
Warning: no config file specified. Using the template config:
/[...]/opencanary_correlator.conf
$ cp /[...]/opencanary_correlator.conf opencanary-correlator.conf
```

In the config file, fill the twilio or mandrill details (or both), and the notification addresses for both.

```
{
  "console.sms_notification_enable": true,
  "console.sms_notification_numbers": ["+336522334455"],
```

(continues on next page)

(continued from previous page)

```
  "console.email_notification_enable": true,
  "console.email_notification_address": ["notifications@opencanary.org"],
  "twilio.auth_token": "fae9206628714fb2ce00f72e94f2258f",
  "twilio.from_number": ""+1201253234"",
  "twilio.sid": "BD742385c0810b431fe2ddb9fc327c85ad",
  "console.mandrill_key": "9HCjwugWjibxww7kPFej",
  "scans.network_portscan_horizon": 1000,
}
```

With that in place, ensure that redis is running and then run the correlator daemon.

```
$ pgrep redis-server || echo 'Redis is not running!'
$ opencanary-correlator --config=./opencanary-correlator.conf
```

To configure OpenCanary daemons to send their events to correlator, edit the **logger** field in its config and restart the daemon to reload the config.

```
"logger": {
     "class" : "PyLogger",
     "kwargs" : {
         "handlers": {
             "json-tcp": {
                 "class": "opencanary.logger.SocketJSONHandler",
                 "host": "127.0.0.1", # change to correlator IP
                 "port": 1514
             }
         }
     }
}
```

1.2.2 Troubleshooting

You can test that the Correlator alerts are working be sending an event direclty to it (without using OpenCanary).

```
echo '{"dst_host": "9.9.9.9", "dst_port": 21, "local_time": "2015-07-20 13:38:21.
↪281259", "logdata": {"PASSWORD": "default", "USERNAME": "admin"}, "logtype": 2000,
↪"node_id": "AlertTest", "src_host": "8.8.8.8", "src_port": 49635}' | nc -v␣
↪localhost 1514
```

The tool JQ can be used to check that the config file is well-formed JSON.

```
$ jq . ./opencanary-correlator.conf
```

CHAPTER 2

Services

Try these out in the OpenCanary configs for more typical server personalities.

2.1 Linux Web Server

Inside ~/.opencanary.conf:

```
{
 "ftp.banner": "FTP server ready",
 "ftp.enabled": true,
 "ftp.port":21,
 "http.banner": "Apache/2.2.22 (Ubuntu)",
 "http.enabled": true,
 "http.port": 80,
 "http.skin": "nasLogin",
 "http.skin.list": [
     {
         "desc": "Plain HTML Login",
         "name": "basicLogin"
     },
     {
         "desc": "Synology NAS Login",
         "name": "nasLogin"
     }
 ],
 "ssh.enabled": true,
 "ssh.port": 8022,
 "ssh.version": "SSH-2.0-OpenSSH_5.1p1 Debian-4",
[..] # logging configuration
}
```

2.2 Windows Server

The Samba and RDP modules require an extra installation steps. It's a good idea to consult the README before trying this out.

Inside ~/.opencanary.conf:

```
{
    "smb.auditfile": "/var/log/samba-audit.log",
    "smb.configfile": "/etc/samba/smb.conf",
    "smb.domain": "corp.thinkst.com",
    "smb.enabled": true,
    "smb.filelist": [
       {
            "name": "2016-Tender-Summary.pdf",
            "type": "PDF"
       },
       {
         "name": "passwords.docx",
         "type": "DOCX"
       }
    ],
    "smb.mode": "workgroup",
    "smb.netbiosname": "FILESERVER",
    "smb.serverstring": "Windows 2003 File Server",
    "smb.sharecomment": "Office documents",
    "smb.sharename": "Documents",
    "smb.sharepath": "/changeme",
    "smb.workgroup": "OFFICE",
    "rdp.enabled": true,
    "rdp.port", 3389,
    [..] # logging configuration
}
```

2.3 MySQL Server

Inside ~/.opencanary.conf:

```
{
     "mysql.banner": "5.5.43-0ubuntu0.14.04.1",
     "mysql.enabled": true,
     "mysql.port": 3306,
     "ssh.enabled": true,
     "ssh.port": 22,
     "ssh.version": "SSH-2.0-OpenSSH_5.1p1 Debian-4",
     [..] # logging configuration
}
```

2.4 MSSQL Server

Inside ~/.opencanary.conf:

```
{
    "mssql.enabled": true,
    "mssql.port": 1433,
    "mssql.version": "2012",
    "rdp.enabled": true,
    "rdp.port", 3389,
[..] # logging configuration
}
```

CHAPTER 3

Alerting

Getting Started walks through two different ways to configure alerting: logging directly to a file, and sending alerts to the Correlator for email and SMS alerts. Other possibilities are below:

3.1 Email Alerts

To have an OpenCanary daemon directly send email alerts edit the logger section of the *~/.opencanary.conf*. The file format is JSON.

In the configurations below, set these configuration variables:

- **mailhost** - The SMTP mailhost and port.
- **fromaddr** - The from address. Usually does not have to exist.
- **toaddres** - An array of addresses that will receive the alert. Keep it short.
- **subject** - The email's subject.
- **credentials** - Optional parameter, if the SMTP server requires authentication.
- **secure** - Optional parameter if TLS support is mandatory or wanted.

More information can be found on the PyLogger page.

3.1.1 Send to a GMail address

```
[..] # Services configuration
    "logger": {
    "class" : "PyLogger",
    "kwargs" : {
        "handlers": {
            "SMTP": {
                "class": "logging.handlers.SMTPHandler",
```

(continues on next page)

(continued from previous page)

```
                "mailhost": ["smtp.gmail.com", 25],
                "fromaddr": "noreply@yourdomain.com",
                "toaddrs" : ["youraddress@gmail.com"],
                "subject" : "OpenCanary Alert"
            }
        }
    }
}
```

Depending on your ISP and their outbound spam protection mechanisms, you may need to send to TCP port 587, set up an app password and use credentials, as well as setting an empty tuple for the **secure** parameter. Your configuration would then look like:

```
[..] # Services configuration
    "logger": {
    "class" : "PyLogger",
    "kwargs" : {
        "handlers": {
            "SMTP": {
                "class": "logging.handlers.SMTPHandler",
                "mailhost": ["smtp.gmail.com", 587],
                "fromaddr": "noreply@yourdomain.com",
                "toaddrs" : ["youraddress@gmail.com"],
                "subject" : "OpenCanary Alert",
                "credentials" : ["youraddress", "abcdefghijklmnop"],
                "secure" : []
            }
        }
    }
}
```

3.1.2 Send with SMTP authentication

```
[..] # Services configuration
    "logger": {
    "class" : "PyLogger",
    "kwargs" : {
        "handlers": {
            "SMTP": {
                "class": "logging.handlers.SMTPHandler",
                "mailhost": ["authenticated.mail.server", 25],
                "fromaddr": "canary@yourdomain.com",
                "toaddrs" : ["youraddress@yourdomain.com"],
                "subject" : "OpenCanary Alert",
                "credentials" : ["myusername", "password1"],
                "secure" : []
            }
        }
    }
}
```

3.2 HPFeeds

OpenCanary can be used directly (without the Correlator) with daemons supporting the hpfeeds protocol.

To enable hpfeeds add the following to the logging section of settings.json:

```
"hpfeeds": {
    "class": "opencanary.logger.HpfeedsHandler",
    "host": "127.0.0.1",
    "port": 10000,
    "ident": "test",
    "secret":"12345",
    "channels":["test.events"]
}
```

CHAPTER 4

Indices and tables

- genindex
- modindex
- search