



OWASP DeepViolet TLS/SSL JAVA API & Tools

**Project Leader
Milton Smith**

Twitter: @deepvioletapi

Blog: <https://www.securitycurmudgeon.com/>

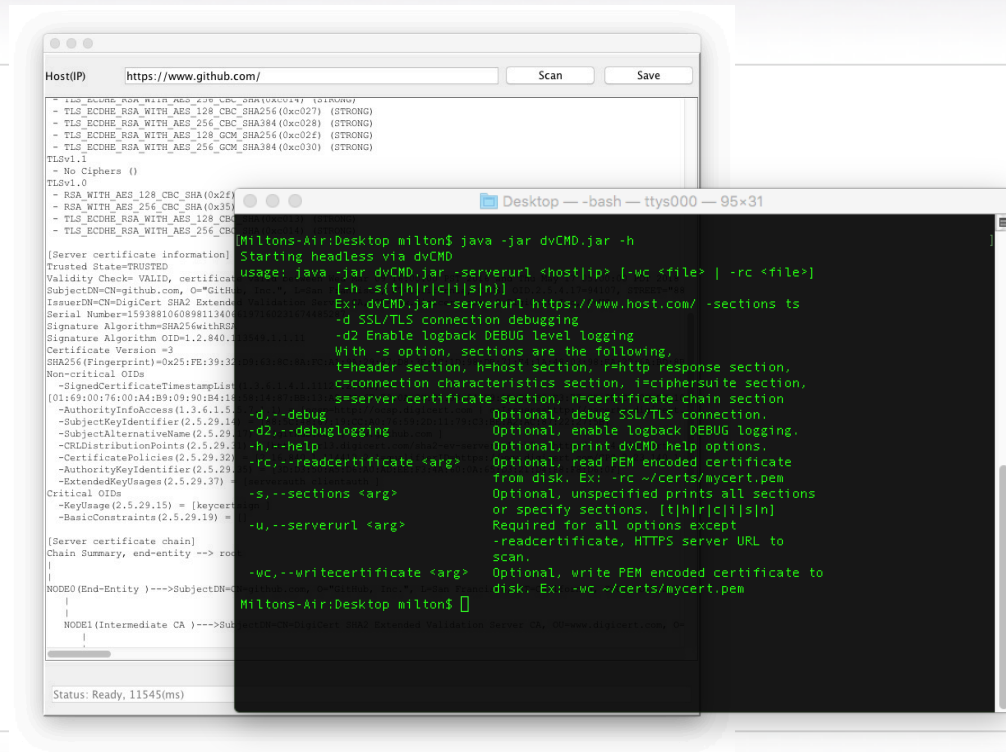
Black Hat EU 2016 London
Tools Arsenal

What is DeepViolet?

TLS/SSL scanning API

2 reference cases demonstrating API

Command line tool & desktop application



Why Build DeepViolet?

Why build DeepViolet(DV)? I did not set out to build a tool for the public.

DV was a learning tool for me. Heartbleed was in the popular press, I wanted to learn more about underlying TLS/SSL protocols. When I finished the original code I posted it to my github site.

I was approached several times to add improvements to DV.

Asked others why they liked it. Most common answer is that there are few available choices for libraries that provide TLS/SSL scanning features for applications.

Great tools exist today like OpenSSL, Qualys SSL Server Test, Mozilla Observatory, etc.

Yes, my favorites as well. No intention to compete with any tools.

What Can DeepViolet API/Tools Do?

Identify Weak Server Cipher Suites

Print X.509 Certificates & Metadata

Identify Weak Signature Algorithms

Print Trust Chains

Identify Certificates About to Expire

Print Trust Status, Trusted or Not Trusted

And more...

Getting Started with the API

```
IDSession session = DVFactory.initializeSession(url);
```

```
IDVOnEng eng = DVFactory.getIDVOnEng(session);
```

```
// Get certificates, ciphersuites, print some reports...
```

```
// Review unit tests in com.mps.deepviolet.test.api to get  
started...
```

DeepViolet Desktop Application



1) Provide a URL and Click

2) Report is generated

3) Save report to disk

Easy as that. Adapt as needed.

DeepViolet Command Tool

```
Desktop -- -bash -- ttys000 -- 95x31
[Miltons-Air:Desktop milton$ java -jar dvCMD.jar -h
Starting headless via dvCMD
usage: java -jar dvCMD.jar -serverurl <host[ip> [-wc <file> | -rc <file>]
      [-h -s{t|h|r|c|i|s|n}]
      Ex: dvCMD.jar -serverurl https://www.host.com/ -sections ts
      -d SSL/TLS connection debugging
      -d2 Enable logback DEBUG level logging
      With -s option, sections are the following,
      t=header section, h=host section, r=http response section,
      c=connection characteristics section, i=ciphersuite section,
      s=server certificate section, n=certificate chain section
      -d,--debug                Optional, debug SSL/TLS connection.
      -d2,--debuglogging        Optional, enable logback DEBUG logging.
      -h,--help                 Optional, print dvCMD help options.
      -rc,--readcertificate <arg> Optional, read PEM encoded certificate
                                from disk. Ex: -rc ~/certs/mycert.pem
      -s,--sections <arg>      Optional, unspecified prints all sections
                                or specify sections. [t|h|r|c|i|s|n]
      -u,--serverurl <arg>     Required for all options except
                                -readcertificate, HTTPS server URL to
                                scan.
      -wc,--writecertificate <arg> Optional, write PEM encoded certificate to
                                disk. Ex: -wc ~/certs/mycert.pem
Miltons-Air:Desktop milton$
```

1) Try a command line like this,
`java -jar dvCMD.jar -serverurl https://www.google.com/ -s hrcisn`

2) Report is generated

3) Redirect output to file or pipe to **grep** to search certificate metadata

Easy as that. Adapt as needed.

Additional References


OWASP Project Site:

https://www.owasp.org/index.php/OWASP_DeepViolet_TLS/SSL_Scanner

GitHub Site: <https://github.com/spoofzu/DeepViolet>

Download: <https://github.com/spoofzu/DeepViolet/releases>

Follow Online: twitter, [@deepvioletapi](#)



;o)

