



# Meeting 101

Overview/Agenda



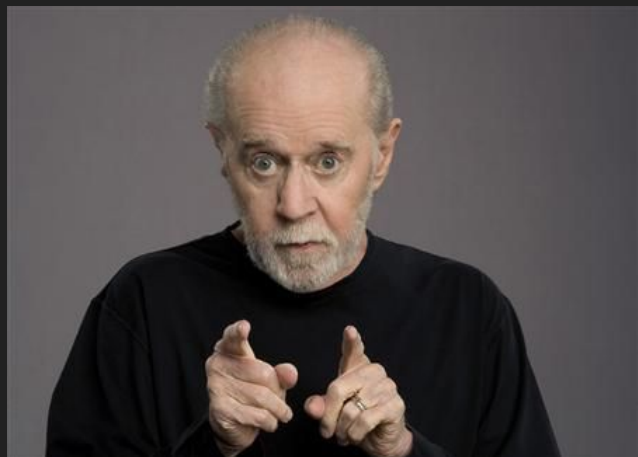
# Welcome

SecDSM.org @SecDSM

Get on #Slack - [secdsm.slack.com](https://secdsm.slack.com)

# Obligatory Disclaimer

Content delivered (or generated) at SecDSM is  
unrated. You have been warned!








Thanks to:  
PFG for Gravitate  
Casey's for Pizza

# New/Old Stuff

Website Automation  
Suggestions?  
We need speakers

# Upcoming Events

	Event	Information	Date/Time/Location
	SeckKC Roadtrip	<a href="http://seckc.org/">http://seckc.org/</a>	9/13 Meet at the Target parking lot at the West Glenn shopping center in the south west corner at 1:00pm and will leave at 1:15pm
	CornCon 2016	<a href="http://corncon.net">http://corncon.net</a>	9/17 - Davenport, IA
	BSIDES STL	<a href="http://www.securitybsides.com/w/page/99388307/BSidesSTL%20-2016">http://www.securitybsides.com/w/page/99388307/BSidesSTL%20-2016</a>	9/17
	DerbyCon	<a href="https://www.derbycon.com/">https://www.derbycon.com/</a>	9/21 - 9/25 - Hyatt Louisville Kentucky
	Secure Iowa Conference	<a href="http://secureiowaconference.com/">http://secureiowaconference.com/</a>	10/4 - Ankeny, IA

# SecDSM - status update

501c - Non Profit

Paperwork in-progress

Board/Leadership formation in-progress

# Let's get going

**Beating sslabs and securityheaders.io** - Brandon Murphy (Tool Talk) - 6:10

**Consuming SSLabs.com API** - Ben Schmitt (Tool Talk) - 6:30

**Security Onion – A quick start guide** (Aaron Tekippe) - 7:00





# SSLabs.com API

Automate Scores and Report on Changes

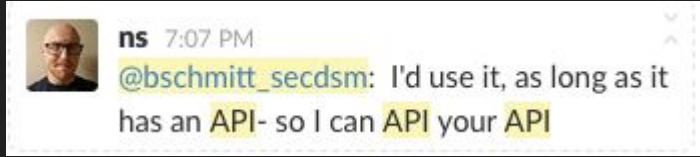
# SSLabs.com

Free service to scan a public website and provide a “grade”. Brandon spoke about getting your site clean - how do you keep it clean?

# Use the API and schedule/compare scans

What's the cost, license and capabilities?

# Rules of the API Road



The SSLabs API has a terms of service:

[https://www.ssllabs.com/downloads/Qualys\\_SSL\\_Labs\\_Terms\\_of\\_Use.pdf](https://www.ssllabs.com/downloads/Qualys_SSL_Labs_Terms_of_Use.pdf)

Key points: no cost, no screen scraping (use the API for automation), give credit, you cannot charge, you must display their API info properly, use wisely.

*An NS clause: you may not distribute, proxy, or otherwise make the API available for access or use by any person or entity other than your authorized employees, including but not limited to acting as a service bureau or developing a competing product or service offering.*

# API end points

```
API_URLS = [  
    'https://api.dev.ssllabs.com/api/v2', # dev  
    'https://api.ssllabs.com/api/v2' # stable
```

These are unauthenticated - no API key/secret needed

# There are existing projects to use the API

<https://github.com/ssllabs/ssllabs-scan/blob/stable/ssllabs-api-docs.md>

<https://www.ssllabs.com/projects/ssllabs-apis/index.html>

<https://github.com/takeshixx/python-ssllabs>

We will focus on the features I've stubbed in the python-ssllabs tool (and will contribute back via a PR if the author is willing to have them)

# APIs => JSON

Hierarchical file format:

Parsing for the grade?

```
{
  "criteriaVersion": "20091",
  "endpoints": [
    {
      "delegation": 1,
      "details": {
        "cert": {
          "altNames": [
            "*.dwolla.com",
            "dwolla.com"
          ],
          "commonNames": [
            "*.dwolla.com"
          ],
          "crlRevocationStatus": 2,
          "crlURIs": [
            "http://crl.godaddy.com/gdig2s1-219.crl"
          ],
          "issuerLabel": "Go Daddy Secure Certificate Authority - G2",
          "issuerSubject": "CN=Go Daddy Secure Certificate Authority - G2,OU=http://certs.godaddy.com/repository/",
          "issues": 0,
          "mustStaple": 0,
          "notAfter": 1525268600000,
          "notBefore": 1459805498000,
          "ocspRevocationStatus": 2,
          "ocspURIs": [
            "http://ocsp.godaddy.com/"
          ],
          "pinSha256": "v7wT1IpRU6fvi82z2Q1eoiFuiGXj0G9tYHvV9ywrjqY=",
          "revocationInfo": 3,
          "revocationStatus": 2,
          "sct": false,
          "sgc": 0,
          "sha1Hash": "1d57a906087eee1d7318c8012c5924566732fa82",
          "sigAlg": "SHA256withRSA",
          "subject": "CN=*.dwolla.com,OU=Domain Control Validated",
          "validationType": "D"
        },
        "chaCha20Preference": true,
        "chain": {
          "certs": [
            {
              "crlRevocationStatus": 2
            }
          ]
        }
      }
    }
  ]
}
```

# Key info for daily comparison

```
json_str = (json.dumps(info, indent=4, sort_keys=True))
```

```
resp = json.loads(json_str)
```

```
print "\nGrade: " + (resp['endpoints'][0]['grade'])
```

```
print "IP Address: " + (resp['endpoints'][0]['ipAddress'])
```

```
print "Host: " + (resp['host'])
```

```
print "Start Time: " + str((resp['startTime']))
```



# Algorithm

1. Run scan against target - get JSON feedback - cache in data structure, write copy to disk.
2. Get previous scan value (the grade) from sqlite db
3. Compare values - if the same, your job is done and no alert. If different, alert and diff JSON files on-disk and send relevant diff info in email
4. Commit new scan to sqlite db

# Demo

That's it - questions?

I'll post this as a Github GIST soon and work with the author if he/she is interested in the functionality (German InfoSec professional)