



 REVERSING  
 2020



SESSION

# WYATT ROERSMA

Senior Cyber Defense  
Operator, SimSpace

**MAKING YARA TESTING  
EASY - THE SIMPLE  
WEB PLUGIN TO  
SIMPLIFY YOUR DAY**



By: Wyatt Roersma





# Who Am I?

Wyatt Roersma

@WyattRoersma

<https://github.com/wroersma>

<https://wyattroersma.com>

[twitch.tv/vtriple](https://twitch.tv/vtriple)

<https://aucr.io>





# Who Am I?

Open Source Projects:

- Volatility
- Cuckoo
- mmbot
- aucr



# Outline

- Why?
- What is AUCR?
- Who would use AUCR?
- Architecture
- Yara\_Plugin
- Unum
- Demo
- Summary



# What is AUCR?

- A Micro Service Web Framework
- Python 3
- Flask
- Modern Design
  - Object Storage
  - Message Queue
- Modular
- <https://github.com/aucr/aucr>



# Why AUCR?

- OS Security software and security
- Portal hell
- Personal Use Case(CTF's)
- AWS/GCP/Azure like platform for analysts





**DISCLAIMER**



**I ♥ Free Software**

**[ilovers.org](http://ilovers.org)**



cuckoo 



# AUCR

build passing codecov 82% container ready coverage unknown code quality B

## Overview

Analyst Unknown Cyber Range is a micro services flask framework. The goal of this project to make highly scalable web services in a master framework so users have a single web interface to do all the things from. Think of what GCP/AWS is for admin users but for users(currently with a DFIR focus).

## Database support

- sqlite
- mysql
- postgres

## Developer setup

Example Setup with Temporary an example and just running with flask. If you use pycharm you can setup flask app to debug through the code. Python >= 3.6

```
pip install PyYAML
pip install -r requirements.txt
export FLASK_APP=aucr.py
export FLASK_DEBUG=1
flask run
```

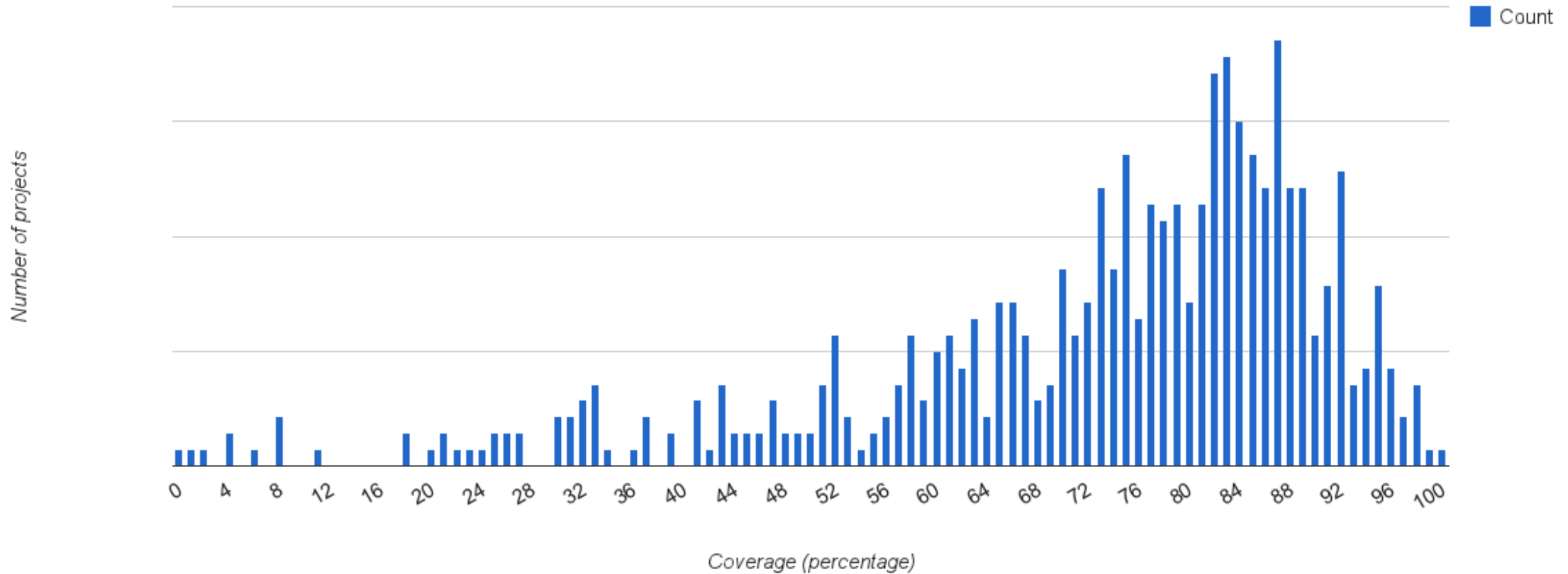
## Easy Docker use

```
sudo docker pull quay.io/wroersma/aucr
sudo docker run aucr -p 5000:5000
```

## Environment Variables

Here is an example env variables the aucr flask app will need. I use aucr local as my host for all systems but normally

Histogram of average coverage over one month



- <https://testing.googleblog.com/2014/07/measuring-coverage-at-google.html>

# Code Coverage

C++	Java	Go	JavaScript	Python
56.6%	61.2%	63.0%	76.9%	84.2%

- <https://testing.googleblog.com/2014/07/measuring-coverage-at-google.html>

As a helpful suggestion by a player I've changed the wording in #3 to ask for what a specific AV vendor uses to classify the malware to make the question easier. If anyone has suggestions please feel free to let me know! thanks for the help

→ ↺ 🏠 <https://grrcon.e-corp.biz>

I have two possible options for downloads this year. The first download is a .ov solve the challenge. The second is the stand alone forensics files you may do

- Challenge files
- Security Onion.ova

## Security Onion Credentials

```
User: eadmin
Password: grrcon2016
```



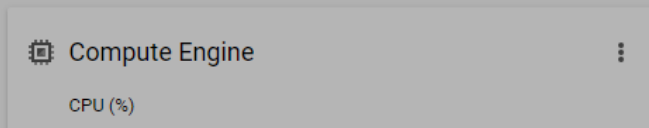
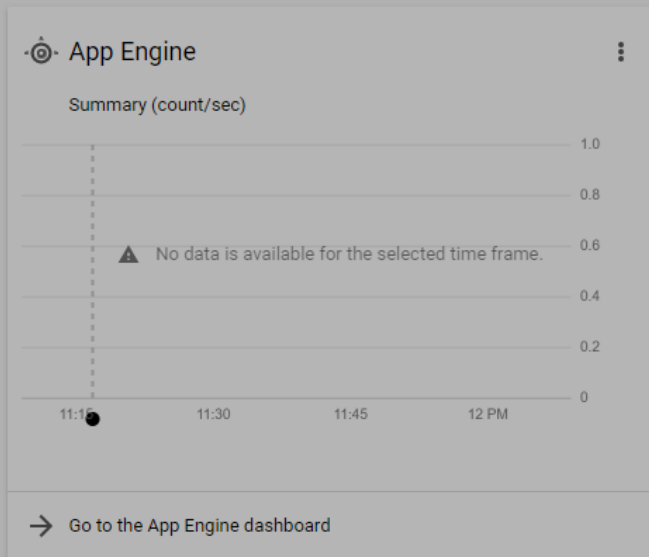
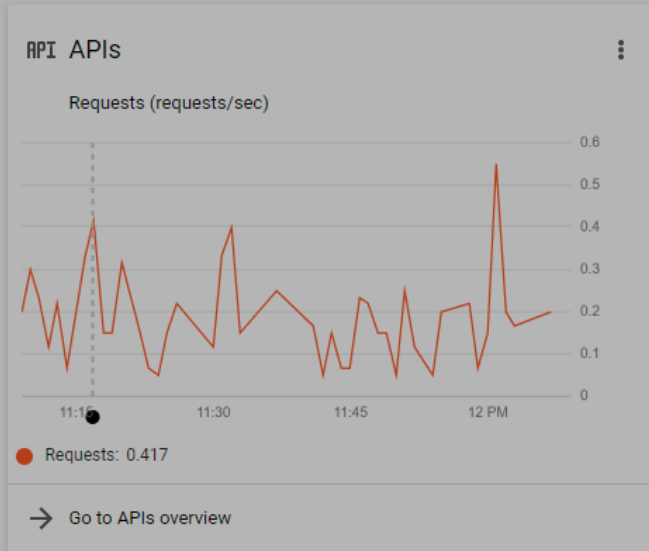
Join the [E-Corp corporate slack chat](#) and get help from the in range IH on call.

Until the end of the conference and if people want more I will provide the last set of things I have.

Currently the challenge files are located in this google drive link  
<https://drive.google.com/open?id=0Bz3L4ZnVIUY8YnZyaVV1RWF2QVvk>

The challenge will start around 9 am. Please see Wyatt Roersma with any questions or concerns. I will have USB sticks with the data on them at the conference. If you see Adam Lesperance be sure to thank him! He has been a key part behind the last 3 challenges we have created together!


























[illegible]

## Google Cloud Platform status

All services normal

→ [Go to Cloud status dashboard](#)

## Billing


                      

[View detailed charges](#)

## Error Reporting ⋮

No sign of any errors. Have you set up Error Reporting?

→ [Learn how to set up Error Reporting](#)

 News

Demonstrating our commitment to protecting user privacy and student data


3 days ago



Thinking about cloud security? Join us for a new round of Google Cloud Security Talks

3 days ago

Build your own event-sourced system using Cloud Spanner

3 days ago

 Read all news

 Documentation 

Learn about Compute Engine



## AUCR



Cases



CTF



Cuckoo



IDS



Yara



People



MMBOT



Unum

own Cyber Range

Search



# /yatt!

## TF

ipate in the challenge! **Please note: This event is for the Threat Hunter Player Type!!!**





# Feature's

- User/Group Permissions
  - API Oauth token management
  - 2fa
  - Password reset via email link
  - Navbar rendering
  - Current Auth support
    - LDAP
    - Local Database



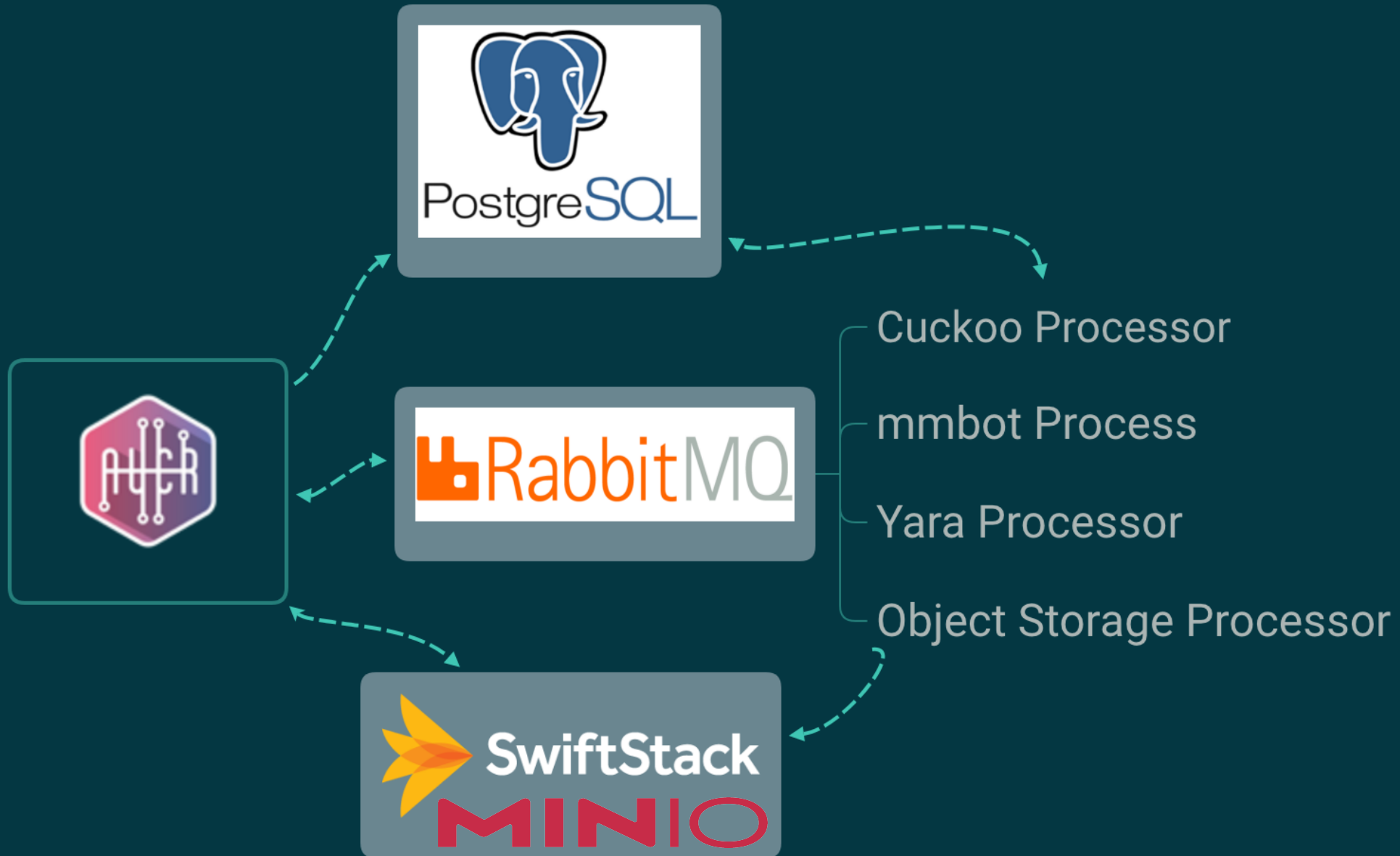
# Feature's Continued

- Database Support
  - Mysql
  - Sqlite
  - Postgres
  - ElasticSearch(Full Text Search)
- Message Pipeline
- Auto Plugin Loading
- Multi-Language Support



# Feature's Continued

- Registration Email Whitelisting
- Error Notification Reporting
- MDL CSS
- Kubernetes
- Privacy Policy Rendering
- Mobile Page Rendering





# Current Plugin Features

- Message Queue Pipeline
  - Yaml config files
- Navbar rendering based on Groups
- Auto Import
- Optional ES Indexing
- API handling
- Page Templates



# UNUM

- File Management
- <https://github.com/AUCR/unum>
- File Upload/Download
- Group Permission Based
- API

# ES Indexing



```
class UNUM(SearchableMixin, PaginatedAPIMixin, db.Model):
```

```
    """Upload File data default table for aucr."""
```

```
    __searchable__ = ['id', 'description', 'classification', 'created_by', 'md5_hash', 'file_name', 'created_time_stamp']
```

```
    __tablename__ = 'unum'
```

```
    id = db.Column(db.Integer, primary_key=True)
```

```
    description = db.Column(db.String(256), index=True)
```

```
    created_time_stamp = db.Column(db.DateTime, index=True, default=datetime.utcnow)
```

```
    modify_time_stamp = db.Column(db.DateTime, index=True, default=datetime.utcnow)
```

```
    classification = db.Column(db.Integer, db.ForeignKey('classification.id'))
```

```
    file_name = db.Column(db.String(512))
```

```
    created_by = db.Column(db.Integer, db.ForeignKey('user.id'))
```

```
    group_access = db.Column(db.Integer, db.ForeignKey('groups.id'))
```

```
    md5_hash = db.Column(db.String(128), db.ForeignKey('uploaded_file_table.md5_hash'))
```

```
    def __repr__(self):
```

```
        return '<unum {}>'.format(self.md5_hash)
```





```
main:
  user: [
    link : [
      page: 'auth.users',
      icon: 'face',
      title: 'People'
    ]
  ]
  admin: [
    link: [
      page: 'auth.groups',
      icon: 'group',
      title: 'Groups'
    ]
  ]
]
```



Unum



	Upload File ID	Description	MD5	File_Name	Classification
<input type="checkbox"/>	<a href="#">1</a>	APIUpload Known Bad Document	b60e085150d53fce271cd481435c6e1e	b2921ed8848d729b935571fbca...165eca9a	KnownBad
<input type="checkbox"/>	<a href="#">2</a>	APIUpload Known Bad Document	070cb070164957f802d42faa0e7cc498	97d154433f6d285a71642a2034...6a02902b	KnownBad
<input type="checkbox"/>	<a href="#">3</a>	APIUpload Known Bad Document	e11da1aeaf76b1f92aa15b9ddc7f9eae	5ad8fa0c2f8023245961f649c9...429cf7fa	KnownBad
<input type="checkbox"/>	<a href="#">4</a>	APIUpload Known Bad Document	a1d1832046b801fb5ec5125b7bd0c81a	c4b7100160a3679bd41ed4b880...363f74dc	KnownBad
<input type="checkbox"/>	<a href="#">5</a>	APIUpload Known Bad Document	dd72749fe33df4f024a196927bcb2245	4e64d9d3ed70321e6ecec9124c...9e4752a2	KnownBad
<input type="checkbox"/>	<a href="#">6</a>	APIUpload Known Bad Document	9b956c26d7953d1d0f32a14bf75a5c89	bd9cf83cb9b9e6c63b8a6cc885...ede6c990	KnownBad
<input type="checkbox"/>	<a href="#">7</a>	APIUpload Known Bad Document	189de6b04e1ec7170487bd2346536d00	7bd4d8d48a4e64ee8bdd8814c8...1e193b9d	KnownBad
<input type="checkbox"/>	<a href="#">8</a>	APIUpload Known Bad Document	256f96d2b31a781888b43f5f68b10b83	136379754edd05c20d5162aed7...c9018aa6	KnownBad
<input type="checkbox"/>	<a href="#">9</a>	APIUpload Known Bad Document	15b03c940e77f7923ce69055027d7937	ab7f36ea43ed812f4b8b559b73...d0080c0b	KnownBad
<input type="checkbox"/>	<a href="#">10</a>	APIUpload Known Bad Document	cf5f4962475a95e85c05bc47a2b6c2d3	089f8d15897fd782b5da46e796...e5bc0fe4	KnownBad

&lt; \_ &gt;



FILE

URL

HASH

## Upload New File

Description



Choose File

No file chosen

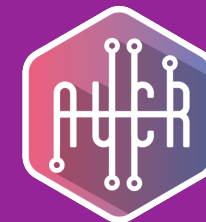
Classification



Group Access



CREATE



## Edit Upload File

Upload ID: 10 MD5: [cf5f4962475a95e85c05bc47a2b6c2d3](#)

Description

APIUpload Known Bad Document

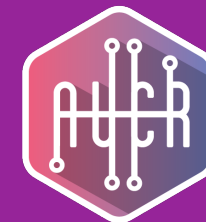
Classification

KnownBad

Group Access

user

SAVE



## UNUM File Search Results

File ID : [1078](#)

MD5: [6b8e854588081b58297bf26ea8b8aa44](#)

Description : APIUpload Known Good Document

File Name : b6ded1a559c333a0f7f21e9badd6bdf32a326a1990f8311e2b6723984bb6d02c

File ID : [1079](#)

MD5: [a65bde91be98a021eda8f010f124ad1e](#)

Description : APIUpload Known Good Document

File Name : 0fe9eaa0b48375ff52a14772d6c81d293228a833eedbdfd5c8e0a614d0757ac1.json

File ID : [1080](#)

MD5: [e82e51edae553a467ea4f481d8fe6490](#)

Description : APIUpload Known Good Document

File Name : 789c35a93e5d8d72c79d374a8be2a8f5b0b5554afddf30e3cc0a5bbd1f4e1833.json

File ID : [1081](#)

MD5: [c601f627ba389cb36adf4654892d6d39](#)

Description : APIUpload Known Good Document

File Name : 590516ea132357b91fbd9209a8ad10a9b48d905d58a5184956571ae1afba2f15.json

File ID : [1082](#)

MD5: [179b306a5635749d0c081d79b56a60e1](#)

Description : APIUpload Known Good Document

File Name : e1c0b7c4b081ee6b446b0f1ffd63c04dc3bfdb9ca9946d4fdb63c3fba64761aa.json

File ID : [1083](#)

MD5: [b362ec5698b061433cb012c1093fecc4](#)

Description : APIUpload Known Good Document

File Name : c97ede4a00b681c860561c118c06a5bf0e7e2315759c20dd5ae2c895f50c0e7a.json

File ID : [1084](#)

MD5: [1cd3ef7a0a45f37a727f032c1b083a3f](#)

Description : APIUpload Known Good Document

File Name : a9667c5e9ba831d9a9ea20dee84078f99fc3a64257867f271a1242ac5ef45b75.json

File ID : [1085](#)

MD5: [7af7ca70e8d4661e0ca9dd4dbec42b69](#)

Description : APIUpload Known Good Document

File Name : f66e1bf173e2af781f85e0486b54219c7ee307a21f3727f85ce4f14c09694dea

File ID : [1086](#)

MD5: [04dc393a1e1303ffbe4edb4320261b75](#)

Description : APIUpload Known Good Document

File Name : 910eae0f1bf9a3aa2dedccc3e597b0709d8dbe7c26f5f777c0c053708a612bb3.json

File ID : [1087](#)

MD5: [7b30d69ba69099925eea45c660358b91](#)

Description : APIUpload Known Good Document

File Name : b83216e3ed3b865470a3692b048d96b9336332fe8ef521fc1cd59e595ad3ce29



# Unum API example

```
import ujson
import requests
API_URL = 'http://0.0.0.0:5000'
API_KEY = 'CRyPP50DBJUIMCxErcNmGAv8i3cZkQU+799KmojUdtBpKVpml3DBZRY4ql=='
file_name = "some_mal_file"
headers = {'Authorization': 'Bearer ' + API_KEY}
with open(file_name, 'rb') as file_object:
    file_data = file_object.read()
payload = {'filename': str(file_name), 'group_access': 2, 'description': "bad", 'classification': 1}
response = requests.post('{} /api/unum_file_upload/'.format(
    API_URL), headers=headers, data=file_data, params=payload)
test = ujson.loads(response.text)
print(ujson.dumps(test, indent=4, sort_keys=True))
```



# JSON Results

```
{  
  "file_id": 22,  
  "md5": "12e8f6658618e9169958802fa261ef42"  
}
```

# Yara



“YARA is a tool aimed at (but not limited to) helping malware researchers to identify and classify malware samples.

With YARA you can create descriptions of malware families (or whatever you want to describe) based on textual or binary patterns.

Each description, a.k.a rule, consists of a set of strings and a boolean expression which determine its logic.“

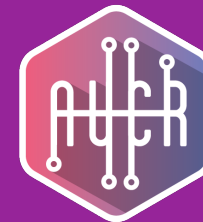
<https://virustotal.github.io/yara/>





# Yara Plugin

- Yara Rule Management
- [https://github.com/AUCR/yara\\_plugin](https://github.com/AUCR/yara_plugin)
- API push/pull rules/results
- Cuckoo report usage
- <https://virustotal.github.io/yara/>



The Yara Rule tester\_rule has been updated and the rule is running.

YARA



	Rule ID	Rule List Name	Author	Total Hits	Last Modified
<input type="checkbox"/>	<a href="#">1</a>	tester_rule	admin	23	2020-06-29 13:49:08



Analyst Unknown Cyber Range

yara.yara\_search

EDIT RULE

CURRENT MATCHES

List Name

tester\_rule

Yara Rules

rule tester\_rule{  
 meta:  
 description = "test"  
 strings:  
 \$str1 = "GetProcAddress"  
 \$str2 = "LoadLibraryA"  
 condition:  
 all of them  
}

SAVE



EDIT RULE   CURRENT MATCHES

File ID	MD5 Hash	Classification
<a href="#">1</a>	5788757e2bb6ea12b43d05fd2ebc669b	KnownBad





admin said



admin said





admin said 4 days ago

Message

Yara Rule has been created for the file MD5: 97d179cacf7cf5d61f924597266e8920

You can see the rule set using ID:596.

```
rule tdo_jpg2_image_generated_1 {
```

```
  meta:
```

```
    filetype = "doc"
```

```
    tlp = "amber"
```

```
    author = "Halo_Yara_Generated"
```

```
    version = "1.0"
```

```
    license = "internal use only"
```

```
    md5 = "97d179cacf7cf5d61f924597266e8920"
```

```
    weight = 100
```

```
    family = "Office Document embedded pic"
```

```
    filename = "/opt/aucr/upload/97d179cacf7cf5d61f924597266e8920"
```

```
    scope = "[detection, hunting, prevention]"
```

```
    intel = "[]"
```

```
  strings:
```

```
    $jpg2_img_value_1 = {380300d2b903e3645e907ce3e4291ba2f82cbffffd8ffe000104a46494600010101006000600000ffe10144457869660  
00049492a00e20000005b53434d5d61637477696e2c302c302c302c303b687474703a2f2f706f7274616c2e6875642e676f762f687564}
```

```
  condition:
```

Reply

Halo -

MMBOT Report

id	11
processed_time_stamp	2018-11-15T00:57:58Z
vba_lang_features	sender_vbcomponents_helpcontext_vbe_space
vba_avg_param_per_func	0.09
vba_cnt_comment_loc_ratio	0.01
vba_cnt_comments	6
vba_cnt_func_loc_ratio	0.03
vba_cnt_functions	11
vba_cnt_loc	424
vba_entropy_chars	4.97
vba_entropy_func_names	3.56
vba_entropy_words	3.81
vba_mean_loc_per_func	38.55
function_names	TWJEtz, LazXm, WzhWI, BLZuAAic, dfbNNKQYR, ilmGRKb, DwcFUu, utmwDIKwi, rdGauAEVOw, FnIjwiYNOj, AutoOpen
prediction	malicious
confidence	0.86
md5_hash	12e8f6658618e9169958802fa261ef42



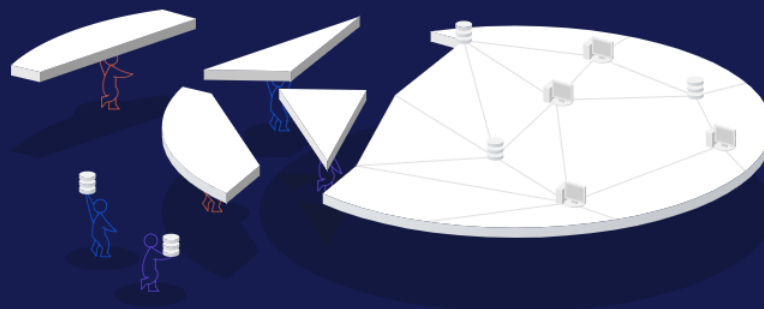
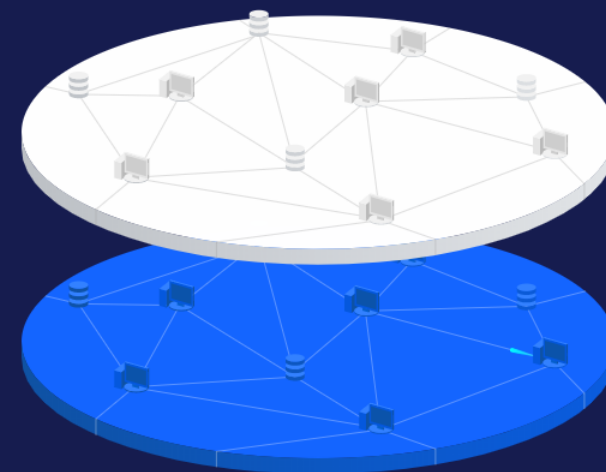
# Moving Forward

- VCS Support (git)
- UI Improvements
- Yara Syntax Rule formatting
- Dashboard for string matches
- Halo Plugin Coming Soon!



# Don't just plan for the future. Simulate it.

SimSpace is the visionary yet practical platform for measuring how your security system responds under actual, sustained attack. We are moving global business from total denial to full preparedness.



## From the Department of Defense to defending your network

The principles and methods behind SimSpace were born in the crucible of the Department of Defense. They call it "train like you fight." We have advanced and refined those simulation skills — leveraging Red, Blue and White Teams — for the everyday realities of enterprise companies. We predict to protect.

## ELSA 1.7.3: Basic File Detection with Yara Rules v2

### Outline

#### Chain

Basic File Detection with YARA Rules

#### Chain

Reading YARA Rules

#### Chain

Example #1: Very Specific Malware

#### Chain

Example #2: Very General Malware

#### Chain

Knowledge Check: Reading YARA Rules

Knowledge Check: YARA vs. File Hashes

Downloading and Using YARA

#### Chain

Further Reading

[+ Create New Chain](#)

QUESTION

INFORMATION

### Basic File Detection with YARA Rules



YARA rules are characterizations of digital fingerprints used by security researchers to classify malware families and share threat intelligence. Originally created at VirusTotal, engines to process YARA rules now exist in a large number of antivirus products. This module will introduce the basic concepts on how to read and interpret YARA signatures for malware and file samples.

#### Learning Objectives:

- Knowing what YARA is.
- Common or useful YARA rule use cases.
- Understanding of YARA rule format structure.

#### Learning Outcomes:

- How to read a YARA rule.
- How to run a custom YARA rule against a set of files.

### VM Access

[+ Add VM](#)

### Attachments

[+ Add File](#)





# Conclusion

- Yara\_plugin for AUOCR useful?
- Beta @ <https://aucr.io>
- Dev - <https://www.youtube.com/watch?v=5Qr3jy7N3Ig>
- Build local instance from [https://github.com/aucr/yara\\_plugin](https://github.com/aucr/yara_plugin)
- Interested in training for your company? Lets Talk!



Speaker



 REVERSING  
2020

# Q/A WYATT ROERSMA

Senior Cyber Defense  
Operator, SimSpace

MAKING YARA TESTING  
EASY - THE SIMPLE  
WEB PLUGIN TO  
SIMPLIFY YOUR DAY

NEXT SPEAKER



**Moshe  
Caplan**  
Senior Malware  
Analyst,  
JPMorgan Chase