



MIS|TI™ PRESENTS

InfoSecWorld
Conference & Expo 2018

BEYOND AUTOMATED AWS SECURITY ASSESSMENT: ADVANCED USAGE OF SCOUT2

Loic Simon

Director of Product Security @ Tanium
@l01cd3v

INTRODUCTION: SCOUT2 – WHAT?

- Security Tool for AWS
 - Static analysis of the configuration of AWS services and resources
- Open Source
 - Available on GitHub at <https://github.com/l01cd3v/AWSScout2>
- Python
 - Available via PIP: pip install awsscout2
- Command line tool
- Standalone HTML report
- Does not require AWS pentesting form to be filled out
- **Supports custom rules and rulesets**

INTRODUCTION: SCOUT2 – WHY?

BIZ & IT —

AWS console breach leads to demise of service with “proven” backup plan

Code Spaces closes shop after attackers destroy Amazon-hosted customer data.

Hackers Use Amazon's AWS Computing Resources to Mine Bitcoin



15 FEB 2018 NEWS
FedEx S3 Bucket Exposes Private Details on Thousands Worldwide

Security

Dev put AWS keys on Github. Then BAD THINGS happened

Fertile fields for Bitcoin yields - with a nasty financial sting



InfoSecWorld
Conference & Expo 2018

INTRODUCTION: SCOUT2 – HOW?

1. Preparation

- Installation & setup of required tools
- Configuration of AWS credentials in test environment

2. Getting familiar with the tools

- Run Scout2 in test environment
- Review of the first report
- Discovery and usage of custom summaries

3. Adding value with custom input

- Creation and management of exceptions
- Creation and management of known CIDRs

4. Tweaking the analysis with a custom ruleset

- Update of parameterized rules
- Enable a non-default rule

5. Adding functionality with new rules

- Creation a new parameterized rule with new input
- Creation of a new rule
- Creation of a new parameterized rule from scratch

PREPARATION



INSTALLATION & SETUP

Option1: Install AWSScout2

```
(sudo) pip install awsscout2  
(sudo) pip install awscli  
git clone https://github.com/nccgroup/aws-recipes
```

Option2: Use the VMWare or VirtualBox image available on the USB stick

username: aws
password: scout2

If using a virtual machines, ensure that at least 2048MB of RAM has been allocated to the VM; 4096MB is preferable.

CONFIGURATION OF AWS CREDENTIALS

1. Browse to [**https://training.awsscout2.net**](https://training.awsscout2.net)
2. Enter your email address and submit the sign up form
3. Check your email for your credentials
4. Browse to [**https://console.aws.amazon.com**](https://console.aws.amazon.com)
5. In the account alias or account ID field, enter “l01cd3v-scout2”
6. Enter the username and passwords that were generated and provided via email
7. Create a new password when prompted
8. Confirm you were able to sign into the AWS web console

CONFIGURATION OF AWS CREDENTIALS (2)

1. In the AWS web console, click on “Services” then on “IAM”

<https://console.aws.amazon.com/iam/home>

[Ignore the error messages]

1. Click on “Users”, then click on your username
2. Under the “Security credentials” tab, click on “create an AWS access key”
3. In your terminal, enter the following command

`aws configure --profile scout2lab`

5. When prompted for the AWS access key ID, copy and paste the ID displayed in the web console
6. When prompted for the secret key, copy and paste the secret value displayed in the web console
7. Leave the default region name and output format to “None”
8. Confirm the AWS credentials have been successfully configured with the following command

`aws --profile scout2lab iam list-users`

FIRST RUN



RUN SCOUT2 AND EXPLORE RESULTS

In the terminal, run the following command

```
Scout2 --profile scout2lab
```

Confirm no error messages are displayed, and the HTML report opens in a browser

Explore the results by clicking on links in the various dashboard

Explore the custom summaries available in Scout2, such as the IAM permissions summary

RUN SCOUT2 AND EXPLORE RESULTS: DASHBOARDS

Scout2 Report

file:///private/tmp/scout2-report/report-backup.html

Scout2 Analytics Compute Database Management Messaging Network Security Storage Regions Filters Help

Account ID: 179374595322

Dashboard

Summary:

Service	# of Resources	# of Rules	# of Findings	# of Checks
Lambda	3	0	0	0
Cloudformation	26	1	2	26
CloudTrail	47	5	5	111
CloudWatch	2	1	1	2
Directconnect	0	0	0	0
EC2	65	22	125	1594
EFS	0	0	0	0
Elasticache	0	0	0	0
Elb	0	1	0	0
Elbv2	5	3	0	0
Emr	0	0	0	0
IAM	68	28	40	279
RDS	31	7	0	0

RUN SCOUT2 AND EXPLORE RESULTS: DASHBOARDS

The screenshot shows the Scout2 Report interface with the title "Scout2 Report" at the top. The URL in the address bar is "file:///private/tmp/scout2-report/report-backup.html#services.iam.findings". The main content is the "IAM Dashboard" which displays various findings categorized into colored boxes:

- Cross-account AssumeRole policy lacks external ID and MFA** (Red):
 - Roles checked: 20
 - Roles flagged: 1
- AssumeRole policy allows all principals** (Red):
 - Roles checked: 20
 - Roles flagged: 1
- Unused role for EC2** (Grey):
 - Roles checked: 0
 - Roles flagged: 0
- Group with inline policies** (Orange):
 - groups checked: 12
 - groups flagged: 4
- Inline group policy allows NotActions** (Red):
 - Policies checked: 4
 - Policies flagged: 1
- Inline group policy allows iam:PassRole *** (Red):
 - Policies checked: 4
 - Policies flagged: 1
- Inline user policy allows sts:AssumeRole *** (Red):
 - Policies checked: 4
 - Policies flagged: 1
- Inline role policy allows sts:AssumeRole *** (Red):
 - Policies checked: 6
 - Policies flagged: 1
- Inline user policy allows NotActions** (Red):
 - Policies checked: 6
 - Policies flagged: 1
- Inline role policy allows iam:PassRole *** (Red):
 - Policies checked: 6
 - Policies flagged: 1
- Inline role policy allows sts:AssumeRole *** (Red):
 - Policies checked: 6
 - Policies flagged: 1
- Inline user policy allows iam:PassRole *** (Red):
 - Policies checked: 4
 - Policies flagged: 1
- Managed policy allows NotActions** (Green):
 - Policies checked: 28
 - Policies flagged: 0
- Managed policy allows iam:PassRole *** (Green):
 - Policies checked: 28
 - Policies flagged: 0
- Minimum password length too short** (Red):
 - Password policy checked: 1
 - Password policy flagged: 1
- Password expiration disabled** (Red):
 - Password policy checked: 1
 - Password policy flagged: 1
- Role with inline policies** (Orange):
- Password reuse enabled** (Red):
- Lack of MFA (root account)** (Green):

RUN SCOUT2 AND EXPLORE RESULTS: DASHBOARDS

A screenshot of a web browser window titled "Scout2 Report". The address bar shows the URL "file:///private/tmp/scout2-report/report-backup.html#services.route53.findings". The page content is the "Route53 Dashboard" from the Scout2 tool. The dashboard features three cards:

- Domain not set to autorenew** (Green background):
 - Domains checked: 1
 - Domains flagged: 0
- Domain transfer not locked** (Red background):

Domain transfer locking helps mitigating risk of unauthorized domain name transfer in the event of AWS account compromised.
- Domain transfer lock not supported by TLD** (Green background):
 - Domains checked: 1
 - Domains flagged: 0

The browser interface includes standard navigation buttons (back, forward, search) and a top menu bar with links like "Scout2", "Analytics", "Compute", etc.

RUN SCOUT2 AND EXPLORE RESULTS: SUMMARIES

Scout2 Report

file:///private/tmp/scout2-report/report-backup.html#services.iam.permissions

Scout2 Analytics ▾ Compute ▾ Database ▾ Management ▾ Messaging ▾ Network ▾ Security ▾ Storage ▾ Regions ▾ Filters ▾ Help ▾ Loic

IAM Permissions

Action Permission	Details
*	<p>Groups:</p> <ul style="list-style-type: none">Allow:<ul style="list-style-type: none">Administrators<ul style="list-style-type: none">Resources<ul style="list-style-type: none">* granted in AdministratorAccessDeny:<ul style="list-style-type: none">Scout2Group004<ul style="list-style-type: none">Resources<ul style="list-style-type: none">* granted in DenyAllBlockedUsers<ul style="list-style-type: none">Resources<ul style="list-style-type: none">* granted in DenyAllScout2Group002<ul style="list-style-type: none">Resources<ul style="list-style-type: none">* granted in DenyAllScout2Group003<ul style="list-style-type: none">Resources<ul style="list-style-type: none">* granted in DenyAllAllServices<ul style="list-style-type: none">Resources<ul style="list-style-type: none">* granted in EnforceTLS
	<p>Condition: [object Object]</p> <p>Roles:</p> <ul style="list-style-type: none">Allow:<ul style="list-style-type: none">Admin<ul style="list-style-type: none">Resources<ul style="list-style-type: none">* granted in AdministratorAccessAWSCloudFormationStackSetExecutionRole<ul style="list-style-type: none">Resources<ul style="list-style-type: none">* granted in AdministratorAccessDeny:<ul style="list-style-type: none">Scout2Role001<ul style="list-style-type: none">Resources<ul style="list-style-type: none">* granted in DenyAllScout2Role004<ul style="list-style-type: none">Resources<ul style="list-style-type: none">* granted in DenyAll

javascript:showObject('services.iam.groups.AGPAI3QSYONRJA7JCWW4I!')

CUSTOM INPUT

MANAGE EXCEPTIONS



MANAGE EXCEPTIONS: EXPECTATION

The screenshot shows the Scout2 Report interface for the IAM service. The top navigation bar includes links for Scout2, Analytics, Compute, Database, Management, Messaging, Network, Security (which is currently selected), Storage, Regions, Filters, and Help. The title bar indicates the report is for 'Scout2 Report' and the specific URL is 'file:///Users/loic.simon/aws/scout2-report/report.html#services.iam.findings'. The main content area is titled 'IAM Dashboard' and displays a grid of 15 findings categorized by color: red, green, and orange.

Category	Description	Details
Red (Cross-account AssumeRole policy lacks external ID and MFA)	Cross-account AssumeRole policy lacks external ID and MFA	• Roles checked: 4 • Roles flagged: 1
Green (AssumeRole policy allows all principals)	AssumeRole policy allows all principals	• Roles checked: 4 • Roles flagged: 0
Green (Unused role for EC2)	Unused role for EC2	• Roles checked: 0 • Roles flagged: 0
Green (Group with inline policies)	Group with inline policies	• groups checked: 0 • groups flagged: 0
Green (Inline group policy allows NotActions)	Inline group policy allows NotActions	• Policies checked: 0 • Policies flagged: 0
Green (Inline role policy allows iam:PassRole *)	Inline role policy allows iam:PassRole *	• Policies checked: 0 • Policies flagged: 0
Green (Inline user policy allows iam:PassRole *)	Inline user policy allows iam:PassRole *	• Policies checked: 1 • Policies flagged: 0
Green (Managed policy allows iam:PassRole *)	Managed policy allows iam:PassRole *	• Policies checked: 13 • Policies flagged: 0
Red (Minimum password length too short)	Minimum password length too short	• Password policy checked: 1 • Password policy flagged: 1
Red (Password reuse enabled)	Password reuse enabled	• Password policy checked: 1 • Password policy flagged: 1
Orange (Role with inline policies)	Role with inline policies	• Policies checked: 13 • Policies flagged: 0
Red (Lack of MFA (root account))	Lack of MFA (root account)	• Policies checked: 0 • Policies flagged: 0
Green (Inline group policy allows sts:AssumeRole *)	Inline group policy allows sts:AssumeRole *	• Policies checked: 0 • Policies flagged: 0
Green (Inline role policy allows sts:AssumeRole *)	Inline role policy allows sts:AssumeRole *	• Policies checked: 1 • Policies flagged: 0
Green (Inline user policy allows sts:AssumeRole *)	Inline user policy allows sts:AssumeRole *	• Policies checked: 0 • Policies flagged: 0
Green (Managed policy allows sts:AssumeRole *)	Managed policy allows sts:AssumeRole *	• Policies checked: 13 • Policies flagged: 0

MANAGE EXCEPTIONS: EXPECTATION

The screenshot shows the Scout2 Report interface for the IAM Dashboard. The top navigation bar includes links for Scout2, Analytics, Compute, Database, Management, Messaging, Network, Security, Storage, Regions, Filters, and Help. The main content area is titled "IAM Dashboard" and displays a grid of 18 findings categorized by color:

- Red (3 findings):**
 - Cross-account AssumeRole policy lacks external ID and MFA
 - Group with inline policies
 - Inline group policy allows sts:AssumeRole *
- Green (4 findings):**
 - Inline role policy allows sts:AssumeRole *
 - Inline user policy allows sts:AssumeRole *
 - Managed policy allows sts:AssumeRole *
 - Password reuse enabled
- Yellow (1 finding):**
 - Role with inline policies
- Grey (11 findings):**
 - AssumeRole policy allows all principals
 - Unused role for EC2
 - Inline group policy allows NotActions
 - Inline role policy allows NotActions
 - Inline user policy allows NotActions
 - Managed policy allows NotActions
 - Minimum password length too short
 - Password expiration disabled
 - Lack of MFA (root account)

Each finding box contains a brief description and a list of metrics: Roles/Groups checked and Roles/Groups flagged.

MANAGE EXCEPTIONS: GUIDE

1. In the Scout2 HTML report, browse to a finding
2. Click on the highlighted value
3. Click on “OK” at the “Mark this item as an exception” prompt
4. Click on “Help” then “Export exceptions”
5. Re-Run Scout2 and provide the list of exceptions as an argument

Scout2 --profile <> --local --exceptions ~/Downloads/exceptions.json

6. Browse to the service dashboard and confirm that the exception count has decreased

MANAGE EXCEPTIONS: UNDER THE HOOD

```
$ cat ~/Downloads/exceptions-report.json

exceptions =
{
  "iam": {
    "iam-password-policy-minimum-length": [
      "iam.password_policy.MinimumPasswordLength"
    ]
  }
}
```

MANAGE EXCEPTIONS: LAB

Instructions provided during lab

CUSTOM INPUT

USE A LIST OF
KNOWN
IP ADDRESSES



USE A LIST OF KNOWN IP ADDRESSES: EXPECTATION

The screenshot shows the Scout2 Report interface with the following details:

Scout2 Report - File:///private/tmp/scout2-report/report-backup.html#services.ec2.filters.ec2-security-group-with-public-cidr-grant.items

Scout2 - Analytics, Compute, Database, Management, Messaging, Network, Security, Storage, Regions, Filters, Help

us-west-1 - vpc-e8eff28d

us-west-2 - vpc-620d1706, Scout2

TestSecurityGroup018

- Information**
 - Description: Use of port ranges
 - Region: ap-southeast-1
 - VPC: (EC2-Classic)
 - ID: sg-4c891719
- Egress Rules** (0)
- Ingress Rules** (1)
 - TCP
 - Ports:
 - 1-1000
 - IP addresses:
 - 1.2.3.4/32
- Usage**

TestSecurityGroup008

- Information**
 - Description: RDP port open to all
 - Region: ap-southeast-1
 - VPC: (EC2-Classic)
 - ID: sg-4e89171b
- Egress Rules** (0)
- Ingress Rules** (1)
 - TCP
 - Ports:
 - 3389
 - IP addresses:
 - 0.0.0.0/0

USE A LIST OF KNOWN IP ADDRESSES: EXPECTATION

The screenshot shows the Scout2 Report interface with the URL `file:///private/tmp/scout2-report/report-backup.html#services.ec2.filters.ec2-security-group-with-public-cidr-grant.items`. The left sidebar lists regions: us-west-1 (selected) and us-west-2. The main content area displays two security groups:

- TestSecurityGroup018**
 - Information**: Description: Use of port ranges, Region: ap-southeast-1, VPC: (EC2-Classic), ID: sg-4c891719
 - Egress Rules**: 0
 - Ingress Rules**: 1
 - TCP
 - Ports:
 - 1-1000
 - IP addresses:
 - 1.2.3.4/32 (San Francisco Office)
 - Usage**: 0
- TestSecurityGroup008**
 - Information**: Description: RDP port open to all, Region: ap-southeast-1, VPC: (EC2-Classic), ID: sg-4e89171b
 - Egress Rules**: 0
 - Ingress Rules**: 1
 - TCP
 - Ports:
 - 3389
 - IP addresses:
 - 0.0.0.0/0 (All)
 - Usage**: 0

USE A LIST OF KNOWN IP ADDRESSES: GUIDE

1. Create a list of known IP addresses with the `aws_recipes_create_ip_ranges.py` tool

1. Interactive mode

```
awsrecipes_create_ip_ranges.py --interactive --debug
```

2. From a CSV file

TODO

3. From any AWS account

```
awsrecipes_create_ip_ranges.py --profile <>
```

2. Run Scout2 with the list of known IP addresses

```
Scout2 --profile <> --local --ip-ranges ip-ranges-file.json
```

3. Observe subnet and host names displayed in the security group detailed grants view

USE A LIST OF KNOWN IP ADDRESSES: UNDER THE HOOD

```
{  
  "createDate": "2018-03-10-10-36-46",  
  "prefixes": [  
    {  
      "ip_prefix": "5.6.7.8",  
      "name": "New York Office"  
    },  
    {  
      "ip_prefix": "1.2.3.4/32",  
      "name": "San Francisco Office"  
    }  
  ]  
}
```

USE A LIST OF KNOWN IP ADDRESSES: LAB

Instructions provided during lab

CUSTOM RULESETS

UPDATE OF A
PARAMETERIZED
RULE



UPDATE OF A PARAMETERIZED RULE: EXPECTATIONS

The screenshot shows the Scout2 Report interface with the title "Scout2 Report" at the top. The URL in the address bar is "file:///Users/loic.simon/aws/scout2/scout2-report/report-backup.html#services.iam.findings". The main header is "IAM Dashboard". Below it, there is a grid of 18 cards, each representing a different finding or rule check. The cards are arranged in three rows of six. The colors of the cards indicate the severity or status of the finding:

- Red Cards (Top Row):**
 - Cross-account AssumeRole policy lacks external ID and MFA
 - AssumeRole policy allows all principals
 - Unused role for EC2
- Orange Cards (Second Row):**
 - Group with inline policies
 - Inline group policy allows NotActions
 - Inline group policy allows iam:PassRole *
- Red Cards (Third Row):**
 - Inline group policy allows sts:AssumeRole *
 - Inline role policy allows NotActions
 - Inline role policy allows iam:PassRole *
- Red Cards (Fourth Row):**
 - Inline role policy allows sts:AssumeRole *
 - Inline user policy allows NotActions
 - Inline user policy allows iam:PassRole *
- Red Cards (Fifth Row):**
 - Inline user policy allows sts:AssumeRole *
 - Managed policy allows NotActions
 - Managed policy allows iam:PassRole *
- Green Cards (Sixth Row):**
 - Managed policy allows sts:AssumeRole *
 - Minimum password length too short
 - Password expiration disabled
- Red Card (Bottom Row):**
 - Password reuse enabled
- Orange Card (Bottom Row):**
 - Role with inline policies
- Green Card (Bottom Row):**
 - Lack of MFA (root account)

UPDATE OF A PARAMETERIZED RULE: EXPECTATIONS

The screenshot shows the Scout2 Report interface with the title "Scout2 Report" at the top. The URL in the address bar is "file:///Users/loic.simon/aws/scout2/scout2-report/report-backup.html#services.iam.findings". The main content area is titled "IAM Dashboard" and displays a grid of 18 findings categorized by color:

Color	Category	Description	Details
Red	Cross-account AssumeRole policy lacks external ID and MFA	AssumeRole policy allows all principals	• Roles checked: 20 • Roles flagged: 1
Red	Unused role for EC2	Unused role for EC2	• Roles checked: 0 • Roles flagged: 0
Orange	Group with inline policies	Inline group policy allows NotActions	• Policies checked: 4 • Policies flagged: 1
Red	Inline group policy allows sts:AssumeRole *	Inline role policy allows NotActions	• Policies checked: 6 • Policies flagged: 1
Red	Inline role policy allows sts:AssumeRole *	Inline user policy allows NotActions	• Policies checked: 4 • Policies flagged: 1
Red	Inline user policy allows sts:AssumeRole *	Managed policy allows NotActions	• Policies checked: 28 • Policies flagged: 0
Green	Managed policy allows sts:AssumeRole *	Minimum password length too short	• Password policy checked: 1 • Password policy flagged: 1
Red	Password reuse enabled	Role with inline policies	
Green	Lack of MFA (root account)	Role with inline policies	

UPDATE OF A PARAMETERIZED RULE: GUIDE

1. Start the ruleset generator

```
Scout2RulesGenerator --ruleset-name demo
```

2. Find a parameterized rule

1. For example, Security > IAM > “Minimum password length too short”

3. Modify the parameter value

4. Click on the “Generate Ruleset” link in the top right corner

5. Run Scout2 with the new ruleset

```
./Scout2.py --profile <> --local --ruleset ~/Downloads/demo-ruleset.json
```

6. Confirm results have been updated accordingly

UPDATE OF A PARAMETERIZED RULE: UNDER THE HOOD

```
{  
  "about": "Short brief about the ruleset.",  
  "rules": {  
    ...  
    "iam-password-policy-minimum-length.json": [  
      {  
        "args": [  
          "8"  
        ],  
        "enabled": true,  
        "level": "danger"  
      }  
    ],  
    ...  
  }  
}
```

CUSTOM RULESETS

ENABLE AN
OPTIONAL RULE



ENABLE AN OPTIONAL RULE: EXPECTATIONS

The screenshot shows the Scout2 Report interface for an AWS environment. The top navigation bar includes links for Scout2, Analytics, Compute, Database, Management, Messaging, Network, Security, Storage, Regions, Filters, and Help. The main content area displays a grid of findings categorized by service and port status. Each finding is presented in a card with a red or orange background, indicating severity. Most findings are in red, while some like 'All ports open' and 'Use of port ranges' are in orange. Each card contains a title, a summary section with rule counts, and a detailed list of rules checked and flagged.

Category	Description	Rules Checked	Rules Flagged
DNS port open to all	MongoDB port open to all	72	1
MySQL port open to all	NFS port open to all	72	1
PostgreSQL port open to all	RDP port open to all	72	1
SSH port open to all	TCP port open to all	72	4
All ports open	UDP port open to all	72	2
FTP port open	Unrestricted network traffic within security group	38	17
Security group whitelists AWS CIDRs	Use of port ranges	75	1
	Unused security groups	62	32

ENABLE AN OPTIONAL RULE: EXPECTATIONS

The screenshot shows the Scout2 Report interface for an AWS environment. The top navigation bar includes links for Scout2, Analytics, Compute, Database, Management, Messaging, Network, Security, Storage, Regions, Filters, and Help. The main content area displays a grid of findings categorized by service and port status. Each finding is represented by a card with an information icon (i) and a 'Details' link (d). The findings are as follows:

Category	Description	Rules Checked	Rules Flagged
Ports Open to All	DNS port open to all	72	1
	MySQL port open to all	72	1
	PostgreSQL port open to all	72	1
	SSH port open to all	72	4
All ports open	75	20	
FTP port open	75	1	
Security group whitelists AWS CIDRs	72	24	
Ports Open to Specific	MongoDB port open to all	72	1
	NFS port open to all	72	1
	RDP port open to all	72	1
	TCP port open to all	72	4
Telnet port open	75	1	
Security group whitelists unknown CIDRs	72	16	
Ports Open to None	MsSQL port open to all	72	1
	Oracle DB port open to all	72	1
	SMTP port open to all	72	1
	UDP port open to all	72	2
Unrestricted network traffic within security group	38	17	
Use of port ranges	75	1	
Unused security groups	62	32	

At the bottom, there are dropdown menus for 'demo (2).json' and 'demo (1).json', a 'Show All' button, and a close button.

ENABLE AN OPTIONAL RULE: GUIDE

1. Start the ruleset generator

```
./Scout2RulesGenerator.py --ruleset-name demo
```

2. Find an optional rule (*i.e.* enabled check box not selected)

1. For example, Compute > EC2 > “Security group whitelists unknown CIDRs”

3. Enable the rule (*i.e.* tick the check box)

4. Click on the “Generate Ruleset” link in the top right corner and save the file

5. Run Scout2 with the new ruleset

```
./Scout2.py --profile <> --local --ruleset demo.json
```

6. Confirm a new rule exists in the corresponding service’s dashboard

ENABLE AN OPTIONAL RULE: LAB

Instructions provided during lab

CUSTOM RULES

REUSE A
PARAMETERIZED
RULE



REUSE A PARAMETERIZED RULE: GUIDE

1. Open the ruleset file in a text editor

```
<scout2_dir>/AWSScout2/rules/data/rulesets/default.json
```

2. Identify the parameterized rule reference

3. Create a new dictionary entry with a payload

```
{  
    "args": [  
        "Arg0-Value",  
        ...  
        "ArgN-Value"  
    ],  
    "enabled": true,  
    "level": "danger"  
}
```

4. Run Scout2 with the new ruleset

```
./Scout2.py --profile scout2lab --local --ruleset scout2-labXXX.json
```

REUSE A PARAMETERIZED RULE: LAB

Instructions provided during lab

CUSTOM RULES

CREATE A NEW RULE



CREATE A NEW RULE: GUIDE

1. Create a new rule file and open it in a text editor

```
<scout2_dir>/AWSScout2/rules/data/findings/demo.json
```

2. Create the new rule definition respecting the following schema

```
{  
    "description": "Name of finding in dashboard",  
    "rationale": "Why this rule matters",  
    "path": "service.path_to_resources.id",  
    "dashboard_name": "Name of resource in dashboard",  
    "conditions": [ "operator-and-or",  
        [ "path_to_test_value", "test_case", "expected_value" ],  
        [ "path_to_test_value", "test_case", "expected_value" ]  
    ],  
    "id_suffix": "path_to_highlighted_value"  
}
```

3. Start the Ruleset Generator, identify the new rule, enable it, and generate a new ruleset

4. Run Scout2 with the new ruleset

CREATE A NEW RULE: LAB

Instructions provided during lab

CUSTOM RULES

CREATE A NEW
PARAMETERIZED
RULE



CREATE A NEW PARAMETERIZED RULE: GUIDE

Similar to the creation of a new rule, but requires additional information

1. Name of the arguments: to be displayed in the ruleset generator
2. Key: unique identifier for the finding (defaults to the name of the file defining the rule)
3. Argument values must be referenced using the _ARG_N_ notation in any rule attribute

CREATE A NEW RULE: GUIDE

1. Create a new rule file and open it in a text editor

<scout2_dir>/AWSScout2/rules/data/findings/parameterized-demo.json

2. Create the new rule definition respecting the following schema

```
{  
    "description": "Name of finding in dashboard",  
    "arg_names": [ "arg_name_0", ..., "arg_name_N" ],  
    "key": "paramaterized-demo-_ARG_0_",  
    "rationale": "Why this rule matters",  
    "path": "service.path_to_resources.id",  
    "dashboard_name": "Name of resource in dashboard",  
    "conditions": [ "operator-and-or",  
        [ "path_to_test_value", "test_case", "_ARG_0_" ],  
        [ "path_to_test_value", "test_case", "expected_value" ]  
    ],  
    "id_suffix": "path_to_highlighted_value"  
}
```

CREATE A NEW RULE: GUIDE

1. Create a new rule file and open it in a text editor

<scout2_dir>/AWSScout2/rules/data/findings/parameterized-demo.json

2. Create the new rule definition respecting the following schema

```
{  
    "description": "Name of finding in dashboard",  
    "arg_names": [ "arg_name_0", ..., "arg_name_N" ],  
    "key": "paramaterized-demo-_ARG_0_",  
    "rationale": "Why this rule matters",  
    "path": "service.path_to_resources.id",  
    "dashboard_name": "Name of resource in dashboard",  
    "conditions": [ "operator-and-or",  
        [ "path_to_test_value", "test_case", "_ARG_0_" ],  
        [ "path_to_test_value", "test_case", "expected_value" ]  
    ],  
    "id_suffix": "path_to_highlighted_value"  
}
```

CREATE A NEW RULE: GUIDE

1. Create a new rule file and open it in a text editor

<scout2_dir>/AWSScout2/rules/data/findings/parameterized-demo.json

2. Create the new rule definition respecting the following schema

```
{  
    "description": "Name of finding in dashboard",  
    "arg_names": [ "arg_name_0", ..., "arg_name_N" ],  
    "key": "paramaterized-demo-ARG_0_",  
    "rationale": "Why this rule matters",  
    "path": "service.path_to_resources.id",  
    "dashboard_name": "Name of resource in dashboard",  
    "conditions": [ "operator-and-or",  
        [ "path_to_test_value", "test_case", "_ARG_0_" ],  
        [ "path_to_test_value", "test_case", "expected_value" ]  
    ],  
    "id_suffix": "path_to_highlighted_value"  
}
```

CREATE A NEW RULE: GUIDE

1. Create a new rule file and open it in a text editor

`<scout2_dir>/AWSScout2/rules/data/findings/parameterized-demo.json`

2. Create the new rule definition respecting the previous schema
3. Start the Ruleset Generator, identify the new rule, enable it, and generate a new ruleset
4. Run Scout2 with the new ruleset

CREATE A NEW PARAMETERIZED RULE: LAB

Instructions provided during lab



MIS|TI™ PRESENTS

InfoSecWorld

Conference & Expo 2018

**THANK YOU
PLEASE FILL OUT YOUR EVALUATIONS!**

Loic Simon

Director of Product Security @ Tanium
@l01cd3v