

# CSV Manager for AWS Security Hub

## Table of Contents

<b>Introduction.....</b>	<b>4</b>
Purpose – CSV Manager for Security Hub Exporter (csvExporter) .....	4
Purpose – CSV Manager for Security Hub Updater (csvUpdater) .....	4
GovCloud Compatibility & FedRAMP Status .....	4
Requirements .....	4
Environment .....	5
<b>Values for Configuration and Maintenance.....</b>	<b>7</b>
Values You Must Have .....	7
Values You May Want.....	7
<b>Process Descriptions .....</b>	<b>8</b>
Installing CSV Manager for Security Hub .....	8
Changing csvExporter Reporting Intervals .....	8
Updating the CSV Manager for Security Hub Code .....	9
Run an Unscheduled Export from Your Workstation .....	9
Run an Unscheduled Export Using an SSM Automation.....	9
Running an Update from a Command Prompt.....	10
Full CSV Manager for Security Hub Export & Update in GovCloud .....	11
<b>CSV Manager for Security Hub Updater (csvUpdater) Guide.....</b>	<b>14</b>
Updatable Columns .....	14
Update Value Rules.....	16
Editing csvExporter CSV Files.....	16
<b>Command and Executable Reference .....</b>	<b>18</b>

<b>Prepare Executable Package (csvPrepare.py)</b>	<b>18</b>
Important Notes	19
Example	19
<b>HubAcclerator Findings Updater (csvUpdater.py)</b>	<b>19</b>
Command Format	19
Important Notes	20
Command Examples	20
<b>CSV Manager for Security Hub Exporter (csvExporter.py)</b>	<b>21</b>
Command Format	21
Important Notes	22
Command Example	22
Lambda Format	23
Lambda Example	23
<b>CSV Manager for Security Hub Objects (csvObjects.py) Reference</b>	<b>24</b>
<b>CSV Reference</b>	<b>38</b>

## Introduction

CSV Manager for Security Hub exports SecurityHub findings to a CSV file and allows you to mass-update SecurityHub findings by modifying that CSV file.

---

### Purpose – CSV Manager for Security Hub Exporter (csvExporter)

csvExporter exports findings from AWS Security Hub to a file in CSV format and stores the results in an S3 bucket. You can use this CSV file to generate reports on your SecurityHub findings. csvExporter can be executed as a Unix command, as a Lambda function, or periodically using AWS EventBridge to invoke the csvExporter Lambda function.

---

### Purpose – CSV Manager for Security Hub Updater (csvUpdater)

csvUpdater allows the AWS Security Hub user or administrator to bulk-update specific fields in Security Hub findings by modifying columns in the CSV file generated by csvExporter. See the CsvExporter Update Guide for more details. csvExporter can be executed as a Unix command or as a Lambda function.

---

### GovCloud Compatibility & FedRAMP Status

CSV Manager for Security Hub is compatible with GovCloud. Refer to <https://aws.amazon.com/compliance/services-in-scope/> for the current FedRAMP levels of the services:

- CloudWatch
  - Security Hub
  - Systems Manager (Parameter Store)
  - S3 Storage
  - Security Token Service (STS, IAM)
- 

### Requirements

1. You MUST have the following:
  - a. A workstation, server, or EC2 instance with Python 3.9 or greater installed
  - b. AWS account with Security Hub enabled
  - c. AWS user credentials

2. You MAY need the following:
    - a. An AWS assumable role ARN if your credentials from 1.c. do not have sufficient privileges
- 

## Environment

1. CSV Manager for Security Hub CloudFormation stacks and the resources described below are installed in the Security Hub master account
2. CSV Manager for Security Hub use the following AWS services: S3, Security Hub, Security Token Service (STS, IAM), Parameter Store (SSM), and Automations (SSM)
3. CSV Manager for Security Hub include the following CloudFormation templates (CFTs)
  - a. The [CsvDatastore](#) template (CsvDatastore.yaml) – creates the S3 bucket and SSM parameters used by CSV Manager for Security Hub
  - b. The [CsvExporter](#) template (CsvExporter.yaml) – creates necessary resources for CSV Manager for Security Hub execution
4. CSV Manager for Security Hub consist of the following code:
  - a. [csvPrepare.py](#) – executed after the CSV Manager for Security Hub datastore is created to load the code into the datastore
  - b. [csvObjects.py](#) – contains classes used by other CSV Manager for Security Hub code
  - c. [csvExporter.py](#) – executed as a Lambda function or from the command line to extract Security Hub findings
  - d. [csvUpdater.py](#) – executed as a Lambda function or from the command line to update Security Hub findings
5. CSV Manager for Security Hub CloudFormation stacks defines the following resources:
  - a. CsvDatastore
    - i. An S3 bucket into which findings are exported (the CSV Manager for Security Hub datastore)
    - ii. An S3 bucket policy that provides access to the CSV Manager for Security Hub datastore
    - iii. Four SSM Parameter Store parameters for sharing between CloudFormation, Lambda, SSM, etc.
  - b. CsvExporter
    - i. An IAM role granting permissions to the Lambda function & SSM automation

- ii. A Lambda function that invokes `csvExporter.py`
- iii. An AWS EventBridge rule to periodically invoke the csvExporter Lambda function
- iv. A Lambda permissions entry enabling the EventBridge rule
- v. An SSM Automation Document that simplifies unscheduled csvExporter exports

## Values for Configuration and Maintenance

To install and maintain CSV Manager for Security Hub you need the following information:

---

### Values You Must Have

1. **<<ACCESSPRINCIPALS>>** - a list of AWS principals (account numbers, user ARNs, assumed role ARNs, etc.) who must access the CSV Manager for Security Hub datastore
  2. **<<AWSCREDENTIALS>>** - AWS credentials as a set of exports, or programmed in your .aws/config and .aws/credentials files, or some other mechanism, that provides baseline access to the installation account.
  3. **<<ROLEARN>>** - The role to be assumed when the [csvExporter.py](#) and [csvPrepare.py](#) programs are executed as commands. Must have sufficient privileges to perform all CSV Manager for Security Hub operations in S3, SSM, Lambda, etc.
  4. **<<FREQUENCY>>** - A valid EventBridge cron or rate expression that determines how frequently the csvExporter Lambda is executed by EventBridge
  5. **<<SOURCEDIRECTORY>>** - a directory on your workstation where the CSV Manager for Security Hub distribution code is stored or maintained
  6. **<<REGIONS>>** - a comma separated list of AWS regions for which you desire Security Hub findings
- 

### Values You May Want

7. **<<FILTERS>>** - either a canned filter name (HighActive is the only valid canned filter at this time) or a JSON object that meets the SecurityHub GetFindings API filter requirements.
8. **<<CODEFOLDER>>** - the folder in the target S3 bucket that contains the Lambda code archive (the default is "Code")
9. **<<FINDINGSFOLDER>>** - the folder in the target S3 bucket that contains findings exports (the default is "Findings")
10. **<<EXPIRATIONPERIOD>>** - the number of days before findings exports are deleted permanently (the default is 365)
11. **<<GLACIERTRANSITIONPERIOD>>** - the number of days before an export is migrated to AWS Glacier (the default is 31)

## Process Descriptions

Following are descriptions of some processes you might carry out with CSV Manager for Security Hub.

---

### Installing CSV Manager for Security Hub

You need the following values above to install CSV Manager for Security Hub:

- <<ACCESSPRINCIPALS>>
- <<AWSCREDENTIALS>>
- <<ROLEARN>>
- <<FREQUENCY>>
- <<SOURCEDIRECTORY>>

Perform the following steps to install CSV Manager for Security Hub:

1. Workstation – Store the CSV Manager for Security Hub distribution package (usually a ZIP archive) in a workstation directory <<SOURCEDIRECTORY>>
  2. Workstation – Login to the AWS console using <<AWSCREDENTIALS>>
  3. AWS Console – Build a CloudFormation stack [CsvDatastore](#) using the CsvDatastore.yaml CloudFormation template (CFT)
  4. Workstation – Export or configure your <<AWSCREDENTIALS>>
  5. Workstation – Execute the [csvPrepare.py](#) command to zip and store the Lambda code in the CSV Manager for Security Hub datastore
    - a. Note the “code archive S3 key” in the last message of [csvPrepare.py](#) output
    - b. Copy that value into your clipboard (or otherwise record it for the next step)
  6. AWS Console – Build a CloudFormation stack [CsvExporter](#) using the CsvExporter.yaml CloudFormation template (CFT)
    - a. Paste the value from 3.a. for the CodeArchive parameter
- 

### Changing csvExporter Reporting Intervals

You need the <<FREQUENCY>> value to update the reporting intervals.



1. Update the [CsvExporter](#) stack using the existing template
  2. Specify a valid EventBridge cron or rate interval in the Frequency parameter
- 

### Updating the CSV Manager for Security Hub Code

You need the `<<ROLEARN>>` and `<<SOURCEDIRECTORY>>` values to update the CSV Manager for Security Hub code.

1. Update the code as necessary
  2. Execute the `csvPrepare.py` command to zip and store the updated Lambda code in the CSV Manager for Security Hub datastore
  3. Note the message “(r) IMPORTANT! Provide the following value for the "code archive S3 key" parameter in the CsvExporter stack: `<<ARCHIVEKEY>>`”
  4. Update the [CsvExporter](#) and [CsvImporter](#) stacks using the existing template, providing the `<<ARCHIVEKEY>>` value for the CodeArchive parameter
- 

### Run an Unscheduled Export from Your Workstation

You need the `<<ROLEARN>>` to generate an extemporaneous export. You may want to specify `<<FILTERS>>` to limit the number of findings in your report.

1. Execute `csvExporter.py` as a command as described below
  2. Note the message “(i) Uploaded to S3 bucket `<<BUCKET>>` object `<<FINDINGSKEY>>`”
  3. View the CSV Manager for Security Hub datastore S3 `<<BUCKET>>` and download object `<<FINDINGSKEY>>`
- 

### Run an Unscheduled Export Using an SSM Automation

You may want to specify `<<FILTERS>>` to limit the number of findings in your report. The default for running the SSM automation is the HighActive canned filter.

1. Navigate to your SSM Automation Documents and find the CsvExporter automation
2. Scroll to the bottom of the displayed page
3. Press the **Execute** button
4. When the execution completes, click the **Step ID** in the **Execution Detail** section of the page

5. Note the following details in the **Output** section:

```
{
  "message": "Export succeeded",
  "bucket": "<<BUCKET>>",
  "exportKey": "<<FINDINGSKEY>>",
  "resultCode": 200
}
```

6. View the CSV Manager for Security Hub datastore S3 <<BUCKET>> and download object <<FINDINGSKEY>>

You can also navigate to the SSM Documents section and locate the SSM Automation associated with the CsvExporter stack and invoke the automation in that way.

---

## Running an Update from a Command Prompt

Running an csvUpdater update is best performed from a command prompt. csvUpdater is tested under Unix-like operating systems with the AWS CLI, AWS SDK, and Python 3.6 or greater installed. It may work under similarly configured Windows workstations, but that has not been tested.

1. Open a command line prompt and change to the directory where the CSV Manager for Security Hub code are stored
2. Perform an unscheduled export (in this example, the S3 bucket and folder initiated by the CsvDatastore CloudFormation stack is used by default). You should always create a new export before performing an update to ensure you have the most recent findings, including updates applied by others.

```
$ python3 ./csvExporter.py
```

```
INFO:root:005i supported regions default to ['region-1', 'region-2', 'region-3']
INFO:botocore.credentials:Found credentials in environment variables.
INFO:root:007i using environment credentials for ssm
INFO:root:007i using environment credentials for s3
INFO:root:007i using environment credentials for securityhub
INFO:root:016i retrieving findings from region region-1
INFO:root:016i retrieving findings from region region-2
INFO:root:016i retrieving findings from region region-3
INFO:root:019i retrieved          38 total findings from all regions
INFO:root:(i) Findings written to /tmp/SecurityHub-20210120-134534.csv
```

```
INFO:root:(i) Uploaded to S3 bucket <<BUCKET>> object <<KEY>>
INFO:root:(i) Local file deleted
```

3. Take note of the <<BUCKET>> and <<KEY>> values in the output from the command.
4. Make a local copy of the affected S3 object (this requires the AWS CLI to be installed, but you can also use the S3 interface of the AWS web console):

```
$ aws s3 cp s3://<<BUCKET>>/<<KEY>> ./Working.csv
download: s3://<<BUCKET>>/<<KEY>> to ./Working.csv
```

5. Use an editor or spreadsheet program to make changes in the local copy.
6. Perform an update using the local copy:

```
$ python3 ./csvUpdater.py -input ./Working.csv
INFO:root:005i supported regions default to ['region-1', 'region-2', 'region-3']
INFO:root:(i) reading from local file ./working.csv
INFO:botocore.credentials:Found credentials in environment variables.
INFO:root:007i using environment credentials for s3
INFO:root:007i using environment credentials for securityhub
INFO:root:(i) processing 38 records from CSV
INFO:root:(i) processed 37 findings and identified 8 update sets
INFO:root:(i) processing update sets
INFO:root:(i) 37 findings processed, 0 findings not processed
```

---

## Full CSV Manager for Security Hub Export & Update in GovCloud

The following is an example of running CSV Manager for Security Hub in GovCloud:

1. Open a command line prompt and change to the directory where the CSV Manager for Security Hub code are stored
2. Perform an unscheduled export (in this example, the S3 bucket and folder initiated by the CsvDatastore CloudFormation stack is used by default). You should always create a new export before performing an update to ensure you have the most recent findings, including updates applied by others.

```
$ ./csvExporter.py --partition=aws-us-gov --regions=us-gov-east-1
INFO:root:120d available regions for ssm in aws-us-gov are ['us-gov-east-1', 'us-gov-west-1']
```

```

INFO:root:120d available regions for s3 in aws-us-gov are ['us-gov-east-1', 'us-gov-west-1']
INFO:root:120d available regions for securityhub in aws-us-gov are ['us-gov-east-1', 'us-gov-west-1']
INFO:root:155d authorize ssm client region us-gov-east-1 partition aws-us-gov
INFO:botocore.credentials:Found credentials in environment variables.
INFO:root:170i using environment credentials for ssm
INFO:root:140d create ssm client in region us-gov-east-1
INFO:root:155d authorize s3 client region us-gov-east-1 partition aws-us-gov
INFO:root:170i using environment credentials for s3
INFO:root:140d create s3 client in region us-gov-east-1
INFO:root:155d authorize securityhub client region us-gov-east-1 partition aws-us-gov
INFO:root:170i using environment credentials for securityhub
INFO:root:140d create securityhub client in region us-gov-east-1
INFO:root:280i retrieving findings from region us-gov-east-1
INFO:root:320i retrieved      139 total findings from all regions
INFO:root:csvExporter.070i Findings written to /tmp/⟨⟨FILENAME⟩⟩
INFO:root:csvExporter.080i Uploaded to S3 bucket ⟨⟨BUCKET⟩⟩ object ⟨⟨KEY⟩⟩
INFO:root:csvExporter.100i Local file deleted

```

3. Take note of the ⟨⟨BUCKET⟩⟩ and ⟨⟨KEY⟩⟩ values in the output from the command.
4. Make a local copy of the affected S3 object (this requires the AWS CLI to be installed, but you can also use the S3 interface of the AWS web console):

```

$ aws s3 cp s3://⟨⟨BUCKET⟩⟩/⟨⟨KEY⟩⟩ ./Working.csv
download: s3://⟨⟨BUCKET⟩⟩/⟨⟨KEY⟩⟩ to ./Working.csv

```

5. Use an editor or spreadsheet program to make changes in the local copy.
6. Perform an update using the local copy:

```

$ ./csvUpdater.py --partition aws-us-gov --regions us-gov-east-1 --input s3://⟨⟨BUCKET⟩⟩/⟨⟨KEY⟩⟩
INFO:root:csvUpdater.120i reading from S3 bucket ⟨⟨BUCKET⟩⟩ object ⟨⟨KEY⟩⟩
INFO:root:120d available regions for ssm in aws-us-gov are ['us-gov-east-1', 'us-gov-west-1']
INFO:root:120d available regions for s3 in aws-us-gov are ['us-gov-east-1', 'us-gov-west-1']
INFO:root:120d available regions for securityhub in aws-us-gov are ['us-gov-east-1', 'us-gov-west-1']
INFO:root:155d authorize s3 client region us-gov-east-1 partition aws-us-gov
INFO:botocore.credentials:Found credentials in environment variables.

```

INFO:root:170i using environment credentials for s3  
INFO:root:140d create s3 client in region us-gov-east-1  
INFO:root:155d authorize securityhub client region us-gov-east-1 partition aws-us-gov  
INFO:root:170i using environment credentials for securityhub  
INFO:root:140d create securityhub client in region us-gov-east-1  
INFO:root:csvUpdater.010i processing 139 records from CSV  
INFO:root:csvUpdater.040i processed 138 findings and identified 5 update sets  
INFO:root:csvUpdater.050i processing update sets  
INFO:root:csvUpdater.060i 138 findings processed, 0 findings not processed

## CSV Manager for Security Hub Updater (csvUpdater) Guide

csvUpdater allows you to bulk-update Security Hub findings by modifying certain columns in the CSV file produced by csvExporter.

### Updatable Columns

The following columns can be modified using csvUpdater. If you change other columns in the CSV file, they will be silently ignored by csvUpdater.

Column Name	Description	Valid Values
<b>Confidence</b>	The percentage confidence you have in the finding	Integer from 0 to 100
<b>Criticality</b>	The importance of affected resources as assigned by the customer	Integer from 0 to 100
<b>NoteText</b>	Any note with which you wish to annotate the finding	String
<b>CustomerOwner</b>	The customer owner of the finding (email address, username, etc.)	String
<b>CustomerIssue</b>	A JIRA (or other tracking system) issue identifier (e.g., DSP-789)	String
<b>CustomerTicket</b>	A ServiceNow (or other ticketing system) ticket number	String
<b>ProductSeverity</b>	The native severity as defined by the AWS service or integrated partner product that generated the finding	Float
<b>SeverityLabel</b>	The severity value of the finding	String with fixed values <ul style="list-style-type: none"><li>• <b>INFORMATIONAL</b> - No issue was found.</li><li>• <b>LOW</b> - The issue does not require action on its own.</li><li>• <b>MEDIUM</b> - The issue must be addressed but not urgently.</li><li>• <b>HIGH</b> - The issue must be addressed as a priority.</li></ul>

		<ul style="list-style-type: none"> <li>• <b>CRITICAL</b> - The issue must be remediated immediately to avoid it escalating.</li> </ul>
<b>VerificationState</b>	Indicates the veracity of the finding	String with fixed values <ul style="list-style-type: none"> <li>• <b>UNKNOWN</b> – The default disposition of a security finding</li> <li>• <b>TRUE_POSITIVE</b> – The security finding is confirmed</li> <li>• <b>FALSE_POSITIVE</b> – The security finding was determined to be a false alarm</li> <li>• <b>BENIGN_POSITIVE</b> – A special case of TRUE_POSITIVE where the finding doesn't pose any threat, is expected, or both</li> </ul>
<b>Workflow</b>	The workflow status of the finding	String with fixed values <ul style="list-style-type: none"> <li>• <b>NEW</b> - The initial state of a finding, before it is reviewed.</li> <li>• <b>NOTIFIED</b> - Indicates that you notified the resource owner about the security issue. Used when the initial reviewer is not the resource owner, and needs intervention from the resource owner.</li> <li>• <b>RESOLVED</b> - The finding was reviewed and remediated and is now considered resolved.</li> </ul>

		<ul style="list-style-type: none"> <li>• <b>SUPPRESSED</b> - The finding will not be reviewed again and will not be acted upon.</li> </ul>
--	--	--

## Update Value Rules

The values you specify must conform to the specifications in the table above, but there are some additional rules to help ensure data integrity.

- You can specify “string with fixed values” columns in upper, lower, or mixed case—all are translated to uppercase
- Spaces are compressed and translated to underscores (“\_”) so “**false positive**” is equivalent to “**FALSE\_POSITIVE**”
- If you leave a column blank or set it to blank it will not be updated, which means it is not possible to set any “string” column back to a null string after you have set it to a non-null string. You should specify some value such as “-” to represent an empty string.

## Editing csvExporter CSV Files

You will probably want to edit csvExporter CSV files using Microsoft Excel or a similar spreadsheet program, though they can be edited using a text editor. The normal sequence of operations is:

- Download the desired CSV file from the <<BUCKET>> S3 bucket to an EC2 instance, AWS WorkSpace, or personal workstation.
- Edit the CSV file as necessary using the rules above
- Run csvUpdater.py against the local copy of the CSV file
- Upload the modified CSV file to S3 or some other network storage so you have a permanent and central record of changes

It is also possible to run csvUpdater.py against an S3 object without downloading the object to an EC2, WorkSpace, or workstation, which implies that you have edited the CSV file and then uploaded it to S3.

In all cases you should store the updated CSV file using a unique name in a central location—either by uploading the file to S3, placing it in a WorkDocs or OneDrive folder, or using some other form of network storage.

You should not directly modify the exported CSV file in S3 (you should make a copy), and you should not use the same CSV file for multiple updates. Always run an unscheduled csvExporter export and use that output CSV to make changes. That is because changes



made by one person may not be visible in the current scheduled csvExporter CSV, causing your changes to overwrite the others; and also because users may make changes to findings through the Security Hub web interface that will not appear until a new csvExporter export is performed.

## Command and Executable Reference

The following commands and executables are part of CSV Manager for Security Hub

### Prepare Executable Package (csvPrepare.py)

This is a command creates a CSV Manager for Security Hub distribution archive (ZIP file) containing all CSV Manager for Security Hub executable code. After csvPrepare.py is run, the [CsvExporter](#) stack should be created or updated using the output of this command (see below). Values in <<>> are as defined above.

```
csvPrepare.py [-h] [--role-arn <<ROLEARN>>] --source-directory <<SOURCEDIRECTORY>> [--bucket <<BUCKET>>] [--folder <<FILTERS>>]
               [--region <<REGION>>] [--partition <<PARTITION>>]
```

- **-h** – Obtain usage information
- **--role-arn** – Specify the assumable role to be used to access the target AWS account. Note that you must have configured or set basic AWS credentials on your workstation which can assume this role. If you do not specify a role, your basic AWS credentials must have sufficient authority for S3 and SSM.
- **--source-directory** – Specify the directory on your workstation where the CSV Manager for Security Hub distribution resides (required)
- **--bucket** – Specify an alternate S3 bucket in which to install the csvExporter Lambda archive. If not specified the bucket name is obtained from the SSM /cmsh/bucket parameter which is set in the [CsvDatastore](#) CloudFormation stack. You should generally allow the command to use the default.
- **--folder** – Specify an alternate folder within the S3 bucket to store the csvExporter Lambda archive. If not specified the bucket name is obtained from the SSM /cmsh/prefix/code parameter which is set in the [CsvDatastore](#) CloudFormation stack. You should generally allow the command to use the default.
- **--region** – The region for SSM and S3 operations. This defaults to **region-2**. If you are operating in GovCloud this parameter is required.
- **--partition** – The AWS partition in which operations will occur. This defaults to **aws** (AWS commercial). If you are operating in GovCloud this parameter is required and should be set to **aws-us-gov**.

This command obtains SSM and S3 API clients, zips all python files (.py) in the source directory, and uploads the resulting zip archive to the S3 bucket and code folder.

### Important Notes

*This command can only be successfully used after the [CsvDatastore](#) stack has been created*, at which time the target S3 bucket will exist and the necessary SSM parameters will be configured.

### Example

```
$ python3 csvPrepare.py --role-arn <<ROLE>> --source-directory ~/WorkDocs/customer/CSV Manager for Security Hub
```

(i) Assumed role <<ROLE>> for service ssm

(i) Assumed role <<ROLE>> for service s3

(r) IMPORTANT! Provide the following value for the "code archive S3 key" parameter in the [CsvExporter](#) stack: <<Folder>>/<<FILENAME>>.zip

**IMPORTANT!** Note the value in the last message from [csvPrepare.py](#), and enter it (or copy and paste it) into the CodeArchive parameter of the CsvExporter stack.

### HubAcclerator Findings Updater (csvUpdater.py)

This program can be used either as the target of an AWS Lambda function, or as a command line command. csvUpdater reads the specified input file, identifies updates in updatable columns, and separates those updates into *minimum update sets* (all findings with the same update are grouped together in a single API call) which are applied using the securityhub:batch\_update\_findings API.

### Command Format

The following is the format when [csvUpdater.py](#) is run as a command:

```
csvUpdater.py [-h] [--role-arn <<ROLEARN>>] [--debug] [--regions <<REGIONS>>] [--partition <<PARTITION>>]  
              -input <<S3OBJECTKEY>>|<<LOCALFILEPATH>>
```

- **-h** – Obtain usage information

- **--role-arn** – Specify the assumable role to be used to access the target AWS account. Note that you must have configured or set basic AWS credentials on your workstation which can assume this role. If you do not specify a role, your basic credentials must provide the necessary S3, SSM, and Security Hub access.
- **--debug** – Display more detail if an error occurs.
- **--partition** – Specify the AWS partition in which operations will take place. This defaults to **aws** (AWS commercial), and **aws-us-gov** must be specified if you are operating in GovCloud.
- **--regions** – Specify the AWS regions in which Security Hub findings will be updated. This defaults to region-1, region-2, and region-3. All regions specified must be within the partition specified above.
- **--input** – Is either an S3 object key or a local file path that points to a CSV file containing the necessary updates.
  - **<<S3OBJECTKEY>>** — an S3 object key in the format: **s3://bucket/key** and to which either the **<<ROLEARN>>** or your base credentials are authorize.
  - **<<LOCALFILEPATH>>** — a local file path that points to a CSV file containing the necessary updates to which your user account has access.

### Important Notes

1. **csvUpdater Outcomes.** The result csvUpdater.py may be successful, unsuccessful, or partially successful.
  - 1.1. A successful outcome occurs when all findings in the input CSV file are successfully processed and applied
  - 1.2. An unsuccessful outcome occurs when none of the input CSV findings are successfully processed and applied
  - 1.3. In some cases, there are errors in some of the updates but not others. In that case, the execution is considered partially successful. Shfu.py displays messages describing the details of any updates that fail.
2. **Minimum Update Sets.** You can take advantage of minimum update sets ( $\mu$ -sets) by grouping similar updates. Each update with an identical set of changes will be submitted as a single API call, which will reduce the chance of API errors such as throttling.

### Command Examples

Here is an example of using an S3 object:

```
$ python3 csvUpdater.py --input s3://<<BUCKET>>/<<KEY>>
```

```
INFO:root:005i supported regions default to ['region-1', 'region-2', 'region-3']
INFO:root:(i) reading from S3 bucket <<ROLE>> object <<KEY>>
INFO:botocore.credentials:Found credentials in environment variables.
INFO:root:007i using environment credentials for s3
INFO:root:007i using environment credentials for securityhub
INFO:root:(i) processing 38 records from CSV
INFO:root:(i) processed 37 findings and identified 8 update sets
INFO:root:(i) processing update sets
INFO:root:(i) 37 findings processed, 0 findings not processed
```

Here is an example of using a local file:

```
$ python3 csvUpdater.py --input <<LOCALFILEPATH>>
INFO:root:005i supported regions default to ['region-1', 'region-2', 'region-3']
INFO:root:(i) reading from local file <<LOCALFILEPATH>>
INFO:botocore.credentials:Found credentials in environment variables.
INFO:root:007i using environment credentials for s3
INFO:root:007i using environment credentials for securityhub
INFO:root:(i) processing 4 records from CSV
INFO:root:(i) processed 3 findings and identified 2 update sets
INFO:root:(i) processing update sets
INFO:root:(i) 3 findings processed, 0 findings not processed
```

---

## CSV Manager for Security Hub Exporter (csvExporter.py)

This program exports Security Hub findings to a CSV file, and can be used either as the target of an AWS Lambda function, or as a command line command.

### Command Format

The following is the format when `csvExporter.py` is run as a command:

```
csvExporter.py [-h] [--role-arn <<ROLEARN>>] [--partition <<PARTITION>>] [--regions <<REGIONS>>] [--bucket <<BUCKET>>]
                [--filters <<FILTER>>] [--limit count] [--retain-local]
```

- **-h** – Obtain usage information

- **--role-arn** – Specify the assumable role to be used to access the target AWS account. Note that you must have configured or set basic AWS credentials on your workstation which can assume this role. If this role is not specified, your basic credentials must have sufficient S3, SSM, and Security Hub permissions.
- **--partition** – Specify the AWS partition on which to operate. This defaults to **aws** and **aws-us-gov** must be specified if you are operating in GovCloud.
- **--regions** – Specify a comma separated list of AWS regions from which Security Hub findings will be exported. All regions must exist in the partition you specify above. This defaults to region-1, region-2, and region-3.
- **--bucket** – Specify an alternate S3 bucket into which to export Security Hub findings. If not specified the bucket name is obtained from the SSM CsvExporterBucket parameter which is set in the [CsvDatastore](#) CloudFormation stack. You should generally allow the command to use the default.
- **--filters** – Specify filters to limit the findings exported. These filters are specified as a string which describes a “canned” filter, or a JSON-formatted string that implements filters defined for the AWS Security Hub GetFindings API call. If not specified, no filters will be used, which may result in tens of thousands of exported findings. You may specify a string value of **HighActive** (the only currently defined canned filter) which exports only currently active findings with High or Critical severity.
- **--limit** – Limits the number of findings to return. This is primarily for testing when there are large numbers of findings. The first *count* findings are returned in all cases.
- **--retain-local** – Retains the local CSV file instead of discarding it after the put to the S3 bucket is complete. You should use this option with care as it may cause the accumulation of unwanted local files.

### Important Notes

If this command is used from a command line with an assumable role which extends trust to an AWS Isengard account, you must first go to the Isengard console, obtain temporary credentials, and paste them into your command line window.

Otherwise, you must use **aws config** or some other mechanism to configure credentials into your AWS profile that will be used as the base credentials for the assumable role.

### Command Example

```
$ python3 csvExporter.py --role-arn <<ROLEARN>> --filter HighActive
```

(i) Canned HighActive filter selects active high- and critical-severity findings

- (i) Assumed role <<ROLEARN>> for service ssm
- (i) Assumed role <<ROLEARN>> for service s3
- (i) Assumed role <<ROLEARN>> for service securityhub
- (i) Retrieving findings from region region-1
- (i) Retrieving findings from region region-2
- (i) Retrieving findings from region region-3
- (e) Cannot retrieve findings for region region-3: Account 111122223333 is not subscribed to AWS Security Hub
- (i) Retrieved 360 total findings from all regions
- (i) Findings written to <<LOCALFILEPATH>>
- (i) Uploaded to S3 bucket <<BUCKET>> object <<KEY>>
- (i) Local file deleted

The message prefixed by (e), which indicates an error, is not “normal,” but it does not prevent findings from being exported from the regions which are accessible when the command is executed. Messages prefixed by (s) or (t), which indicate severe or terminal errors, usually cause the command to fail.

## Lambda Format

When `csvExporter.py` is invoked as a Lambda function, the event descriptor may contain the following keys and their associated values:

- **role** – the <<ROLEARN>> value from above. Note that a role is not necessary for Lambda invocation, and if it is not specified, the execution environment credentials (e.g., the Lambda execution role) will be used for authorization.
- **partition** – the AWS partition (`aws`, `aws-cn`, `aws-us-gov`) which contains the resources
- **regions** – the regions <<REGIONS>> from above which findings will be exported (all regions must be in the associated partition)
- **filters** – the filters <<FILTERS>> from above which determine which findings will be selected. Unlike the command invocation, filters passed to the Lambda function are either a string (`HighActive`) or a JSON object—not a JSON-formatted string!
- **bucket** – an alternative S3 bucket to store findings, which defaults to the `SSM CsvExporterBucket` parameter value

## Lambda Example

Following is an example of the CloudWatch log output for a Lambda invocation of `csvExporter.py` using an EventBridge scheduled event.

```

2020-08-26T07:59:52.623-04:00 START RequestId: ebb2ca7b-9e56-44aa-8ada-457e61f09888 Version: $LATEST
2020-08-26T07:59:52.624-04:00 (i) Canned HighActive filter selects active high- and critical-severity
findings
2020-08-26T07:59:52.624-04:00 (i) Invoked by Scheduled Event
2020-08-26T07:59:52.624-04:00 (i) Using execution environment credentials for service ssm
2020-08-26T07:59:52.944-04:00 (i) Using execution environment credentials for service s3
2020-08-26T07:59:53.076-04:00 (i) Using execution environment credentials for service securityhub
2020-08-26T07:59:53.137-04:00 (i) Retrieving findings from region region-1
2020-08-26T07:59:53.996-04:00 (i) Retrieving findings from region region-2
2020-08-26T07:59:54.917-04:00 (i) Retrieving findings from region region-3
2020-08-26T07:59:57.663-04:00 (s) --> Cannot retrieve findings for region region-3: Account 111122223333
is not subscribed to AWS Security Hub
2020-08-26T07:59:57.663-04:00 (i) Retrieved 385 total findings from all regions
2020-08-26T07:59:57.762-04:00 (i) Findings written to <<LOCALFILEPATH>>
2020-08-26T07:59:57.900-04:00 (i) Uploaded to S3 bucket <<BUCKET>> object <<KEY>>
2020-08-26T07:59:57.900-04:00 (i) Local file deleted
2020-08-26T07:59:57.916-04:00 END RequestId: ebb2ca7b-9e56-44aa-8ada-457e61f09888
2020-08-26T07:59:57.916-04:00 REPORT RequestId: ebb2ca7b-9e56-44aa-8ada-457e61f09888 Duration: 5292.65 ms
Billed Duration: 5300 ms Memory Size: 512 MB Max Memory Used: 93 MB Init Duration: 327.36 ms

```

## CSV Manager for Security Hub Objects (csvObjects.py) Reference

This is a python object library containing class and method definitions used by the other executables. You should refer to the distribution package for more details, including any public and private methods you may need.

### Modules

boto3	json	os	sys
botocore	logging	re	time

### Classes

Exception(builtins.BaseException)



FindingValueError

MalformedUpdate

object

Actor

HubActor

S3Actor

SsmActor

Finding

FindingActions

FindingColumn

FindingColumnMap

FindingUpdate

MinimumUpdateList

class **Actor**

Actor(regions=['region-1', 'region-2', 'region-3'], service=None, role=None)

An abstract class for API functions. The class defines clients in each of a set of supported regions, and then carries out actions using those clients.

**authorize()**

If no role is supplied to the actor, the authorization is implicit through the credentials already in the environment. Otherwise, use sts:assume\_role to gain the privileges associated with the supplied role ARN.

**getClient(region=None)**

Create an AWS API client associated with a specific region

### **getRegions(regions=None)**

Return the regions associated with this actor

---

### **primaryClient**

Return the client for the primary region

### **primaryRegion**

Return the first (primary) region from the supported regions list

## **class Finding**

Finding(initializer=None, actor=None)

Represents an AWS SecurityHub finding.

### Parameters

-----

initializer : dict or list

- The dictionary for a SecurityHub finding from the securityhub:get\_findings API call
- A list containing the values of a CSV record containing finding fields in specific columns

### Attributes

- mapping : list of FindingColumn – a list of FindingColumn objects mapping the securityhub:get\_findings dictionary to a CSV record with particular column names
- rowList : list – a list of values representing a CSV row
- rowMap : dict – a dict keyed by the column name, with each value being the value for that column.
- finding : dict – the nested dict returned by the securityhub:get\_findings API, or built from a CSV initializer (see below)
- source : type – set to dict if initialized by a finding dict, and list if it is initialized from a CSV row list

### **fullMap()**

This list maps CSV column names to sequences of nested keys in the Security Hub findings dictionary. See the FindingColumn object for details.

### **getFindingColumn(name=None)**

Return the FindingColumn object associated with a named column.

### **mapColumns(initializer=None)**

Convert a SecurityHub findings dictionary to a CSV row

### **mapFinding(initializer=None)**

Convert an csvExporter CSV record to a SecurityHub finding dictionary

---

### **columns**

Return a list of CSV column names.

### **keys**

Return a dict of key values for this finding. Key values are attributes with the isKey attribute that uniquely identify the finding.

## **class FindingActions**

A class containing static methods used to pre- and post-process values used in SecurityHub finding dictionaries and CSV records

Static methods defined here:

### **checkSeverity(value=0)**

Verify a Security Hub severity value, which must be an integer between 0 and 100, or a floating-point number provided by the source application

### **checkSeverityLabel(value=None)**

Verify a Security Hub severity label which must be one of the values in the list below. The comparison is case insensitive and an uppercase value is always returned.

Valid values are stored in FindingActions.\_SEVERITY\_LABELS

### **checkVerificationState(value=None)**

Verify a Security Hub verification state which must be one of the values in the list below. The comparison is case insensitive, converts whitespace to a single underscore ("\_") and an uppercase value is always returned.

Valid values are in FindingActions.\_VERIFICATION\_STATES

### **checkWorkflow(value=None)**

Verify a Security Hub workflow label which must be one of the values in the list below. The comparison is case insensitive and an uppercase value is always returned.

Valid values are in FindingActions.\_WORKFLOWS

### **delist(list=[])**

Convert a list into a newline-separated string

### **forceInteger(value=None)**

Force a value to be an integer.

### **noteUpdater(value=None, actor=None, finding=None)**

Return the principal ID of the user who is updating a note. This value is only set if there is a corresponding update to the NoteText attribute of the finding.

### **resources(resources=[])**

Convert a list of SecurityHub resources to a newline-separated string

---

Data descriptors defined here:

#### **\_\_dict\_\_**

dictionary for instance variables (if defined)

#### **\_\_weakref\_\_**

list of weak references to the object (if defined)

## class **FindingColumn**

FindingColumn(columnNumber=0, columnName=None, keys=None, isKey=False, isUpdatable=False, d2l=None, d2lParameters={}, l2d=None, l2dParameters={})

Map a SecurityHub finding dictionary to a set of CSV column. This object represents a single column.

### Parameters

columnName : str

The name assigned to the CSV column

keys : list of str

A list of keys used to access the value for this column in a securityhub:get\_findings API response.

isKey : boolean

True if this column will act as a key to uniquely identify a finding

isUpdatable : boolean

True if this column can be updated using securityhub:batch\_update\_findings

d2l : callable, None, or other

A transformation between the API dictionary and the CSV record value

d2lParameters : dict

A \*\* dictionary to be passed to the d2l transform if it is callable

l2d : callable, None, or other

A transformation between the CSV record and API dictionary value

l2dParameters : dict

A \*\* dictionary to be passed to the l2d transform if it is callable

### Notes

The d2l and l2d transforms should be combined, since in all cases so far identified, the transform is the same in both directions.

## **deep(dictionary={})**

Retrieve a nested item from a dict. For example, given { "A" : "B": {1}} and a self.**keys** attribute of ["A", "B"], this will return 1.

### **key(self)**

If this column is a key column, return its value. This method will return "none" if the self.**value** setter is not called first!

### **update(self)**

If this column is an updatable column, return its value. This method will return "None" if self.**value** setter is not called first!

---

### **rawValue**

The raw value that came from the API finding dictionary or the CSV column, depending on how this finding was initialized.

### **value**

Return the value transformed according to its source (the API finding or the CSV column)

## class **FindingColumnMap**

FindingColumnMap(map=[])

Maps all API values to CSV columns, essentially flattening the API dict into a list that can be appended to a CSV file. This class automatically numbers the columns.

Methods defined here:

### **\_\_getitem\_\_**(item=None)

Return an item if indexed by a column number object[number] or a dict key object[key].

### **\_\_len\_\_**()

Return the length of the column map.

## class **FindingUpdate**

FindingUpdate(finding=None)

Represents an update to a Security Hub finding. The update is separated into a key signature and an update signature. See `MinimumUpdateList` for more details.

#### **findingRegion**

Return the region associated with this finding. The region is parsed from the Id value of the associated finding, which should be an ARN.

#### **keyString**

Returns a string containing the key values for the finding as key=value pairs separated by vertical bars.

#### **signature**

Returns an update signature as a string composed of the sorted attribute names to be changed and their proposed values.

class **FindingValueError**(builtins.Exception)

This exception is thrown if some value in a finding is out of bounds, or if there are other problems in the import or update of a finding.

Method resolution order:

FindingValueError

builtins.Exception

builtins.BaseException

builtins.object

class **HubActor**(Actor)

HubActor(regions=[], role=None)

Perform Security Hub API actions.

Method resolution order:

HubActor

Actor

builtins.object

---

Methods defined here:

**downloadFindings**(regions=None, filters={}, limit=0)

Get findings from Security Hub using the securityhub:get\_findings API, applying filters as necessary, and limiting results as necessary.

**getFinding**()

Generator yields each successive finding from a previous downloadFindings operation.

**updateFindings**(region=None, parameters=None)

Update a finding. Parameters are generated by the MinimalUpdateList parameterSets method. This method returns the untouched response structure from the API call.

---

Methods inherited from Actor:

**authorize**()

If no role is supplied to the actor, the authorization is implicit through the credentials already in the environment. Otherwise, use sts:assume\_role to gain the privileges associated with the supplied role ARN.

**getClient**(region=None)

Create an AWS API client associated with a specific region

**getRegions**(regions=None)

Return the regions associated with this actor

---



Data descriptors inherited from Actor:

**primaryClient**

Return the client for the primary region

**primaryRegion**

Return the first (primary) region from the supported regions list

class **MalformedUpdate**(builtins.Exception)

There were errors in the update request

Method resolution order:

MalformedUpdate

builtins.Exception

builtins.BaseException

builtins.object

class **MinimumUpdateList**

Accumulate finding updates and structure the updates so that multiple findings with the same changes will only be submitted once.

**add(finding=None)**

Add a finding to the minimum update list. Findings are aggregated by their update signature (a unique string of keys and values to be changed). Multiple findings with the same update signature will be submitted as a single update to Security Hub.

**parameterSets()**

Generator to yield each update as a set of parameters to the securityhub:batch\_update\_findings API.

---

Static methods defined here:

**apply**(update=None, region=None, actor=None)

Static method to apply an update to SecurityHub findings. The update must be the value yielded by the parameterSets generator, and the actor must be a HubActor object.

This method should probably be moved to HubActor as an object method, it but suffices for now.

class **S3Actor**(Actor)

S3Actor(bucket=None, role=None, prefix='SecurityHub', suffix='.csv', folder='SecurityHub')

Perform AWS Simple Storage Service (S3) API operations

Method resolution order:

S3Actor

Actor

builtins.object

---

**buildFilename**(bucket=None, folder=None, name=None, extension=None)

Construct a fully qualified S3 name from the bucket, folder, filename, and extension.

**filePath**(directory='/tmp')

Return a local fully qualified file path.

**get**(file=None, bucket=None, key=None, split=False)

Retrieve an object from S3 or a local file and return the entire body.

Parameters

file : str – a local file path

bucket : str – mutually exclusive with file, specifies an S3 bucket name

key : str – mutually exclusive with file, specifies an S3 object key

### **parseS3Url(url=None)**

Parse an S3 URL into bucket and key components.

### **put(inputFile=None, outputObject=None)**

Store an object in the S3 bucket. The inputFile is read from the local filesystem and stored to S3 as outputObject.

---

### **filename**

Return the unique S3 key associated with this object.

### **objectKey**

Return an S3 object key from the "folder" and unique filename.

---

Methods inherited from Actor:

### **authorize()**

If no role is supplied to the actor, the authorization is implicit through the credentials already in the environment. Otherwise, use sts:assume\_role to gain the privileges associated with the supplied role ARN.

### **getClient(region=None)**

Create an AWS API client associated with a specific region

### **getRegions(regions=None)**

Return the regions associated with this actor

---

Data descriptors inherited from Actor:

**primaryClient**

Return the client for the primary region

**primaryRegion**

Return the first (primary) region from the supported regions list

class **SsmActor**(Actor)

SsmActor(role=None)

Perform systems manager (SSM) actions.

Method resolution order:

SsmActor

Actor

builtins.object

---

**getValue**(names=[])

Retrieve the value of an SSM parameter.


**putValue**(name=None, description=None, value=None, type='String')

Set the value of an SSM parameter.

---

Methods inherited from Actor:

**authorize**(self)



If no role is supplied to the actor, the authorization is implicit through the credentials already in the environment. Otherwise, use `sts:assume_role` to gain the privileges associated with the supplied role ARN.

**getClient(region=None)**

Create an AWS API client associated with a specific region

**getRegions(regions=None)**

Return the regions associated with this actor

---

Data descriptors inherited from Actor:

**primaryClient**

Return the client for the primary region

**primaryRegion**

Return the first (primary) region from the supported regions list

## CSV Reference

The comma-separated values (CSV) file produced by csvExporter and consumed by csvUpdater has the following format:

Column #	CSV Column Name	Security Hub Finding Dictionary Keys <sup>1</sup>	Key Field <sup>2</sup>	Updatable Field <sup>3</sup>
1	Id	["Id"]	True	
2	ProductArn	["ProductArn"]	True	
3	Criticality	["Criticality"]		True
4	Confidence	["Confidence"]		True
5	NoteText	["Note", "Text"]		True
6	NoteUpdatedBy	["Note", "UpdatedBy"] <sup>4</sup>		
7	CustomerOwner	["UserDefinedFields"] <sup>5</sup>		True
8	CustomerIssue	["UserDefinedFields"]		True
9	CustomerTicket	["UserDefinedFields"]		True
10	ProductSeverity	["Severity", "Product"]		True
11	NormalizedSeverity	["Severity", "Normalized"]		

---

<sup>1</sup> This is the sequence of keys in the securityhub:get\_findings API response that supplies the value for this column. For example ["Workflow", "Status"] means that **finding["Workflow"]**["Status"] will provide the value for that column.

<sup>2</sup> Key fields uniquely identify a Security Hub finding

<sup>3</sup> Updatable fields are fields that can be set using the Security Hub Findings Updater (SFHU). All other fields are ignored when performing an csvUpdater update.

<sup>4</sup> This field is automatically set by csvUpdater whenever you update NoteText

<sup>5</sup> csvUpdater updates the UserDefinedFields dictionary using { **"Owner"**: owner, **"Issue"** : jiraIssue, **"Ticket"** : serviceNowTicket }

12	SeverityLabel	["Severity", "Label"]		True
13	VerificationState	["VerificationState"]		True
14	Workflow	["Workflow", "Status"]		True
15	UpdateVersion	N/A <sup>6</sup>		
16	GeneratorId	["GeneratorId"]		
17	AwsAccountId	["AwsAccountId"]		
18	Types	["Types"]		
19	FirstObservedAt	["FirstObservedAt"]		
20	LastObservedAt	["LastObservedAt"]		
21	CreatedAt	["CreatedAt"]		
22	UpdatedAt	["UpdatedAt"]		
23	Title	["Title"]		
24	Description	["Description"]		
25	StandardsArn	["ProductFields", "StandardsArn"]		
26	StandardsSubscriptionArn	["ProductFields", "StandardsSubscriptionArn"]		
27	ControlId	["ProductFields", "ControlId"]		
28	RecommendationUrl	["ProductFields", "RecommendationUrl"]		
29	StandardsControlArn	["ProductFields", "StandardsControlArn"]		
30	ProductName	["ProductFields", "aws/securityhub/ProductName"]		

---

<sup>6</sup> The UpdateVersion is a fixed value that must be set to "2021-01-01" if the CSV file is formatted correctly to work with csvUpdater

<b>31</b>	CompanyName	["ProductFields", "aws/securityhub/CompanyName"]		
<b>32</b>	Annotation	["ProductFields", "aws/securityhub/annotation"]		
<b>33</b>	FindingId	["ProductFields", "aws/securityhub/FindingId"]		
<b>34</b>	Resources	["Resources"]		
<b>35</b>	ComplianceStatus	["Compliance", "Status"]		
<b>36</b>	WorkflowState	["WorkflowState"]		
<b>37</b>	RecordState	["RecordState"]		