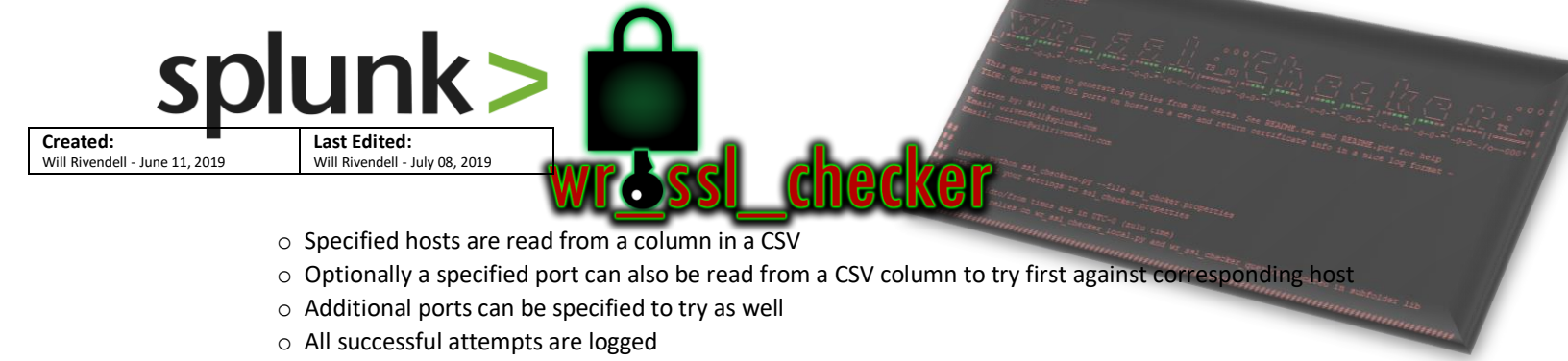
Created:	Last Edited:
Will Rivendell - June 11, 2019	Will Rivendell - July 08, 2019

Contents

About:

- wr_ssl_checker is a python app that gathers the details of SSL certs by checking specified hosts using SSL certificates on listening ports.

Created:	Last Edited:
Will Rivendell - June 11, 2019	Will Rivendell - July 08, 2019



- Specified hosts are read from a column in a CSV
- Optionally a specified port can also be read from a CSV column to try first against corresponding host
- Additional ports can be specified to try as well
- All successful attempts are logged
- A single failed attempt per host is logged if no specified ports respond
- Script can be run in standalone mode with an OS level cron and daily rotating logs are written to a specified folder
- Script can be run in "splunk" (default) mode where when placed in the $SPLUNK_HOME/etc/apps folder it will be run by Splunk and logs will be ingested directly rather than written to disk
- **Recommended** run interval, once daily at midnight

Usage:

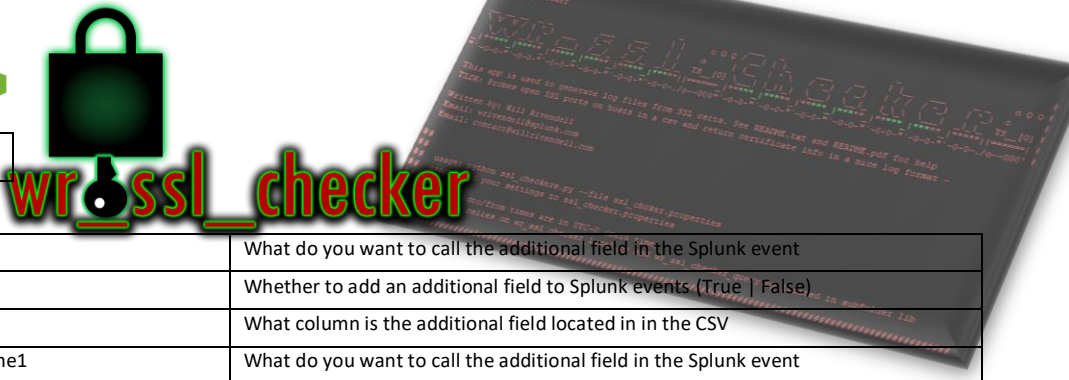
- **Standard CLI usage:** python wr_ssl_checker.py –file wr_ssl_checker.properties
- **Included sh script:** ./wr_ssl_checker.sh (literally just does the exact python command above)

Properties:

- wr_ssl_checker.properties

Property	Example value	Description
--csvpath	/opt/splunk/etc/apps/wr_ssl_checker/bin/csv/	full path to the folder containing the CSV (inc trailing /)
--csvname	test.csv	name of the csv (inc extension)
--csvheaders	True	whether the CSV has headers (True | False)
--host_col_num	0	What column are the hostnames/domains in (columns start at 0 not 1)
--port_col_num	1	What column is the first port to try in (columns start at 0 not 1) - leave as is if not using specified port in CSV
--only_use_specified_port	False	Whether to skip all additional ports and only check the one specified in the CSV column
--additional_ports	443 8443 8000 8089 9997 8889 990	Additional ports to check - all successful checks will be logged for the host
--only_use_additional_ports	True	Whether to skip the port in the column and just loop through the additional ports (True | False)
--getdesc	True	If you have a Description field in your CSV, you can add it to the Splunk fields (True | False)
--desc_col_num	1	If –getdesc is True, specify the column number to pull the desc from in the CSV
--getenvtype	True	If you have an Environment Type field in your CSV, you can add it to the Splunk fields (True | False)
--envtype_col_num	4	If –getenvtype is True, specify the column number to pull the type from in the CSV
--timeout	1	How long(sec) before timing out the current <host> <port> combo
--retry	1	How many times to retry before moving on to next <host> <port> combo (must be greater than 0)
--delay	0	How long(sec) to wait between retries
--standalone	False	When True, wr_ssl_checker functions in "standalone" mode. False is default and is "splunk" mode. See Standalone section of doc for more info.
--outputlogpath	/opt/splunk/etc/apps/wr_ssl_checker/bin/log/	(Standalone only) - where the logs should be written to on disk (inc trailing /)
--outputlogname	ssl_checker.log	(Standalone only) - the name of the output log. Log will be created if doesn't exist and Current DAY will be auto prefixed and subsequent logs will be appended until the next day
--enablelogroll	True	(Standalone only) - Delete older logs after x days (True | False)
--retentiondays	10	(Standalone only) - How many days old, does a log need to be before its removed?
--addl1	True	Whether to add an additional field to Splunk events (True | False)
--addl1_col_num	5	What column is the additional field located in in the CSV

Created:	Last Edited:
Will Rivendell - June 11, 2019	Will Rivendell - July 08, 2019



--addl1_field_name	Cert_Type	What do you want to call the additional field in the Splunk event
--addl2	False	Whether to add an additional field to Splunk events (True | False)
--addl2_col_num	7	What column is the additional field located in in the CSV
--addl2_field_name	my_custom_field_name1	What do you want to call the additional field in the Splunk event
--addl3	False	Whether to add an additional field to Splunk events (True | False)
--addl3_col_num	8	What column is the additional field located in in the CSV
--addl3_field_name	my_custom_field_name2	What do you want to call the additional field in the Splunk event
--addl4	False	Whether to add an additional field to Splunk events (True | False)
--addl4_col_num	15	What column is the additional field located in in the CSV
--addl4_field_name	my_custom_field_name3	What do you want to call the additional field in the Splunk event
--addl5	False	Whether to add an additional field to Splunk events (True | False)
--addl5_col_num	37	What column is the additional field located in in the CSV
--addl5_field_name	my_custom_field_name4	What do you want to call the additional field in the Splunk event
NOTE: all properties must be in this exact order in the file called wr_ssl_checker.properties **NOTE:** all properties start with two -'s, followed by a space and then the value(no spaces other than ports) **eg.** --csvname serverList.csv		

Standalone mode:

- When –standalone has a value of True in the properties file, logs will be written to disk as per the location and name in the properties file. A timestamp of the day will be pre-fixed automatically.
  - Eg. 2019-06-10-ssl_checker.log
- Logs will be retained for x days specified in the properties file at which point the oldest will be removed
- The app should still be placed in $SPLUNK_HOME/etc/apps/
- The inputs.conf file in $SPLUNK_HOME/etc/apps/wr_ssl_checker/local/ should be modified to reflect standalone
  - Uncomment the “monitor” stanza
  - Comment out the “script” and interval stanza
  - Adjust the output log folder to match your properties file if needed
  - **NOTE:** If you use a Deployment Server! You must move your log directory outside of the wr_ssl_checker app folder. Logs will get overwritten otherwise → /opt/splunk/var/log/ssl_log can be used for example
    - **Update wr_ssl_checker.properties to reflect this**
- Splunk will monitor the folder the logs are in
- Splunk will **NOT** run the script at all
  - In standalone mode you **must** manually run the script or put a CRON job in the OS cron

Created:	Last Edited:
Will Rivendell - June 11, 2019	Will Rivendell - July 08, 2019

Splunk(default) mode:

- When –standalone has a value of False in the properties file, no logs are written to disk and the output-specific log properties are not used
- Splunk will automatically run the script as per specified interval in $SPLUNK_HOME/etc/apps/wr_ssl_checker/local/**inputs.conf (can also be changed in the UI)**
  - **The "monitor" stanza line** should be commented OUT
- All lines are fed directly to Splunk during the scripts runtime
- No additional configuration is needed by default -> Copy app to etc/apps and that is all that is needed

splunk>

Created:	Last Edited:
Will Rivendell - June 11, 2019	Will Rivendell - July 08, 2019

wr_ssl_checker

Directory structure:

Created:	Last Edited:
Will Rivendell - June 11, 2019	Will Rivendell - July 08, 2019

Splunk conf files:

- Inputs.conf **– REMOVE this from Search Heads**

```
#[script:///opt/splunk/etc/apps/wr_ssl_checker/bin/wr_ssl_checker.sh]
#interval = 0 0 * * *
[monitor:///opt/splunk/var/log/ssl_log/*]
#### Uncomment monitor for standalone script w/ real log files --> need manual cron to run script i.e. 0 0 * * * (midnight each night)
#### Uncomment script AND interval for Splunk run script w/ real-time output --> interval splunk 0 0 * * * (midnight each night) (no manual cron needed)
index = ssl_cert
sourcetype = cert
disabled = 0
crcSalt = wr
```

- Props.conf

```
[cert]
TIME_PREFIX =
TIME_FORMAT = %Y-%m-%dT%H:%M:%S
MAX_TIMESTAMP_LOOKAHEAD = 25
LINE_BREAKER = ([\r\n]+)(\d{4}\-)
TRUNCATE = 10000
SHOULD_LINEMERGE = false
NO_BINARY_CHECK = true
TRANSFORMS = ssl_cert_hostoverride
description=SSL Cert Info
CHARSET=UTF-8
category=Web
disabled=false
```

- Transforms.conf

```
[ssl_cert_hostoverride]
DEST_KEY = MetaData:Host
REGEX = host=(\S+)
FORMAT = host::$1
```

splunk>

Created:	Last Edited:
Will Rivendell - June 11, 2019	Will Rivendell - July 08, 2019

wr_ssl_checker

- Macros.conf

```
[getRelativeEpochTime(2)]
args = tense,days
definition = relative_time(now(),"$tense$$days$d@d")
errormsg =
iseval = 0
validation = isnum($days$) AND $tense$="+" OR $tense$="-"

[withinDays(3)]
args = date_epoch,tense,days
definition = if($date_epoch$<=`getRelativeEpochTime($tense$,$days$)`, "True", "False")
iseval = 0
errormsg = Must enter a date in epoch format, then either + or - for future or past followed by an int for amount of days to check. `withinDays(1064543434243.00000,+,3)` for example
validation = isnum($days$) AND $tense$="+" OR $tense$="-"

[dateWithinDays(6)]
args = epoch_date,highest,second,third,fourth,lowest
definition = case(\
    `withinDays($epoch_date$,+,$highest$)`=="True" AND `withinDays(($epoch_date$),+,$second$)`=="False", $highest$, \
    `withinDays($epoch_date$,+,$highest$)`=="True" AND `withinDays(($epoch_date$),+,$second$)`=="True" AND `withinDays(($epoch_date$),+,$third$)`="False", $second$,\
    `withinDays($epoch_date$,+,$highest$)`=="True" AND `withinDays(($epoch_date$),+,$second$)`=="True" AND `withinDays(($epoch_date$),+,$third$)`="True" AND `withinDays(($epoch_date$),+,$fourth$)`=="False", $third$,\
    `withinDays($epoch_date$,+,$highest$)`=="True" AND `withinDays(($epoch_date$),+,$second$)`=="True" AND `withinDays(($epoch_date$),+,$third$)`="True" AND `withinDays(($epoch_date$),+,$fourth$)`=="True" AND `withinDays(($epoch_date$),+,$lowest$)`=="False", $fourth$,\
    `withinDays($epoch_date$,+,$highest$)`=="True" AND `withinDays(($epoch_date$),+,$second$)`=="True" AND `withinDays(($epoch_date$),+,$third$)`="True" AND `withinDays(($epoch_date$),+,$fourth$)`=="True" AND `withinDays(($epoch_date$),+,$lowest$)`=="True", $lowest$)
errormsg = First should be an epoch formatted date, followed by 5 integers. Ints should be highest amount of days to lowest. eg. to check `dateWithinDays((expiry_date),90,60,30,10,1)`
iseval = 0
validation = isnum($highest$) AND  isnum($second$) AND  isnum($third$) AND  isnum($fourth$) AND  isnum($lowest$)
```

Created:	Last Edited:
Will Rivendell - June 11, 2019	Will Rivendell - July 08, 2019

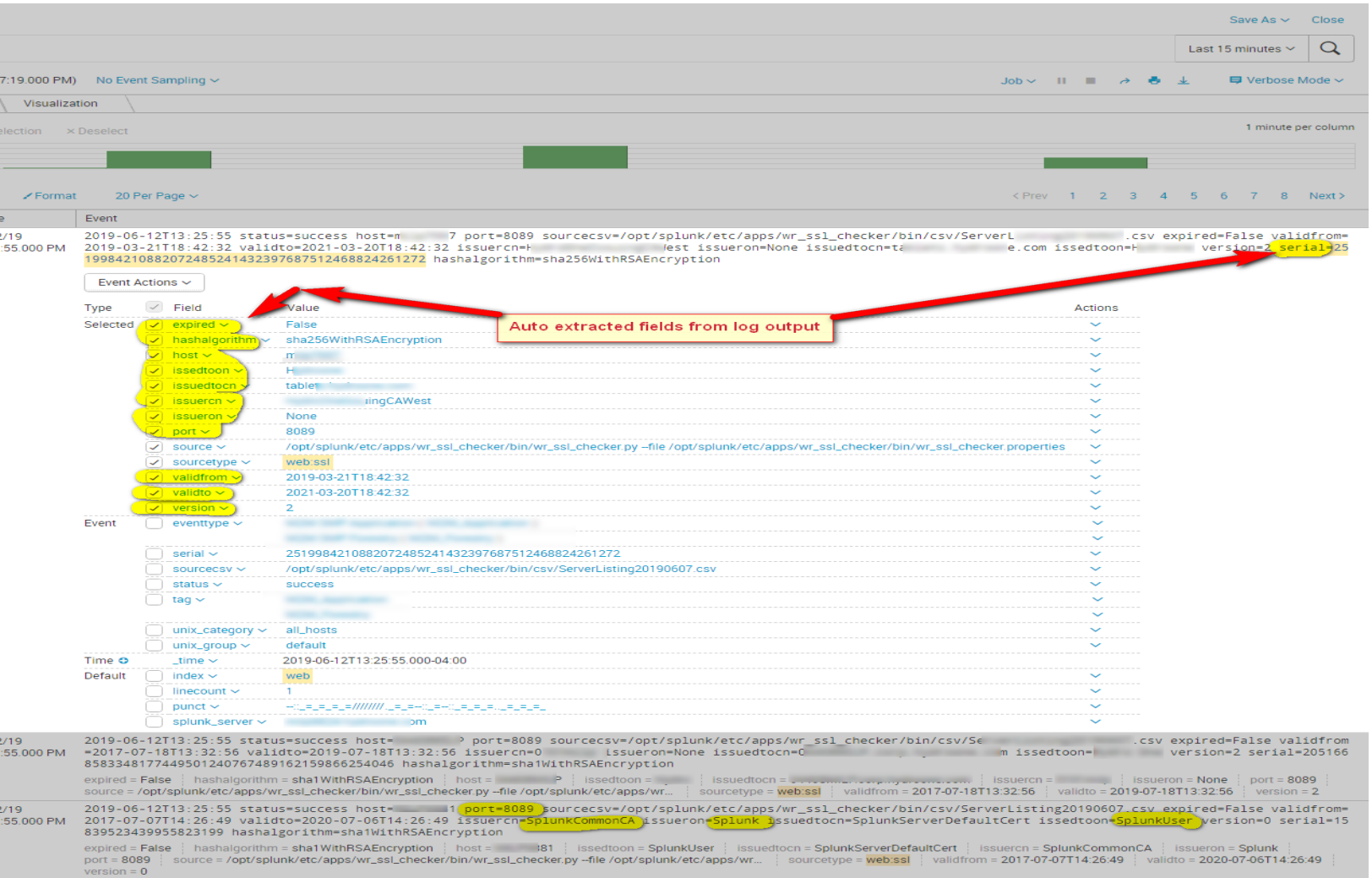
- Indexes.conf – **this stanza CAN be moved into an "org_all_indexes" app if preferred and then it's no longer needed in the app**

```
[volume:primary]
path = /opt/splunk/var/lib/warm

[volume:secondary]
path = /opt/splunk/var/lib/cold


[ssl_cert]
homePath = volume:primary/ssl_cert/db
coldPath = volume:secondary/ssl_cert/colddb
thawedPath = $SPLUNK_DB/ssl_cert/thaweddb
frozenTimePeriodInSecs = 2592000
disabled = 0
```

Splunk raw data in Search:



Created:	Last Edited:
Will Rivendell - June 11, 2019	Will Rivendell - July 08, 2019

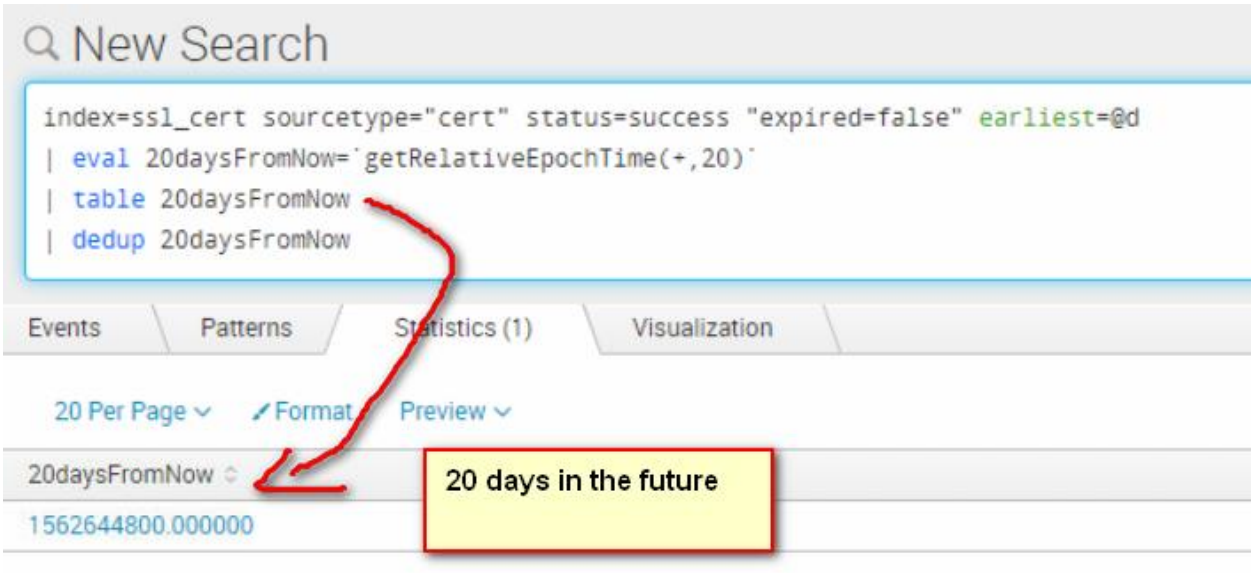
Dependencies

- Python:
  - Python 2.7.x
  - Python modules (all native) imports:
    - **argparse**
    - **os**
    - **csv**
    - **datetime**
    - **time**
    - **openSSL (from pyOpenSSL)**
    - **ssl, socket**
  - All of the above must be installed on the system running the script
  - If any of these are missing, you can install them with pip directly or with pip by downloading the packages and installing the package locally
  - Direct is quickest if system has access to the outside
    - Eg.
      - sudo yum install python-pip
      - apt-get install python-pip
        - pip install pyopenssl
- Splunk Macros:
  - dateWithinDays(6)
    - This macro takes an epoch date and then finds out wither or not that date fits within a set of five numbers and outputs that number accordingly
    - For example, the following search would find the expiry date of a cert in each event since midnight → convert it to epoch time →then find out if that epoch time is under 90, 60, 30, 10 or 1 days and assigns the appropriate number to variable “expireswithin_days”
    - Without this macro, all items expiring within 90 would also show up in 60 and 30 and so on… this macro handles all those cases
    - Numbers can be set to anything, but macro only supports five numbers as is

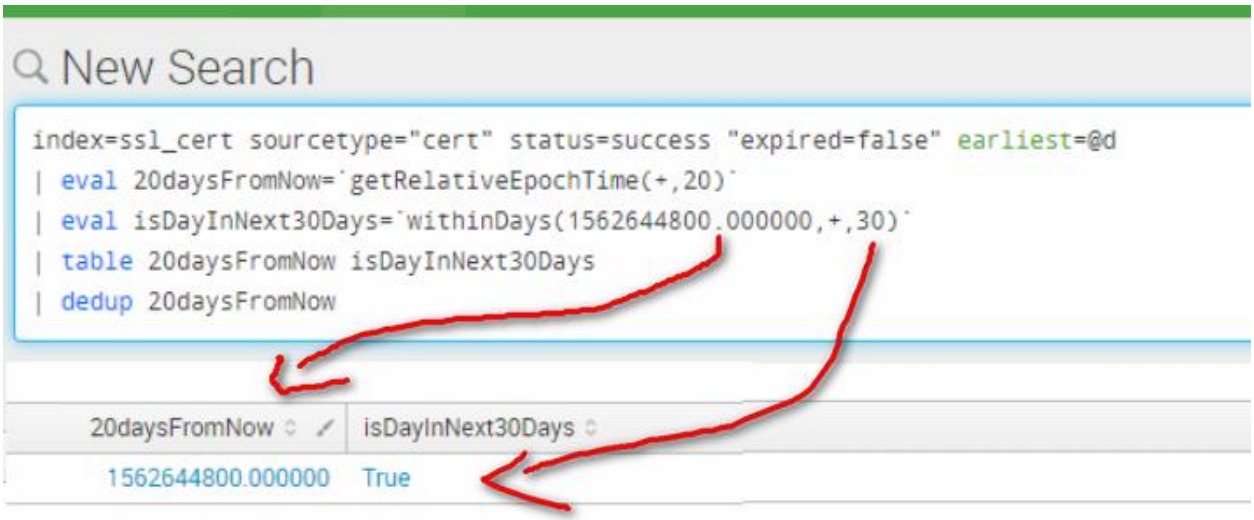
New Search

```
index=ssl_cert sourcetype="cert" status=success "expired=false" earliest=@d
| eval expiry_epoch=strptime(validto,"%Y-%m-%dT%H:%M:%S")
| eval expireswithin_days = `dateWithinDays(expiry_epoch,90,60,30,10,1)`
| search expireswithin_days=30
```

- **getRelativeEpochTime(2)**
  - Takes a date and time in Epoch format
  - Followed by a + or –
  - Followed by an integer (number of days)
  - Result is the time in Epoch of that calculation



- **withinDays(3)** (support for main macro)
  - Provide an epoch date, + or – and an integer for number of days
  - Macro will return True or False



Created:	Last Edited:
Will Rivendell - June 11, 2019	Will Rivendell - July 08, 2019

What goes where/when:

○ Search head:

- For dashboards and transforms etc
  - **Delete**: the wr_ssl_checker/**bin** folder
  - **Delete**: wr_ssl_checker/local/inputs.conf file
  - **Edit**: wr_ssl_checker/local/app.conf → visible set to **True**
  - **Indexes.conf:** stanza can be moved to an "org_all_indexes" if preferred

○ Heavy Forwarder/Indexer:

- Where the actual data collection script runs.
  - Best practice is to run from an HF → data collection not usually done on an indexer directly (but it would work)
  - **Edit**: wr_ssl_checker.properites to reflect your needs
  - Ensure the log folder you specify is read and writable by the splunk user (if using standalone)
  - **Edit**: inputs.conf accordingly
  - This host will need to be able to hit the hostnames/domains and ports specified in the csv → may require network changes
  - **Edit**: wr_ssl_checker/local/app.conf → visible set to **False**
  - **Indexes.conf:** stanza can be moved to an "org_all_indexes" if preferred

Dashboard Main Interactive: