



Contents

About:	1
Usage:	2
Properties:	2
Standalone mode:	3
Splunk(default) mode:	4
Directory structure:	5
Splunk conf files:	6
o Inputs.conf	6
o Props.conf	6
o Transforms.conf	6
o Macros.conf	7
o Indexes.conf	8
Splunk raw data in Search:	8
Dependencies:	9
o Python	9
o Splunk Macros	9
What goes where/when:	11
o Search head	11
o Heavy Forwarder/Indexer	11
Dashboard Main Interactive:	11

About:

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



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

--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

splunk>



wr_ssl_checker



Created:
Will Rivendell - June 11, 2019

Last Edited:
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



Directory structure:

```
[splunk@milp9831 wr_ssl_checker]$ ll -R
.:
total 20
drwx----- 5 splunk splunk 4096 Jun 19 11:51 ./
drwxr-x--- 34 splunk splunk 4096 Jun 19 11:51 ../
drwx----- 4 splunk splunk 4096 Jun 19 17:32 bin/
drwx----- 3 splunk splunk 4096 Jun 19 11:51 local/
drwx----- 2 splunk splunk 4096 Jun 19 11:51 metadata/

./bin:
total 28
drwx----- 4 splunk splunk 4096 Jun 19 17:32 ./
drwx----- 5 splunk splunk 4096 Jun 19 11:51 ../
drwx----- 2 splunk splunk 4096 Jun 19 11:51 csv/
drwx----- 2 splunk splunk 4096 Jun 19 17:32 lib/
-rwxr----- 1 splunk splunk 449 Jun 19 11:51 wr_ssl_checker.properties
-rwxr----- 1 splunk splunk 2655 Jun 19 11:51 wr_ssl_checker.py
-rwxr----- 1 splunk splunk 150 Jun 19 11:51 wr_ssl_checker.sh

./bin/lib:
total 32
drwx----- 2 splunk splunk 4096 Jun 19 17:32 ./
drwx----- 4 splunk splunk 4096 Jun 19 17:32 ../
-rwxr----- 1 splunk splunk 67 Jun 19 11:51 __init__.py
-rwxr----- 1 splunk splunk 3948 Jun 19 11:51 wr_ssl_checker_arguments.py
-rwxr----- 1 splunk splunk 1924 Jun 19 11:51 wr_ssl_checker_errors.py
-rwxr----- 1 splunk splunk 4808 Jun 19 11:51 wr_ssl_checker_local.py
-rwxr----- 1 splunk splunk 2330 Jun 19 11:51 wr_ssl_checker_query.py

./local:
total 32
drwx----- 3 splunk splunk 4096 Jun 19 11:51 ./
drwx----- 5 splunk splunk 4096 Jun 19 11:51 ../
-rw----- 1 splunk splunk 115 Jun 19 11:51 app.conf
drwx----- 3 splunk splunk 4096 Jun 19 11:51 data/
-rwxr----- 1 splunk splunk 486 Jun 19 11:51 inputs.conf
-rwxr----- 1 splunk splunk 2005 Jun 19 11:51 macros.conf
-rwxr----- 1 splunk splunk 284 Jun 19 11:51 props.conf
-rwxr----- 1 splunk splunk 90 Jun 19 11:51 transforms.conf

./local/data:
total 12
drwx----- 3 splunk splunk 4096 Jun 19 11:51 ./
drwx----- 3 splunk splunk 4096 Jun 19 11:51 ../
drwx----- 3 splunk splunk 4096 Jun 19 11:51 ui/

./local/data/ui:
total 12
drwx----- 3 splunk splunk 4096 Jun 19 11:51 ./
drwx----- 3 splunk splunk 4096 Jun 19 11:51 ../
drwx----- 2 splunk splunk 4096 Jun 19 11:51 views/

./local/data/ui/views:
total 28
drwx----- 2 splunk splunk 4096 Jun 19 11:51 ./
drwx----- 3 splunk splunk 4096 Jun 19 11:51 ../
-rwxr----- 1 splunk splunk 13588 Jun 19 11:51 cert_checker_main.xml
-rwxr----- 1 splunk splunk 219 Jun 19 11:51 default.xml

./metadata:
total 16
drwx----- 2 splunk splunk 4096 Jun 19 11:51 ./
drwx----- 5 splunk splunk 4096 Jun 19 11:51 ../
-rwxr----- 1 splunk splunk 59 Jun 19 11:51 default.meta
-rw----- 1 splunk splunk 837 Jun 19 11:51 local.meta
[splunk@milp9831 wr ssl checker]$
```

ROOT

csv location

supporting py libs

properties file for executable

main python executable

script to run in cron for ease

app props

dashboard and UI defaults

splunk conf files



Splunk conf files:

- Inputs.conf – REMOVE this from Search Heads

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

- Props.conf

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

- Transforms.conf

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




o Macros.conf

```

1 [getRelativeEpochTime(2)]
2 args = tense,days
3 definition = relative_time(now(),"$tense$$days$d@d")
4 errormsg =
5 iseval = 0
6 validation = isnum($days$) AND $tense$="+" OR $tense$="-"
7
8 [withinDays(3)]
9 args = date_epoch,tense,days
10 definition = if($date_epoch$<=`getRelativeEpochTime($tense$, $days$)` ,
11 "True", "False")
12 iseval = 0
13 errormsg = Must enter a date in epoch format, then either + or - for
14 future or past followed by an int for amount of days to check.
15 `withinDays(106454343243.00000,+,3)` for example
16 validation = isnum($days$) AND $tense$="+" OR $tense$="-"
17
18 [dateWithinDays(6)]
19 args = epoch_date,highest,second,third,fourth,lowest
20 definition = case(\
21   `withinDays($epoch_date$,+,$highest$)`=="True" AND
22   `withinDays(($epoch_date$),+,$second$)`=="False", $highest$, \
23   `withinDays($epoch_date$,+,$highest$)`=="True" AND
24   `withinDays(($epoch_date$),+,$second$)`=="True" AND
25   `withinDays($epoch_date$,+,$third$)`=="False", $second$, \
26   `withinDays($epoch_date$,+,$highest$)`=="True" AND
27   `withinDays(($epoch_date$),+,$second$)`=="True" AND
28   `withinDays($epoch_date$,+,$third$)`=="True" AND
29   `withinDays($epoch_date$,+,$fourth$)`=="False", $third$, \
30   `withinDays($epoch_date$,+,$highest$)`=="True" AND
31   `withinDays(($epoch_date$),+,$second$)`=="True" AND
32   `withinDays($epoch_date$,+,$third$)`=="True" AND
33   `withinDays($epoch_date$,+,$fourth$)`=="True" AND
34   `withinDays($epoch_date$,+,$lowest$)`=="False", $fourth$, \
35   `withinDays($epoch_date$,+,$highest$)`=="True" AND
36   `withinDays(($epoch_date$),+,$second$)`=="True" AND
37   `withinDays($epoch_date$,+,$third$)`=="True" AND
38   `withinDays($epoch_date$,+,$fourth$)`=="True" AND
39   `withinDays($epoch_date$,+,$lowest$)`=="True", $lowest$)
40 errormsg = First should be an epoch formatted date, followed by 5
41 integers. Ints should be highest amount of days to lowest. eg. to check
42 `dateWithinDays((expiry_date),90,60,30,10,1)`
43 iseval = 0
44 validation = isnum($highest$) AND isnum($second$) AND isnum($third$)
45 AND isnum($fourth$) AND isnum($lowest$)

```



Created:
Will Rivendell - June 11, 2019

Last Edited:
Will Rivendell - July 08, 2019

- o [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/coldddb
thawedPath = $SPLUNK_DB/ssl_cert/thawedddb
frozenTimePeriodInSecs = 2592000
disabled = 0
```

Splunk raw data in Search:

Auto extracted fields from log output

Type	Field	Value	Actions
Selected	expired	False	
	hashalgorithm	sha256WithRSAEncryption	
	host	m	
	port	8089	
	issuedto	table	
	issueron	ingCAWest	
	source	/opt/splunk/etc/apps/wr_ssl_checker/bin/wr_ssl_checker.py -file /opt/splunk/etc/apps/wr_ssl_checker/bin/wr_ssl_checker.properties	
	validfrom	2019-03-21T18:42:32	
	validto	2021-03-20T18:42:32	
	version	2	
Event	eventtype		
	serial	2519984210882072485241432397687512468824261272	
	sourcecsv	/opt/splunk/etc/apps/wr_ssl_checker/bin/csv/ServerListing20190607.csv	
	status	success	
	tag		
	unix_category	all_hosts	
	unix_group	default	
Time	_time	2019-06-12T13:25:55.000-04:00	
Default	index	web	
	linecount	1	
	punct	-=====	
	splunk_server	com	

2019-06-12T13:25:55 status=success host=m port=8089 sourcecsv=/opt/splunk/etc/apps/wr_ssl_checker/bin/csv/ServerListing20190607.csv expired=False validfrom=2017-07-18T13:32:56 validto=2019-07-18T13:32:56 issueron=None issuedto=table issueron=ingCAWest issuedto=table version=2 serial=2519984210882072485241432397687512468824261272 hashalgorithm=sha256WithRSAEncryption

2019-06-12T13:25:55 status=success host=m port=8089 sourcecsv=/opt/splunk/etc/apps/wr_ssl_checker/bin/csv/ServerListing20190607.csv expired=False validfrom=2017-07-18T13:32:56 validto=2019-07-18T13:32:56 issueron=None issuedto=table issueron=ingCAWest issuedto=table version=2 serial=2519984210882072485241432397687512468824261272 hashalgorithm=sha256WithRSAEncryption

2019-06-12T13:25:55 status=success host=m port=8089 sourcecsv=/opt/splunk/etc/apps/wr_ssl_checker/bin/csv/ServerListing20190607.csv expired=False validfrom=2017-07-18T13:32:56 validto=2019-07-18T13:32:56 issueron=None issuedto=table issueron=ingCAWest issuedto=table version=2 serial=2519984210882072485241432397687512468824261272 hashalgorithm=sha256WithRSAEncryption

2019-06-12T13:25:55 status=success host=m port=8089 sourcecsv=/opt/splunk/etc/apps/wr_ssl_checker/bin/csv/ServerListing20190607.csv expired=False validfrom=2017-07-18T13:32:56 validto=2019-07-18T13:32:56 issueron=None issuedto=table issueron=ingCAWest issuedto=table version=2 serial=2519984210882072485241432397687512468824261272 hashalgorithm=sha256WithRSAEncryption



Created:
Will Rivendell - June 11, 2019

Last Edited:
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 whether 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

Q 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

New Search

```
index=ssl_cert sourcetype="cert" status=success "expired=false" earliest=@d
| eval 20daysFromNow='getRelativeEpochTime(+,20)'
| table 20daysFromNow
| dedup 20daysFromNow
```

Events | Patterns | Statistics (1) | Visualization

20 Per Page ▾ Format Preview ▾

20daysFromNow ▾
1562644800.000000

20 days in the future

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

New Search

```
index=ssl_cert sourcetype="cert" status=success "expired=false" earliest=@d
| eval 20daysFromNow='getRelativeEpochTime(+,20)'
| eval isDayInNext30Days='withinDays(1562644800.000000,+,30)'
| table 20daysFromNow isDayInNext30Days
| dedup 20daysFromNow
```

20daysFromNow ▾	isDayInNext30Days ▾
1562644800.000000	True



What goes where/when:

o 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

o 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.properties 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:

