LA-UR-15-28545
Approved for public release; distribution is unlimited.

Title: Patriot Script 1.0.13 User Guide for PEM 1.3.2

Author(s): Cleland, Timothy James
Kubicek, Deborah Ann
Stroud, Phillip David
Cuellar-Hengartner, Leticia
Mathis, Mark

Intended for: Report

Issued: 2015-11-02

LOS ALAMOS NATIONAL LABORATORY

Patriot Script 1.0.13 User Guide for PEM 1.3.2

Timothy J. Cleland, Leticia Cueller, Deborah Kubicek, Mark Mathis, Phil Stroud

8/24/2015

A guide for using the Patriot Script application of PEM.

1 Introduction

Patriot Script is a lightweight desktop application that provides user access to the computationally intensive functions of the PATRIOT server. An analyst can set up attack scenarios, including the attack origin and destination locations, the attack device, and the GNDA options. Patriot Script calls on the PATRIOT server to generate attack routes and evaluate their reliabilities. Patriot Script results are output to a *comma separated values file* that can be used for statistical analysis. Routes generated by Patriot Script can be saved as .kml files for visualization in Google Earth or Patriot Viewer.

Patriot Script is one of three tools available to the analyst to access PATRIOT server functionality. The other two are Patriot Client and SWAMI. Patriot Script allows the analyst to set up a batch of scenarios, where the attack origin, attack destination, device, and GNDA options can all be varied. Patriot Client allows the analyst to run multiple attack origins and destinations, but for one device and GNDA option setting at a time. Whereas Patriot Client is designed for exploratory runs, Patriot Script is designed to execute massive study batch runs with tens of thousands of scenarios. Whereas Patriot Script is focused on vulnerability associated with transit, SWAMI integrates with the full attack model, to perform risk assessments including the adversary capabilities and consequences.

This document provides an updated user guide for Patriot Script Version 1.0.13, for release with PEM 1.3.1 (LAUR-1422817) that adds description and instructions for the new excursion capability (see section 4.5.1).

This document serves as both a quick start guide and a user guide and is organized as follows.

- Section 2 (page 1) describes how to prepare an Excel spreadsheet to specify the scenarios.
- Section 3 (page 5) describes how to open Patriot Script, load the input spreadsheet, and run the scenarios.
- Section 4 (page 7) gives an end-to-end demonstration of the analytic process of conducting a study with Patriot Script.

2 How to specify scenarios for Patriot Script

We first input the desired scenarios to be run in an excel spreadsheet that can be imported by the Patriot script. Four example spreadsheets are provided with the PEM 1.3.1 release:

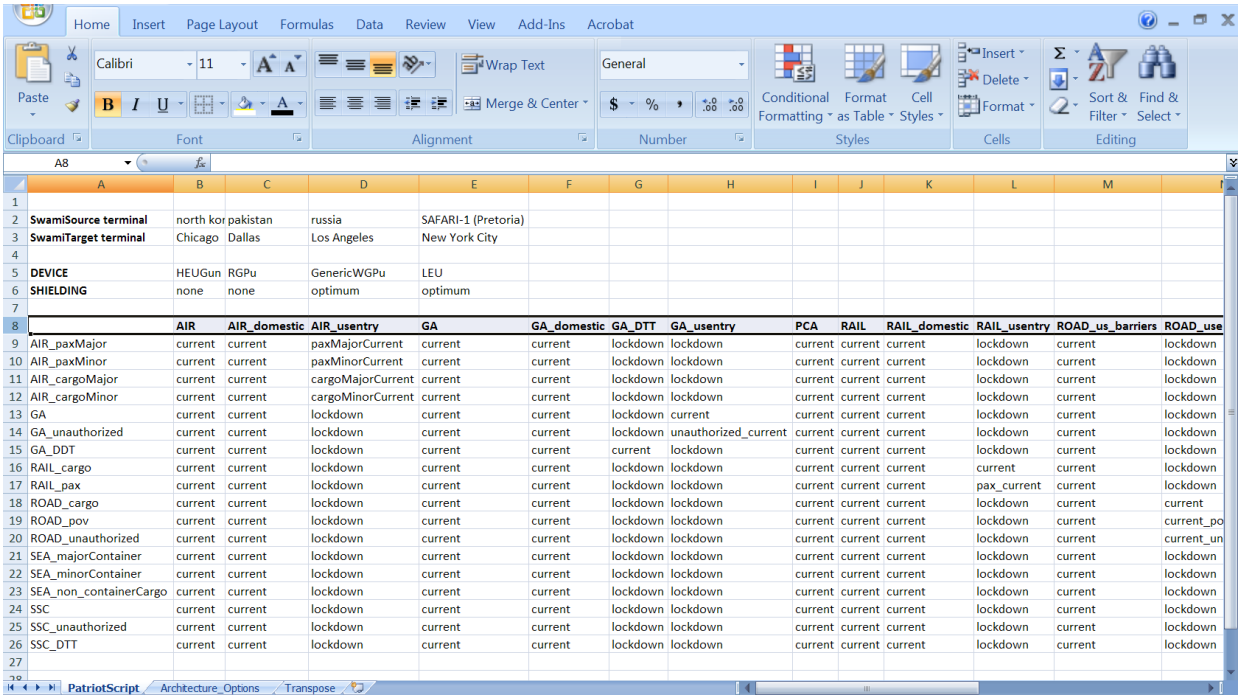
- PatriotScriptExample.xls
- patriot_script-testPatriot-1.3.1.xls
- Arch_options_template_Dec2014.xls
- PatriotScriptDemoRun1.3.1.xls

The second two spreadsheets were modified from their 1.3.0 release versions to provide a template for all 26 architecture options by including the

SSC_arctic, RAIL_us_ring, or ROAD_between_POE architecture options. If an Architecture Option column is omitted, the Architecture Level for that Architecture Option will default to “current”.

Figure 1 shows the format that Patriot Script requires. Note that the spreadsheet shown is for illustrative purposes only. That spreadsheet is not provided with PEM 1.3.1. Nevertheless, the current functionality of Patriot Script 1.0.12 is illustrated perfectly well with the spreadsheet shown.

Figure 1 Specification of Scenarios in an Excel-spreadsheet



	A	B	C	D	E	F	G	H	I	J	K	L	M	
1														
2	**SwamiSource terminal**	north kor	pakistan	russia	SAFARI-1 (Pretoria)									
3	**SwamiTarget terminal**	Chicago	Dallas	Los Angeles	New York City									
4														
5	**DEVICE**	HEUGun	RGPu	GenericWGPu	LEU									
6	**SHIELDING**	none	none	optimum	optimum									
7														
8		**AIR**	**AIR_domestic**	**AIR_usentry**	**GA**	**GA_domestic**	**GA_DTT**	**GA_usentry**	**PCA**	**RAIL**	**RAIL_domestic**	**RAIL_usentry**	**ROAD_us_barriers**	**ROAD_use**
9	AIR_paxMajor	current	current	paxMajorCurrent	current	current	lockdown	lockdown	current	current	current	lockdown	current	lockdown
10	AIR_paxMinor	current	current	paxMinorCurrent	current	current	lockdown	lockdown	current	current	current	lockdown	current	lockdown
11	AIR_cargoMajor	current	current	cargoMajorCurrent	current	current	lockdown	lockdown	current	current	current	lockdown	current	lockdown
12	AIR_cargoMinor	current	current	cargoMinorCurrent	current	current	lockdown	lockdown	current	current	current	lockdown	current	lockdown
13	GA	current	current	lockdown	current	current	lockdown	current	current	current	current	lockdown	current	lockdown
14	GA_unauthorized	current	current	lockdown	current	current	lockdown	unauthorized_current	current	current	current	lockdown	current	lockdown
15	GA_DDT	current	current	lockdown	current	current	current	lockdown	current	current	current	lockdown	current	lockdown
16	RAIL_cargo	current	current	lockdown	current	current	lockdown	lockdown	current	current	current	current	current	lockdown
17	RAIL_pax	current	current	lockdown	current	current	lockdown	lockdown	current	current	current	pax_current	current	lockdown
18	ROAD_cargo	current	current	lockdown	current	current	lockdown	lockdown	current	current	current	lockdown	current	current
19	ROAD_pov	current	current	lockdown	current	current	lockdown	lockdown	current	current	current	lockdown	current	current_po
20	ROAD_unauthorized	current	current	lockdown	current	current	lockdown	lockdown	current	current	current	lockdown	current	current_un
21	SEA_majorContainer	current	current	lockdown	current	current	lockdown	lockdown	current	current	current	lockdown	current	lockdown
22	SEA_minorContainer	current	current	lockdown	current	current	lockdown	lockdown	current	current	current	lockdown	current	lockdown
23	SEA_non_containerCargo	current	current	lockdown	current	current	lockdown	lockdown	current	current	current	lockdown	current	lockdown
24	SSC	current	current	lockdown	current	current	lockdown	lockdown	current	current	current	lockdown	current	lockdown
25	SSC_unauthorized	current	current	lockdown	current	current	lockdown	lockdown	current	current	current	lockdown	current	lockdown
26	SSC_DTT	current	current	lockdown	current	current	lockdown	lockdown	current	current	current	lockdown	current	lockdown
27														

A row-by-row description of the input spreadsheet follows. Some rows must be blank.

2.1 Row 1

Blank.

2.2 Row 2: source locations

Query strings for source locations are constructed from the entries in Row 2. One query is constructed for each non-empty cell starting with column B. Each query is composed of the query string in column A and the query string from one of the following columns. The query string in column A is thus included in each composed query. The analyst must ensure that the combined query string returns exactly one source location. This process generally involves some trial and error using the Origin tab in Patriot Client.

The query string “*swamisource terminal*” will return all of the terminal nodes in the PATRIOT database that contain the keywords *swamisource* and *terminal* in their searchable fields. The example shown generates 4 query strings: “swamisource terminal north korea”, “swamisource terminal

pakistan", "swamisource terminal russia", and "swamisource terminal safari-1(Pretoria)". Each of these strings returns exactly one source location from PATRIOT. The query strings are case insensitive.

2.3 Row 3: target locations

Row 3 is used in the same way as row 2, but to select a set of target locations.

2.4 Row 4

Blank.

2.5 Rows 5 and 6: devices

The set of attack devices is specified by a combination of one of the 8 PATRIOT device labels in row 5 and a PATRIOT shielding option in row 6. The device or material can be represented by various combinations of PATRIOT device label and PATRIOT shielding options. This mapping can also be found in the Architecture_Options worksheet in the Patriot Script template file Arch_options_template_Dec2014.xls, which is provided the PEM 1.3.1 release.

Table 1: Allowable selections for device and shielding

Allowable selections for device and shielding
GenericRGPu - Optimum
GenericWGPu - Optimum
HEU - Optimum
HEUGun - Optimum
HEUImpl – Optimum
LEU – Optimum
RGPu – Optimum
WGPu - Optimum
GenericRGPu - None
GenericWGPu - None
HEU - None
HEUGun – None
HEUImpl – None
LEU – None
RGPu – None
WGPu – None

2.6 Row 7

Blank.

2.7 Row 8: architecture options

Columns B through AA contain the names of the 26 PATRIOT architecture options shown in Table 2.

Table 2 PATRIOT Architecture Options

PATRIOT Architecture Options	
1. AIR 2. AIR_domestic 3. AIR_usentry 4. GA 5. GA_domestic 6. GA_DTT 7. GA_usentry 8. PCA 9. RAIL 10. RAIL_domestic 11. RAIL_usentry 12. RAIL_us_ring 13.ROAD_between_poe	14. ROAD_us_barriers 15. ROAD_usentry 16. ROAD_us_PRD 17. ROAD_us_rings 18. SCP 19. SEA 20. SEA_usentry 21. SLD 22. SLD_mobile 23. SSC 24. SSC_DTT 25. SSC_arctic 26. SSC_usentry

If an Architecture Option column is omitted, the Architecture Level for that Architecture Option will default to “current”.

2.8 Row 9 and following rows: Architecture Levels

Each of these rows serve to indicate the desired combination of architecture levels for each of the 26 PATRIOT architecture options. Column A serves to name the architecture levels combinations. Each GNDA excursion name needs to be unique.

For example, in the example shown in Figure 1, row 9 displays the “Air_paxMajor” GNDA excursion, where all U.S. modes of entry are set to *lockdown*, except for *AIR_usentry* (column D) that was set to *pax_MajorCurrent*, and all other options are set to *current.* For this specific choice of architecture options, Patriot Script will find the most reliable routes between all 4 sources and 4 targets and for all 4 devices producing a total of 64 (4×4×4) routes.

A template is provided with the distribution of this code under the Applications/Data folder called Arch_options_template_April2014.xls. Patriot Script does expect the *correct* spelling for each of the architecture levels. Table 3 contains all the architecture options with all of their respective architecture levels. The second worksheet of this template contains all the architecture levels, all the SWAMI sources, assembly, recovery and targets, along with all the devices names to make it easier to copy and paste into the first worksheet.

Since the Patriot Script uses the scenario descriptions (column A in rows 9 and on) as names for the output files, these fields must not be blank and must not contain the following characters: \ / : * ? “ < > |.

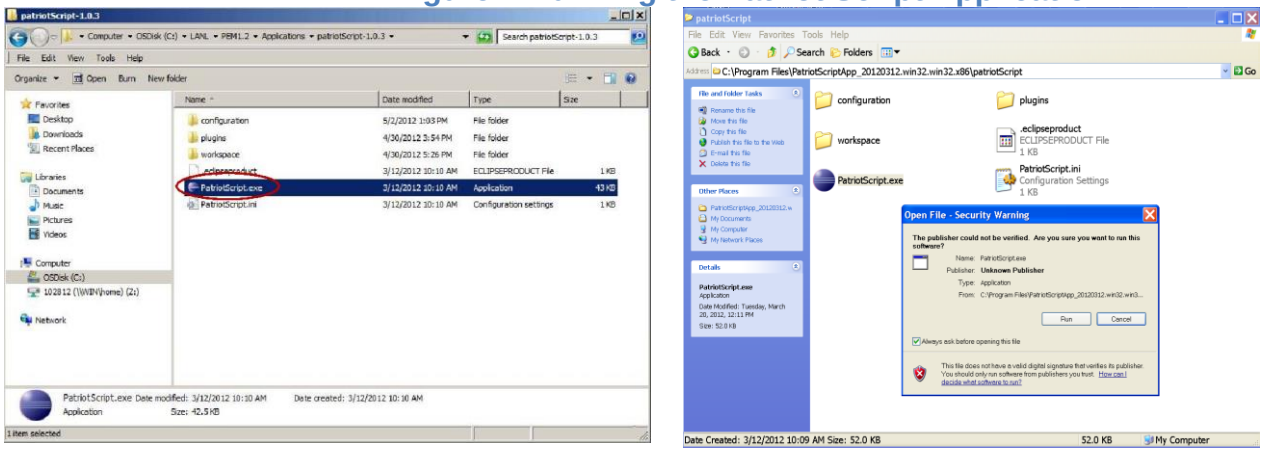
Table 3 Patriot Architecture Options and their Levels

AIR	current	lockdown_worldwide				
AIR_domestic	current	plus50	top26	top50	top75	top100
	lockdown					
AIR_usentry	current	paxCustomsBoost	paxMajorCurrent	paxMajorBoost	paxMinorCurrent	paxMinorBoost
	cargoMinorCurrent	cargoMinorBoost	majorEnhanced	minorEnhanced	lockdown	cargoMajorCurrent
	cargoMajorBoost					
GA	current	lockdown_worldwide				
GA_domestic	current	boost	enhanced	lockdown		
GA_DTT	current	plus50	enhanced	lockdown		
GA_usentry	current	boost	unauthorized_current	unauthorized_boost	enhanced	lockdown
PCA	current	enhanced				
RAIL	current	lockdown_worldwide				
RAIL_domestic	current	plus50	enhanced	lockdown		
RAIL_us_ring	current	boost	enhanced	lockdown		
RAIL_usentry	current	plus50	pax_current	pax_boost	enhanced	lockdown
ROAD_between_p	boost1	boost2	current	enhanced	lockdown	
ROAD_us_barriers	current	plus50desert	plus50mountain	plus50river	lockdown	
ROAD_usentry	current	plus50	current_pov	boost_pov	current_unauthorized	boost_unauthorized
	enhanced	lockdown	bypass50	bypass90		
ROAD_us_PRD	current	PRD_StatesInventory	PRD40k	HPT4k	HPT40k	heavyMDD20000
	heavyMDD4000					
ROAD_us_rings	current	PRD12percent	PRD96percent	HPT12percent	HPT24percent	HPT48percent
	plus50backpack	enhanced	lockdown	HPT96percent	plus50_rpm_equip	
SCP	current	enhanced	lockdown			
SEA	current	lockdown_worldwide				
SEA_usentry	current	plus50	major_container_current	major_container_boost	major_container_enhanced	minor_container_current
	enhanced	lockdown	energywindow	energywindowmasked	grosscount	minor_container_boost
	minor_container_enhanced					
SLD	current	plus50	customs_union	all_roads_off		
SLD_mobile	current	sld_4k	sld_20k			
SSC	current	lockdown_worldwide				
SSC_DTT	current	boost	enhanced	lockdown		
SSC_arctic	current	open				
SSC_usentry	allgroups	allUS80radhunterboom	allUS80radpack	allUS80stride700	current	enhanced
	group1	lockdown	noncontainerCargoBoost	noncontainerCargoCurrent	plus50backpack	plus50hpt
	svsd80radhunterboomHighRiskFirst	svsd80radpackByLane	svsd80radpackByRisk	svsd80radpackHighRiskFirs	svsd80stride700HighRiskFirst	unauthorizedBoost
	unauthorizedCurrent					

3 How to run Patriot Script

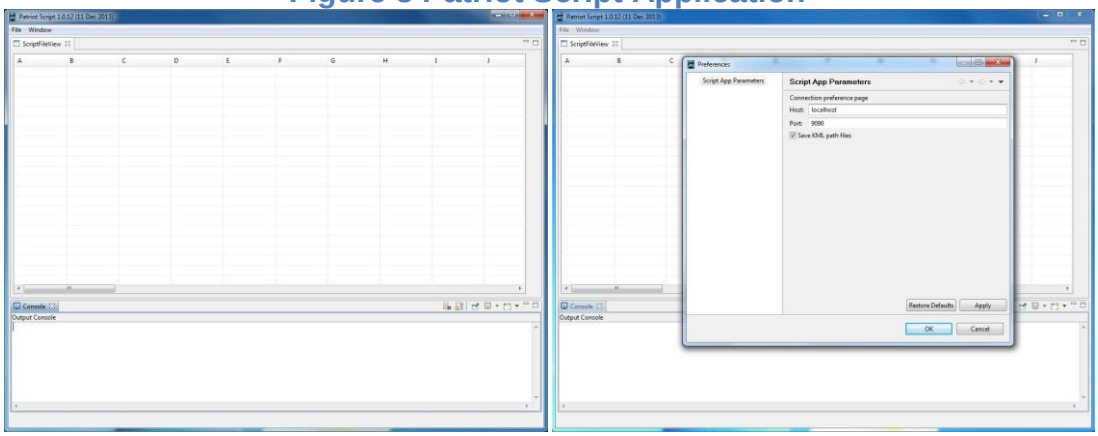
To run the Patriot Script application, double click on the Patriot Script icon located in C:LANL\PEM1.3\Applications\patriotscript.exe. Alternatively, it can be started by double-clicking on the Patriot Script icon on the desktop. If a new window appears (as the image shown on the right hand side of Figure 2), click on the “Run” tab. You can also uncheck the box indicating “Always ask before opening this file” to not have this window pop-up again.

Figure 2 Running the Patriot Script Application



Then the Patriot Script Application will open as shown in Figure 3.

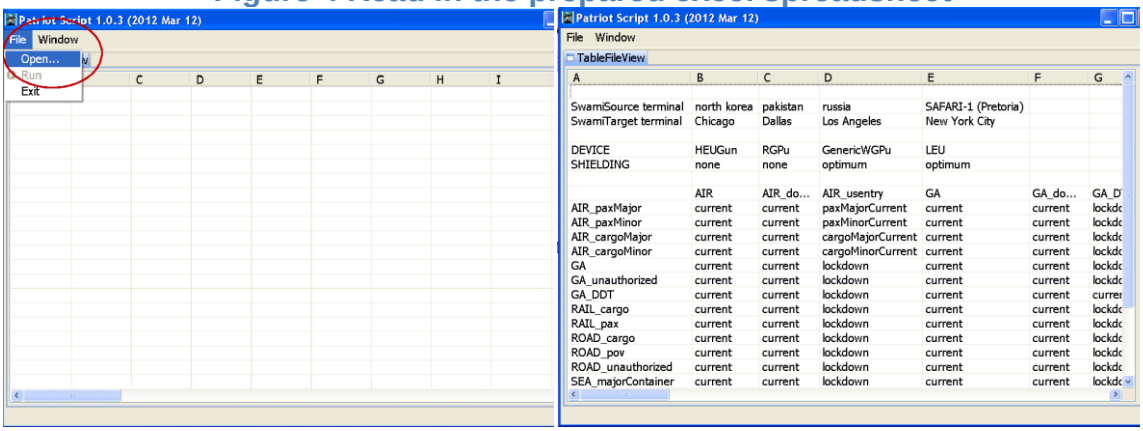
Figure 3 Patriot Script Application



To choose the server and port, go to the *Window* menu in the upper left corner and select *Preferences.* If running with the PEM 1.3 virtual machine, enter pem13 and 9090 for the host and port respectively. Note the preference option to “Save KML path files” is checked by default. That means that the KML path files will be saved to the output location in a separate “KML_AQ” directory. The KML files can be rather large and for large runs with a large number of paths it can be desirable at times to uncheck that option.

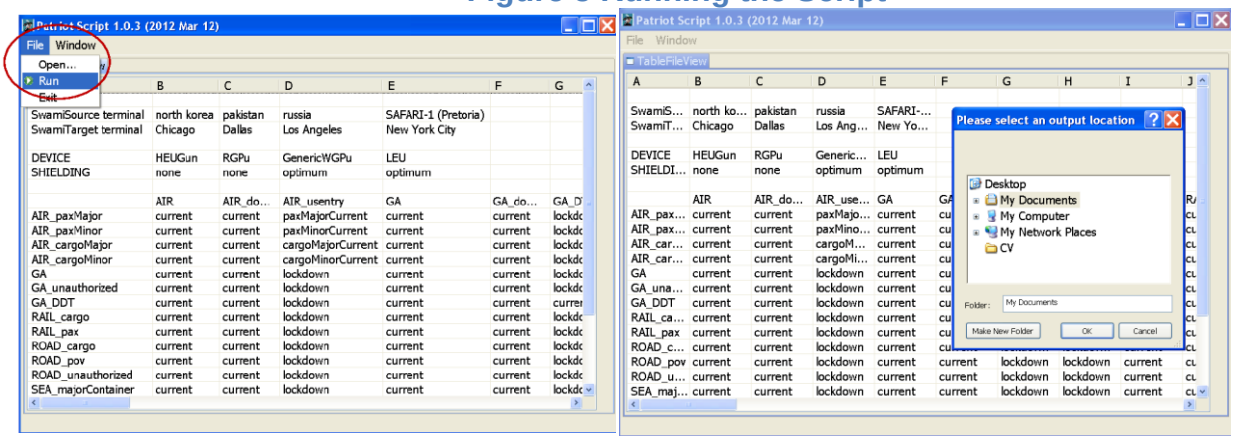
To read in the Excel Spreadsheet for the desired scenarios to be run, go to the FILE tab in the upper left corner, click on *Open* and select the file with the spreadsheet prepared as described in the previous section (see Figure 4).

Figure 4 Read in the prepared excel spreadsheet



To run the application, click again the FILE tab in the upper left corner, and select Run (see Figure 5). The Application will ask for a location where to save the results. Specify the directory where all the results will be saved.

Figure 5 Running the Script



In the directory chosen to save the results, the Patriot Script application will write a csv file with the output for all individual PATRIOT runs. This file can be read with Excel.

The Patriot Script application will also automatically create a new folder with all the KML files containing the most reliable paths. Each file will contain all the paths between all sources and targets for a given device and a given combination of architecture options.

If your directory is empty or missing one of the scenario files, verify that your scenario description is not blank or that it does not contain one of the disallowed characters \ / : * ? “ < > |. If your scenario descriptions are correct and you are still missing some output files, you can check the Patriot Script 1.0.12 log window to see if any other issues are occurring and verify all scenarios were run.

4 Patriot Script process demonstration

4.1 Open or create an input spreadsheet

Open a new Excel spreadsheet file and name it PatriotScriptDemoRun.xls. Alternatively, use the PatriotScriptDemoRun1.3.1.xls spreadsheet provided with the PEM 1.3.1 release.

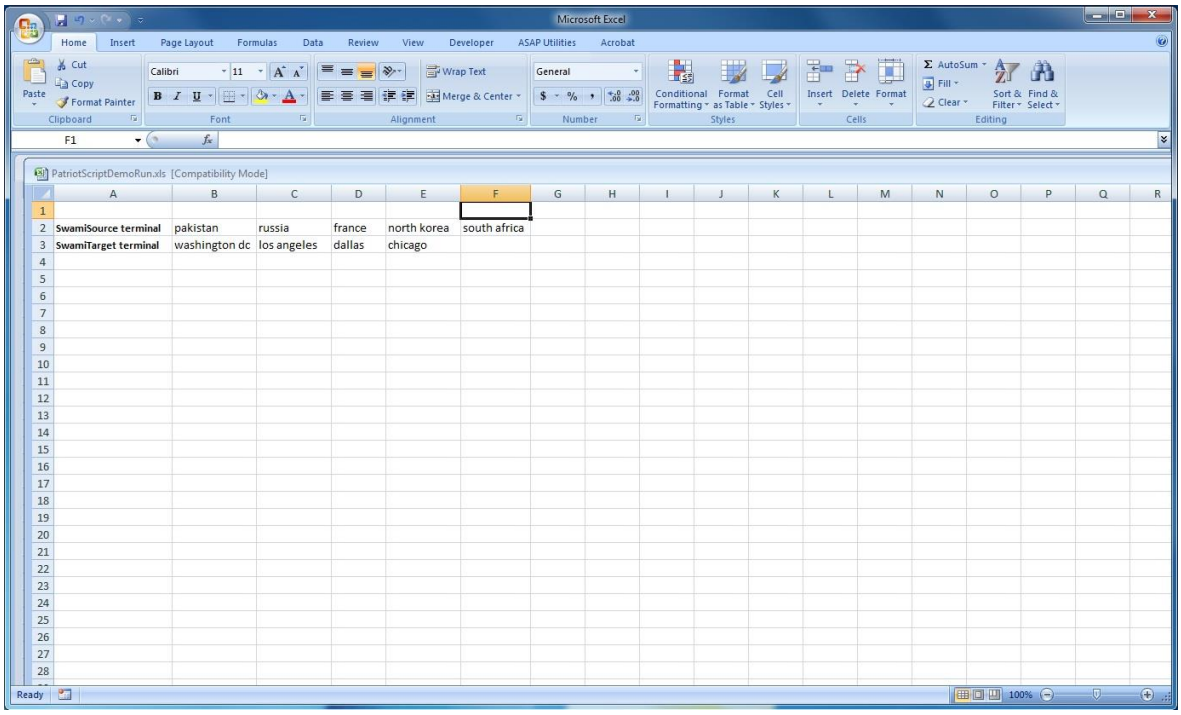
4.2 Specify the origin locations

For this run, we’ve chosen five points of origin: Pakistan, Russia, France, North Korea and South Africa. Leaving row 1 of the spreadsheet blank, in row 2 enter in the first column “swamisource terminal” (case insensitive). Then in the next five columns enter the five origins: Pakistan, Russia, France, North Korea and South Africa.

4.3 Specify the target locations

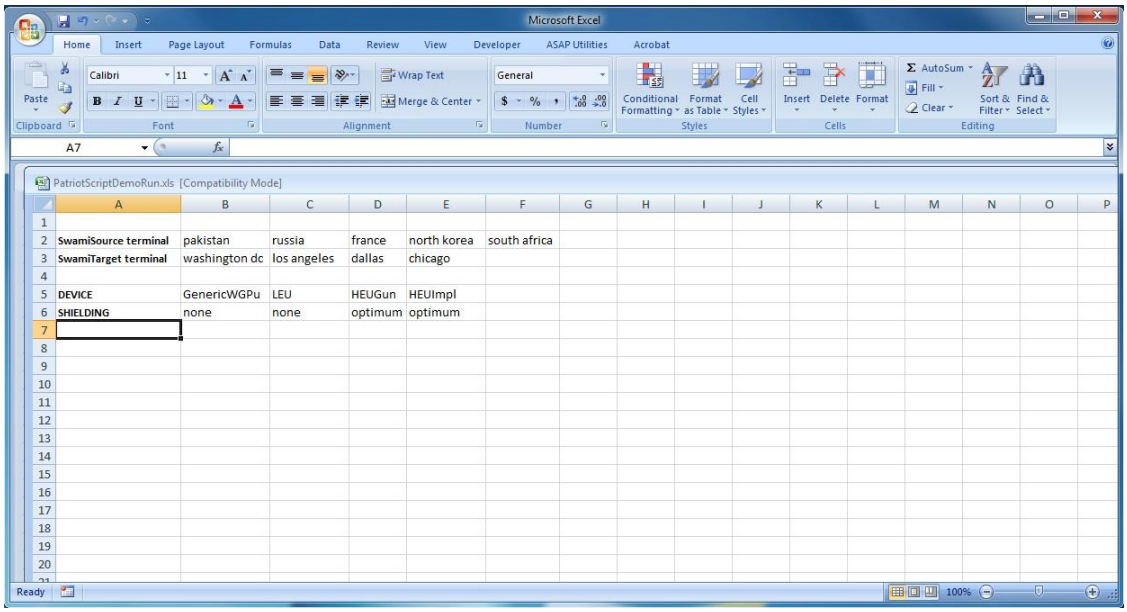
For this run we’ve chosen four target cities: Washington DC, Los Angeles, Dallas, and Chicago. In row 3 enter in the first column “swamitarget terminal” (case insensitive). Then in the next four columns enter:

Washington DC, Los Angeles, Dallas, and Chicago. At this point, your spreadsheet should look something like this:



4.4 Specify devices

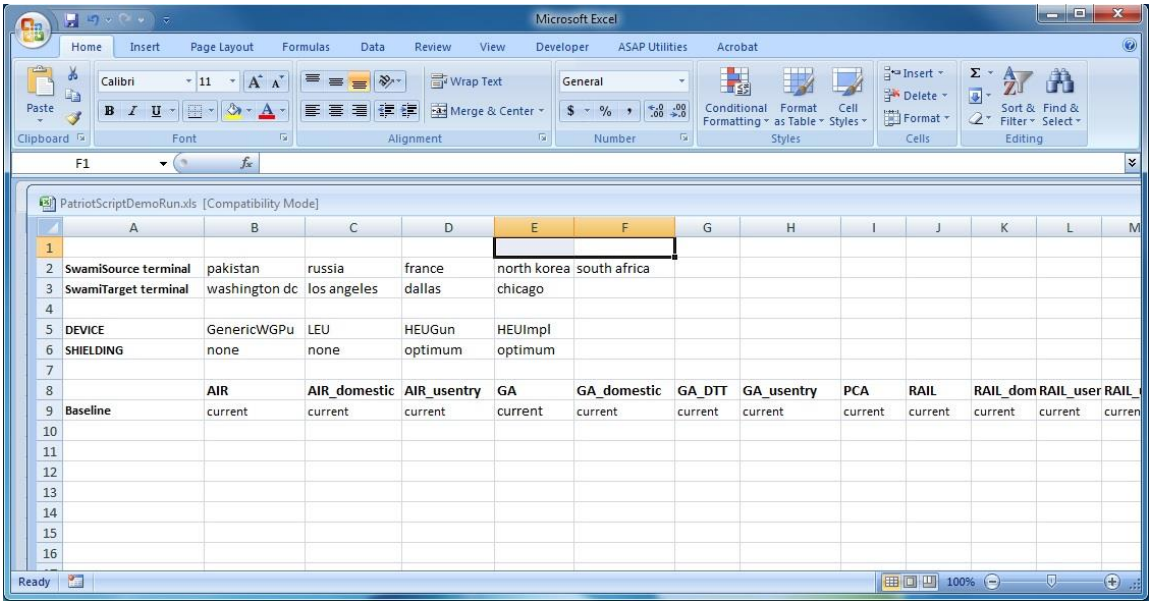
The next step is to choose your combination of devices and shielding levels. For this step, we make use of the Table 1 above or the file, Arch_options_template_Dec2014.xls referred to above. The desired devices listed can be modeled by combining Device/Shielding options. In our example, we've chosen the combinations of GenericWGPu/none, LEU/none, HEUGun/optimum, and HEUImpl/optimum (note that these names ARE case sensitive). Leaving row 4 blank, enter the word, "DEVICE", in the first column of row 5 and the word, "SHIELDING", in the first column of row 6. Then, in the columns after the word, "DEVICE", enter "GenericWGPu", "LEU", "HEUGun", and "HEUImpl" on row 5. Similarly, enter the words, "none", "none", "optimum", "optimum" in the columns following the word, "SHIELDING" on row 6. At this point, your file should look like this:



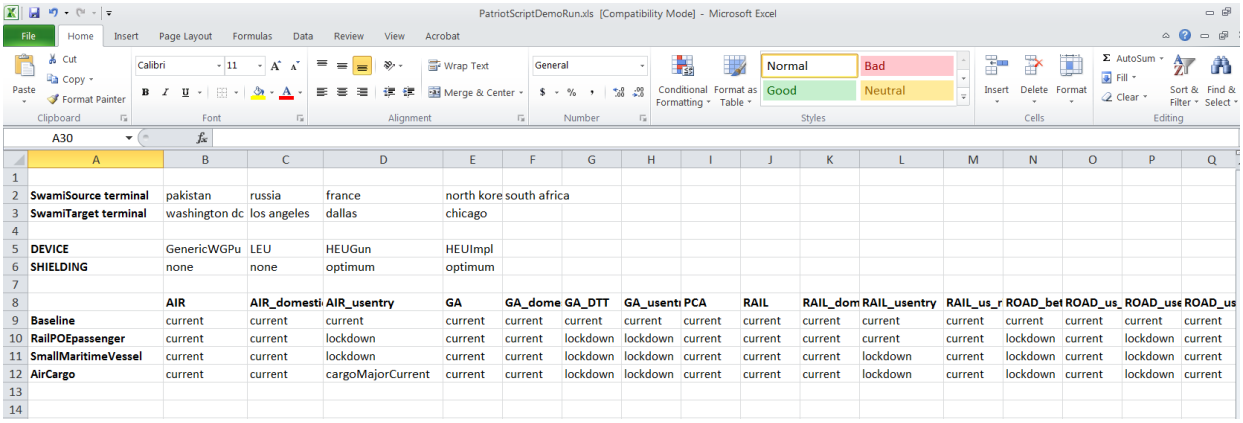
4.5 Specify the GNDA options

Leave row 7 blank, then go to the file Arch_options_template_Dec2014.xls, and on the page with the "Architecture_Options" tab, copy row 10 (the column headers, including the first blank column) and paste it onto row 8 of your spreadsheet.

We have chosen to model 4 scenarios (i.e., 4 sets of architecture option settings). The first one we'll call "Baseline" so enter the word, "Baseline", in the first column of row 9. It will consist of setting all the architecture levels to "current" (i.e. no change from status quo), so enter "current" under each of the column headers ("Copy/Paste" helps here too). Here is what your spreadsheet should look like at this point:



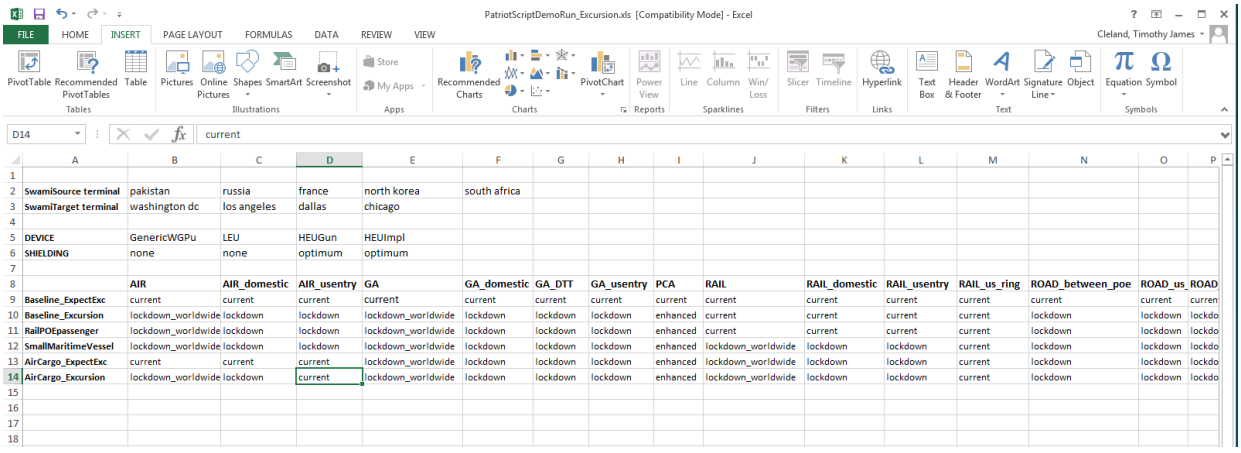
For the remaining scenarios (we've entitled them "RailPOEpassenger", "SmallMaritimeVessel", and "AirCargo"), it's probably easiest to just refer to the PatriotScriptDemoRun1.3.1.xls file and "Copy/Paste" the architecture level values. For the most part, all us_entry options are set to "lockdown" except for one of RAIL_usentry, SSC_usentry, or AIR_usentry. The non us_entry options remain set to "current".



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
1																	
2	**SwamiSource terminal**	pakistan	russia	france	north kore	south africa											
3	**SwamiTarget terminal**	washington dc	los angeles	dallas	chicago												
4																	
5	**DEVICE**	GenericWGPu	LEU	HEUGun	HEUImpl												
6	**SHIELDING**	none	none	optimum	optimum												
7																	
8		**AIR**	**AIR_domesti**	**AIR_usentry**	**GA**	**GA_dome**	**GA_DTT**	**GA_usenti**	**PCA**	**RAIL**	**RAIL_dom**	**RAIL_usentry**	**RAIL_us_r**	**ROAD_bet**	**ROAD_us_**	**ROAD_use**	**ROAD_us**
9	**Baseline**	current	current	current	current	current	current	current	current	current	current	current	current	current	current	current	current
10	**RailPOEpassenger**	current	current	lockdown	current	current	lockdown	lockdown	current	current	current	current	current	lockdown	current	lockdown	current
11	**SmallMaritimeVessel**	current	current	lockdown	current	current	lockdown	lockdown	current	current	current	lockdown	current	lockdown	current	lockdown	current
12	**AirCargo**	current	current	cargoMajorCurrent	current	current	lockdown	lockdown	current	current	current	lockdown	current	lockdown	current	lockdown	current
13																	
14																	

4.5.1 Running an Excursion Scenario

The latest version (as of 30 July 2015, version 1.0.13) adds the capability to run excursion runs via the script. To run that feature requires that the original scenario be named ending with "_ExpectExc" (the name is the first column for each scenario in the scenario list of the input file). The excursion run needs to follow it directly and be named ending with "_Excursion". They can occur anywhere in the scenario list as long as they're adjacent and in that order. For example, the above Patriot Script input file modified to run some excursion runs might look like this:



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
1																
2	**SwamiSource terminal**	pakistan	russia	france	north korea	south africa										
3	**SwamiTarget terminal**	washington dc	los angeles	dallas	chicago											
4																
5	**DEVICE**	GenericWGPu	LEU	HEUGun	HEUImpl											
6	**SHIELDING**	none	none	optimum	optimum											
7																
8		**AIR**	**AIR_domestic**	**AIR_usentry**	**GA**	**GA_domestic**	**GA_DTT**	**GA_usentry**	**PCA**	**RAIL**	**RAIL_domestic**	**RAIL_usentry**	**RAIL_us_ring**	**ROAD_between_poe**	**ROAD_us**	**ROAD**
9	**Baseline_ExpectExc**	current	current	current	current	current	current	current	current	current	current	current	current	current	current	curren
10	**Baseline_Excursion**	lockdown_worldwide	lockdown	lockdown	lockdown_worldwide	lockdown	lockdown	lockdown	enhanced	current	current	current	current	lockdown	lockdown	lockdo
11	**RailPOEpassenger**	lockdown_worldwide	lockdown	lockdown	lockdown_worldwide	lockdown	lockdown	lockdown	enhanced	current	current	current	current	lockdown	lockdown	lockdo
12	**SmallMaritimeVessel**	lockdown_worldwide	lockdown	lockdown	lockdown_worldwide	lockdown	lockdown	lockdown	enhanced	lockdown_worldwide	lockdown	lockdown	current	lockdown	lockdown	lockdo
13	**AirCargo_ExpectExc**	current	current	current	lockdown_worldwide	lockdown	lockdown	lockdown	enhanced	lockdown_worldwide	lockdown	lockdown	current	lockdown	lockdown	lockdo
14	**AirCargo_Excursion**	lockdown_worldwide	lockdown	current	lockdown_worldwide	lockdown	lockdown	lockdown	enhanced	lockdown_worldwide	lockdown	lockdown	current	lockdown	lockdown	lockdo
15																
16																
17																
18																

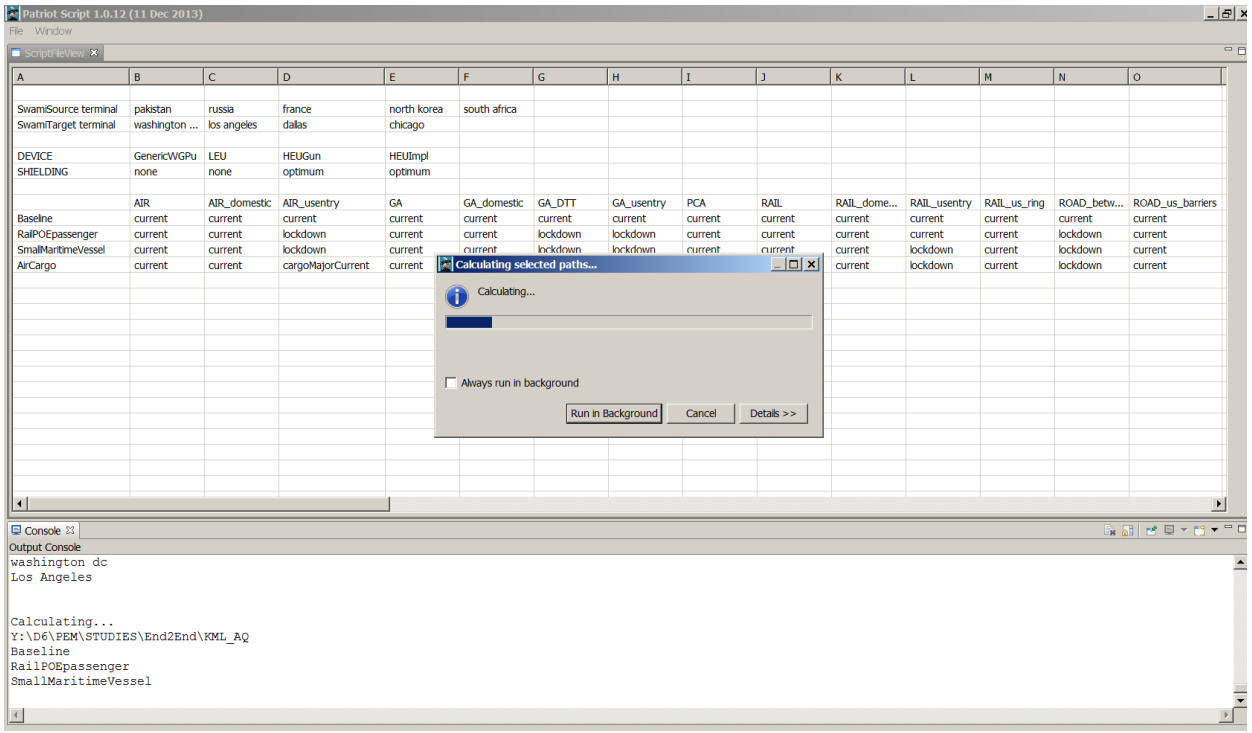
Here, the “Baseline” and “AirCargo” scenarios have been modified to run an excursion after running the original runs by first changing their titles to “Baseline_ExpectExc” and “AirCargo_ExpectExc”, respectively. Then, immediately following each of these newly titled scenarios, is the excursion run with the titles, “Baseline_Excursion” and “AirCargo_Excursion”, respectively. The program will run as usual and the output for this particular file will have six scenarios, two of which will be excursion scenarios.

4.6 Select a PATRIOT server

After the spreadsheet is saved, we can open up Patriot Script and set it to the appropriate PATRIOT server as described above (while choosing the server options in the “Window→Preferences” window, leave the “Save KML path files” checkbox checked as is the default setting). Click on the “File→Open…” menu and navigate to where you created and saved your version of PatriotScriptDemoRun.xls and select it. PatriotScript will show the contents of the file in its view window.

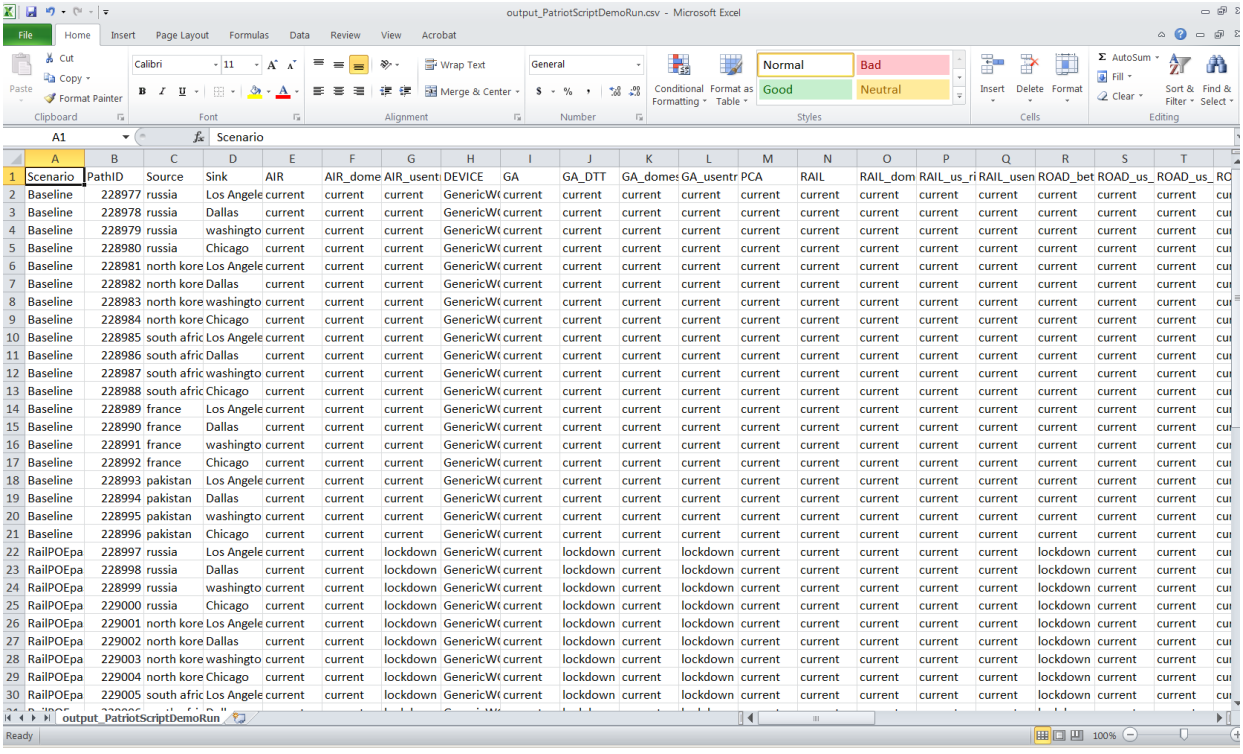
4.7 Start the runs

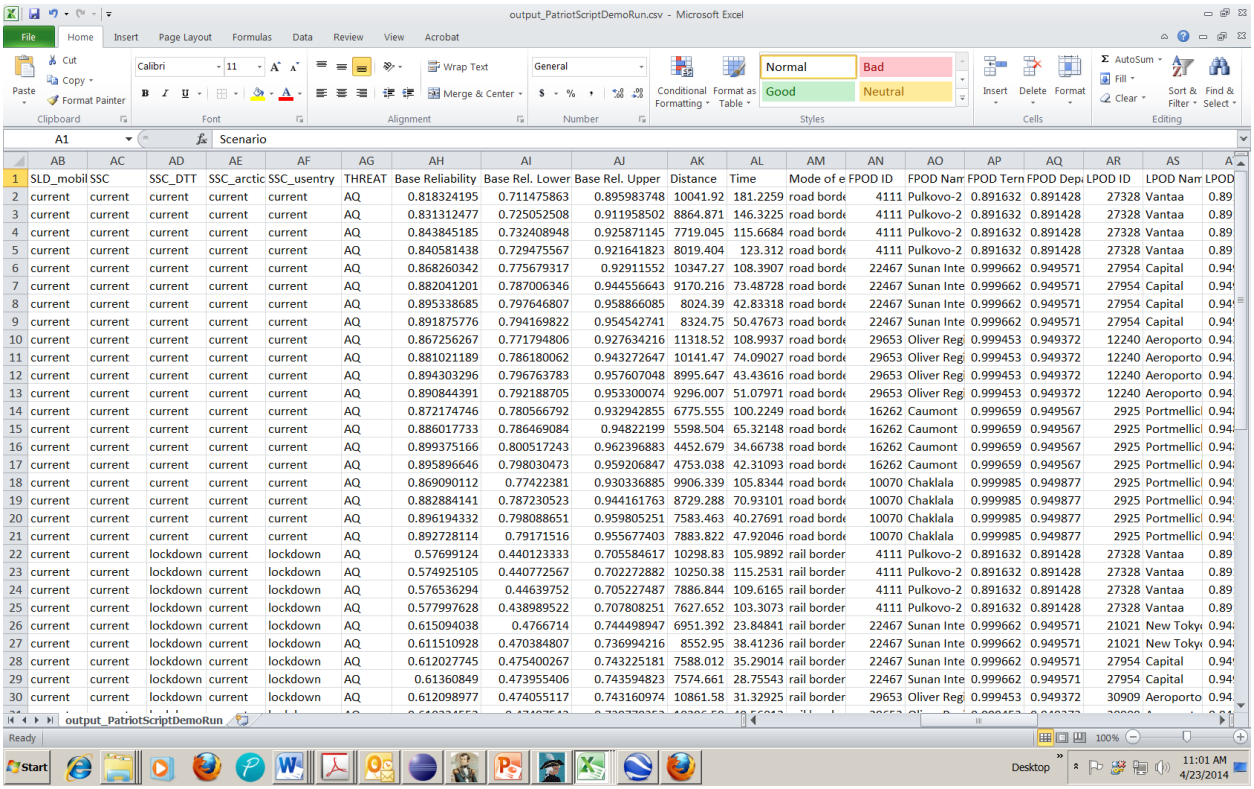
Now to set PatriotScript to run your file, click “File→Run” and choose where you’d like the output files to be stored. Assuming there are no errors in your file, it should begin running, showing a progress bar, and the console view should begin to show some output (see below). When the script is done running (which takes roughly 45 minutes to finish), you should have an output file called “output_PatriotScriptDemoRun.csv” and a folder titled “KML_AQ” which contains all your KML path files. There are a total of 320 paths for this model run.



4.8 View the output

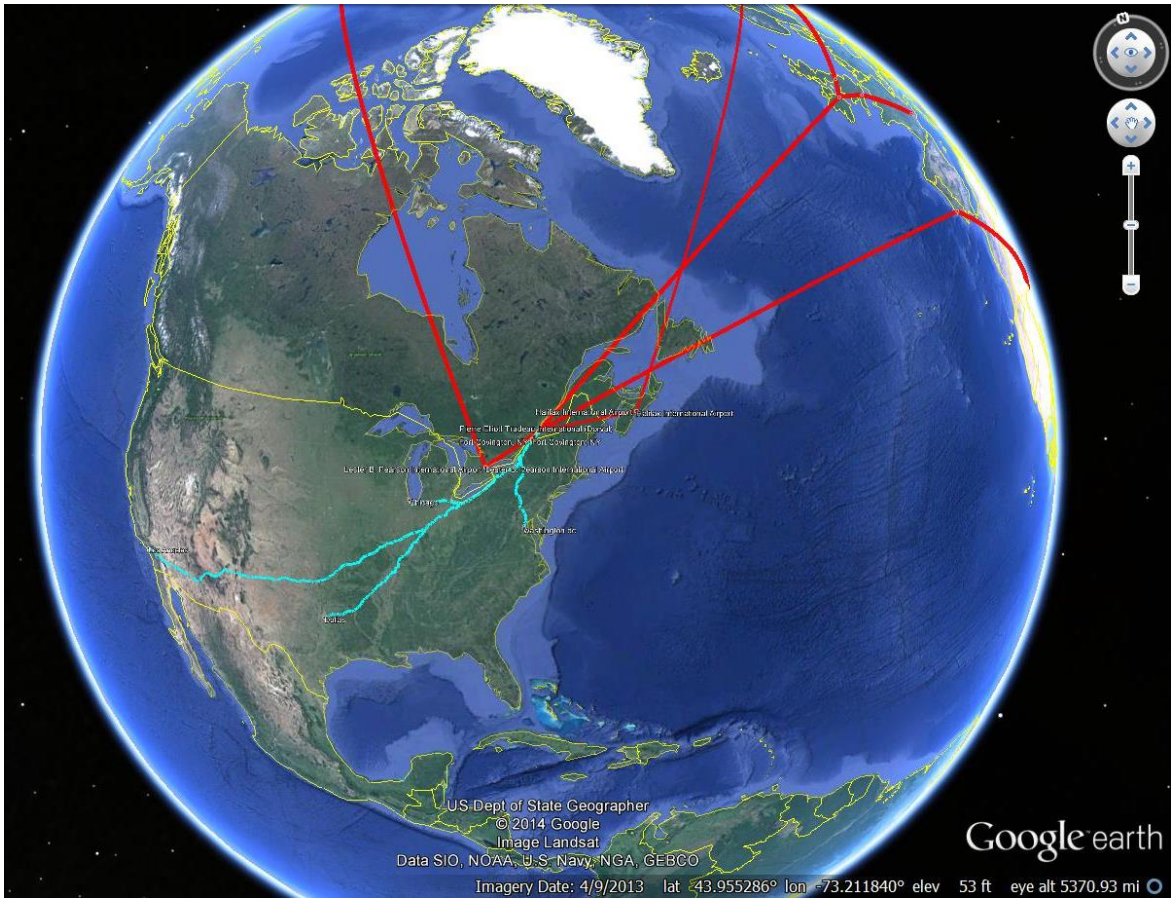
The PatriotScript output written in output_PatriotScriptDemoRun.csv is similar to the one produced by Patriot Client. Each line of the output file is associated with one of the attack routes. In this case the script computed 320 paths (5 origins × 4 destinations × 4 devices × 4 scenarios). The main difference with Patriot Client is that the first column contains the scenario name specified in the script and that all individual architecture options are displayed in separate columns. For example, note that the first 20 lines of the output file contain “Baseline” as the scenario name, corresponding to paths from the 5 origins to the 4 destinations transporting the first device “GenericWGPu/none”. The next 20 lines in the output file correspond to attack routes for the same combinations of origins, destinations and device, but for the second GNDA excursion named RailPOEpassenger, and so on. The individual architecture options and the shielding and device are displayed in columns “E” through “AF”. Column “AG” contains the Threat group, currently displaying “AQ”, and starting in column “AH” the same output as in the Patriot Client is displayed: “Base Reliability”, “Base Rel. Lower”, “Base Rel. Upper”, etc.





4.9 View specific routes

Individual routes can be visualized using Google-Earth or Patriot Viewer. Patriot Script creates a subdirectory named KML_AQ in the output directory chosen by the user. The routes are grouped by device and scenario. For example, the file named “GenericWGpu_non_Baseline” contains the 20 routes from the 5 origins to the 4 destinations while transporting a GenericWGpu device with shielding of “none” using the Baseline scenario.


US Dept of State Geographer
© 2014 Google
Image Landsat
Data SIO, NOAA, U.S. Navy, NGA, GEBCO
Imagery Date: 4/9/2013 lat 43.955286° lon -73.211840° elev 53 ft eye alt 5370.93 mi
Google earth